Universidad del Valle de Guatemala

FACULTAD DE INGENIERÍA

DEPARTAMENTO EN CIENCIAS DE LA COMPUTACIÓN Y TECNOLOGÍAS DE LA INFORMACIÓN

Sistema de monitoreo de Energía Eléctrica para hogares

Trabajo de graduación en modalidad de Megaproyecto presentado

por:

Boris Fernando Becerra Pelaez,
Oscar Estuardo Gil Sánchez,
Diego Renato Pérez Bercian,
Luis Alberto Suriano Saravia, y
Ricardo Alberto Zepeda Flores
para optar al grado académico de Licenciados en Ingeniería en Ciencias de la Computación y Tecnologías de la Información; y
Cristian Gustavo Pinelo Contreras
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

GUATEMALA
2016

Sistema de monitoreo de Energía Eléctrica para hogares

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Sistema de monitoreo de Energía Eléctrica para hogares

Trabajo de graduación en modalidad de Megaproyecto presentado por:

Boris Fernando Becerra Pelaez,
Oscar Estuardo Gil Sánchez,
Diego Renato Pérez Bercian,
Luis Alberto Suriano Saravia, y
Ricardo Alberto Zepeda Flores

para optar al grado académico de Licenciados en Ingeniería en Ciencias de la Computación y Tecnologías de la Información; y

Cristian Gustavo Pinelo Contreras

para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,
2016

Vo.Bo.

(f)_____
(MSc. Douglas Barrios)

(f)_____
(MSc. Carlos Esquit)

(f)_____
(Ing. Lynette García)

Fecha de Aprobación: Guatemala 30 de Noviembre del 2016

PREFACIO

Con base en una serie de entrevistas realizadas a personas guatemaltecas de diversa procedencia, por parte de nuestro equipo, se hizo notoria la falta de una forma de monitorear el consumo de servicios públicos como la energía eléctrica, porque a pesar de que se emplean diversos métodos como el uso limitado de aparatos electrónicos o utilzar bombillos ahorradores en el consumo, existe incertidumbre sobre la efectividad de estos, ya que las personas que los ponen en práctica deben esperar a la llegada de la factura mensual del consumo del servicio. Además fue evidente la falta de conciencia sobre el impacto ambiental que tiene el ritmo de consumo de energía eléctrica.

Debido a las razones anteriores, el problema en que se enfoca este proyecto es que no existe una manera para realizar monitoreo del consumo de energía eléctrica para el usuario promedio guatemalteco, lo cual provoca un consumo descontrolado de electricidad. En respuesta a esto se propone la plataforma con nombre EnerSave, que busca monitorear en tiempo real del consumo eléctrico de los usuarios, y además, predecir el consumo que estos tendrán una hora por delante de la actual. Con esto se busca que las personas tengan más control sobre el consumo que tienen de energía eléctrica. Dicha plataforma es solamente un prototipo por lo cual se incentiva a seguir con su desarrollo y mejora.

Para el presente estudio se contó con la participación de la familia Pérez Bercian, agradeciéndoles por habernos permitido realizar la prueba del prototipo en su hogar y brindarnos el apoyo así como la paciencia necesaria dentro del proceso. De igual modo se contó con el apoyo de nuestros asesores, nuestra coordinadora y nuestro director del departamento de Ingeniería en Ciencias de la Computación y Tecnologías de la Información, que fueron guía y motivación en el desarrollo de este proyecto, además de brindarnos las herramientas y recursos necesarios para llevar a cabo el prototipo de la plataforma.

ÍNDICE

LISTA DE CUADROS

LISTA DE FIGURAS

RESUMEN

El proyecto inició utilizando la metodología de Design Thinking. Esto nos permitió conocer que la falta de monitoreo del consumo eléctrico afecta negativamente a las personas. Con esto se descubrió que en Guatemala existe la necesidad por reducir el consumo de energía eléctrica.

En aras de minimizar el impacto que produce el consumo de energía eléctrica se crea "Ener-Save", una plataforma que permite monitorear el consumo en tiempo real. El usuario adquiere el producto, que incluye un componente electrónico que se conecta a la red eléctrica, y un acceso a la plataforma web, desde la cual puede monitorear su consumo, a través de gráficas y anális de datos. Esta plataforma se dividió en seis módulos: Sensores, almacenamiento de información y servicios web, análisis de datos, visualización, seguridad e integración.

Cada módulo desempeña una función específica dentro del proyecto y el funcionamiento en general se dá de esta forma: el módulo de sensores se encarga de hacer las mediciones, recolectar la información sobre el consumo y enviárla a un sistema de almacenamiento en la nube. El módulo de seguridad, se encarga de proteger la información para que nadie pueda interceptarla y leerla, y que llegue de manera íntegra al servicio de almacenamiento. El módulo de almacenamiento y servicios web, se encarga de almacenar la información enviada por los sensores en un servidor central en la nube, y además, hace la información disponible a través de servicios web para su acceso remoto. En otro servidor, existen los módulos de análisis de datos, integración y visualización. Es el módulo de integración el encargado de comunicar ambos servidores, de forma interna. De esta forma, el módulo de análisis de datos puede acceder a ellos, y hacer un análisis predictivo y de clustering sobre los datos de consumo, que luego son enviados de vuelta al servicio de almacenamiento. Por último, el módulo de visualización se encarga de acceder a los datos de consumo a través de los servicios web y los muestra al usuario final, por medio de una interfaz web y componentes gráficos, que representan las funcionalidades del sistema.

I. INTRODUCCIÓN

Este proyecto busca una solución a la problemática que tienen muchos hogares guatemaltecos al no tener control sobre el consumo que están realizando de energía eléctrica, produciendo aumento en el pago de este servicio. Por ello, la plataforma desarrollada permite a los diversos usuarios visualizar el consumo de electricidad que están realizando pudiendo controlar y predecir el gasto de energía que están teniendo. Nótese que la respuesta dada se limita a proponer un prototipo.

El proyecto inició utilizando la metodología de Design Thinking. Esto permitió observar que el consumo eléctrico puede afectar negativamente a las personas por las consecuencias ambientales y económicas. También se descubrió que las personas no tienen un control efectivo sobre su consumo de energía eléctrica. Se encontró que el análisis del monitoreo de consumo eléctrico puede dar ahorros significativos. Con base en esto se propone un sistema en el cual se pueda monitorear el consumo de energía en tiempo real.

Una vez definida la plataforma a desarrollarse, se procedió a identificar los elementos clave para su desarrollo. Estos fueron: un sensor capaz de medir el consumo de energía, el almacenamiento del consumo en una base de datos para luego ser mostrada y visualizada por el usuario (consumidor), proteger la información del usuario y analizar la información. De estos elementos surgen los módulos que intervienen en el proyecto: Sensores, almacenamiento de la información y servicios web, análisis de datos, visualización, seguridad e integración.

Tras la finalización del proyecto se llegó a la creación de los diferentes módulos especificados anteriormente, los cuales eran capaces de comunicarse de forma lógica. Con ello, se logró un prototipo de sistema el cual es capaz de monitorear el consumo eléctrico de un hogar en específico.

II. Objetivos

A. Objetivo general

Desarrollar una herramienta que permita el monitoreo en el consumo de la energía eléctrica en los hogares guatemaltecos.

B. Objetivos específicos

- Investigar los sensores adecuados para medir voltaje y corriente para uso residencial.
- Implementar un sensor para obtener mediciones de voltaje.
- Implementar un sensor para obtener mediciones de corriente.
- Calcular la potencia consumida a partir de los datos obtenidos por los sensores.
- Investigar y utilizar protocolos de comunicación estándar para transmitir la información a un sistema de control central.
- Transferir los datos desde el control central a un servidor dedicado para su análisis.
- Asegurar el envío y recepción de información privada por medio del uso de algoritmos de cifrado adecuados.
- Proteger los tres pilares de la seguridad de la información, siendo estos la confiabilidad, integridad y disponibilidad de los datos recolectados.
- Seleccionar las herramientas a utilizar para el almacenamiento de los datos recabados.
- Implementar las bases de datos necesarias para almacenar la información de consumo energético de los usuarios de la herramienta.
- Desarrollar una plataforma que permita unificar y comunicar información entre el módulo de Almacenamiento de Información y servicios web y el módulo de análisis de datos
- Establecer estándares de comunicación entre diferentes módulos de la plataforma
- Proponer un método por utilizando algoritmos de análisis de datos para clasificar el consumo de energía de un lugar dado.
- Determinar el mejor de dos algoritmos específicos para predecir los niveles de consumo energético de un lugar específico.

- Mostrar la información sobre el consumo energético a través de una interfaz de usuario que cumpla con las características de utilidad y usabilidad.
- Definir las funcionalidades del sistema de monitoreo de consumo energético, disponibles a través de la interfaz de usuario.

III. Justificación

El proyecto inició con la utilización de Design Thinking para encontrar un problema y centrarse en el usuario para proponer una solución. Se utilizó Design Thinking porque se ha probado que en distintos negocios y profesiones funciona como un protocolo para identificar y solucionar un problema (Urserey, 2014). De esto se identificó la falta de control en el consumo de energía eléctrica como un problema.

De las entrevisas pasadas a consumidores, a inicios del proyecto, los consumidores opinaron que el monitorear el consumo eléctrico en el hogar puede mejorar su economía familiar y mejorar sus hábitos.

Existen sistemas donde el monitoreo de consumo de energía y análisis en base a la información de consumo han representado ahorros significativos, ya sea aplicando el monitoreo a edficios (Anastasi *et al*, 2010) o a hogares (Chandler, 2016). Por la utilidad que tienen estos sistemas se propone un sistema que monitoree el consumo en tiempo real y realice análisis con la información de consumo.

Desde el punto de vista de la Universidad del Valle de Guatemala, este proyecto servirá como una muestra de interés de la institución por colaborar con el desarrollo y bienestar de la población guatemalteca. También se puede utilizar como ejemplo de la calidad y complejidad de los proyectos que se realizan dentro de la universidad.

IV. Marco teórico

El proyecto inició aplicando la metodología de Design Thinking, el cual es un proceso análitico y creativo que permite que una persona se involucre en experimentar, crear, hacer prototipos, obtener retroalimentación y rediseñar (Razzouk & Shute, 2012). Es un proceso en el que se inicia definiendo el problema y luego se buscan soluciones para este. Se divide en cinco etapas. **Empatizar** es la capacidad de entender a las personas dentro del contexto del problema a resolver. **Definir** es la etapa en la que se determina el problema específico que se va a resolver. **Idear** es donde se generan las ideas para resolver el problema. **Crear prototipos** es una etapa interactiva en la que se crean artefactos que dan una idea de lo que puede ser el producto final. Por último se encuentra la etapa de **pruebas**, en la que se busca la retroalimentación de las personas para darse una idea si el usuario final usaría esta solución (Plattner, s.f).

Para poder abordar el problema se debe conocer a que se enfrenta, el consumo de energía es una preocupación importante a nivel mundial. El consumo de energía se puede dividir en cuatro diferentes áreas: residencial, comercial, transporte e industria.

En Estados Unidos se ha visto un alto crecimiento del consumo de energía y se espera que este consumo siga creciendo. Se espera que en dos décadas el consumo de energía aumente en un cuarenta por ciento (The National Academy of Sciences, Engineering).

El alto consumo de energía puede llegar a ser un problema en el futuro si no se controla. Distintas ciudades alrededor del mundo cuentan con proyectos que se enfocan en el ahorro de energía. Algunos ejemplos exitosos de esto son: Issy-lesMoulineaux, Abu Dabi y Chicago.

Issy-les-Moulineaux consta de análisis de producción y consumo de energía. Cuenta con noventa y cuatro hogares con sensores inteligentes que brindan información en tiempo real de consumo. En total se ha ahorrado 500kWh (ISSY Grid, 2011). Abu Dabi tiene como meta usar el sol como única fuente de energía. Se da prioridad a la optimización de recursos y al uso de energía sotenible. Este proyecto tuvo una reducción de consumo de energía de un cuarenta por ciento (Villamendy, 2014). Chicago es un ejemplo que resalta sobre los otros porque su enfoque principal no es el ahorro de energía, sino la implementación de una microred. Esta microred fue creada para estar preparados ante eventos climáticos extremos. El propósito entonces es proveer energía a través de la microred cuando existan eventos climáticos que afecten la red utilizada normalmente (Corporate News, 2014). La experiencia de estas ciudades hace notar la importancia de tener un proyecto de ahorro de energía en Guatemala.

El proyecto se dividió en seis módulos, sobre los cuales se realizaron investigaciones indepen-

dientes. A continuación, se presentan para cada módulo.

A. Sensores y protocolos

1. Potencia alterna monofásica. En el caso de la corriente alterna para un circuito resistivo toda la potencia es disipada o utilizada como energía. Cuando los circuitos no son puramente resistivos parte de la potencia consumida no es aprovechada como energía, haciendo que se cree un desfase entre voltaje y corriente. A este desfase se le conoce como ángulo de factor de potencia. Para estos circuitos hay tres potencias que se pueden calcular: potencia aparente, potencia real y potencia reactiva, de ellas se deriva el triángulo de potencia. Entre menor sea el ángulo de factor de potencia mejor se aprovechará la potencia brindada al circuito. (Vásquez, 2016)

El valor que cobra la empresa eléctrica es la magnitud de la potencia aparente (Vásquez, 2016). La potencia aparente se puede calcular de dos formas diferentes, dependiendo de qué variables se conozcan. Si se conocen los valores máximos de corriente y voltaje se calcula de la siguiente forma:

$$\left|\bar{S}\right| = \frac{|V|\ |I|}{2} \tag{IV..1}$$

Si se conocen los valores RMS (Root Mean Square) basta utilizar la siguiente fórmula para encontrar la magnitud de la potencia aparente:

$$\left|\bar{S}\right| = V_{rms} * I_{rms} \tag{IV..2}$$

Los valores RMS representan una media del valor absoluto de la señal alterna.

2. Sensor de efecto Hall. Es un dispositivo semiconductor que al exponerse a un campo magnético genera un voltaje de salida. Por lo tanto, los senseres de efecto Hall pueden revelar la intensidad de un campo magnético o el nivel de corriente a través de un dispositivo. Por esto hay dos aplicaciones evidentes para este tipo de sensores: medir la intensidad de un campo magnético cercano a un sensor (si la corriente aplicada es fija) y medir el nivel de corriente a través de un sensor (si se conoce la intensidad del campo magnético) (Boylestad, R. 2004).

3. Protocolos de comunicación Existen dos formas comúnmente utilizadas para el envío de información: en serie y paralelo.

a. Comunicación serial . Se envían los datos de una terminal a otra, bit por bit a través de un sólo canal siguiendo reglas de comunicación llamadas protocolos. Si la comunicación requiere de una señal de reloj se le conoce como síncrona, de lo contrario es asíncrona. Para que dos sistemas se puedan comunicar es necesario que coincidan en las características del envío: baud rate, cantidad de bits, bits de parada y paridad. Se distinguen tres tipos: orientados a caracter, orientados a bloque y orientados a bit (s.a., 2011). Algunos de los protocolos más comunes en serie son:

- RS 232. Es el estándar hallado en las computadoras personales y similares. Es utilizado para variados propósitos, como conectar impresoras o módems, así como en instrumentación industrial. Trabaja en un rango de -12 V a 12 V, siendo el primero equivalente a un 0 lógico y el segundo a un 1 lógico. Este protocolo se puede utilizar para comunicaciones seriales en distancias de hasta 50 pies (National Instruments, 2006).

- SPI. De las siglas en inglés (Serial Peripheral Interface), es un tipo de comunicación serial síncrona, requiere de una línea para la señal de reloj y líneas adicionales para indicar a qué dispositivo de la red se desea enviar la información y una línea donde se envían los datos. Siempre hay un dispositivo maestro y uno o varios esclavos, el maestro el el que determina todas las operaciones de envío y recepción que se realizan (Grusin, 2013).

  Es necesario indicar en que flanco del reloj se va a muestrear la data, ya sea en flanco de subida o de bajada, para que el sistema que recibe sepa en qué momento debe tomar el bit en la línea de datos. Se deben agregar tantas líneas como esclavos hayan en la red pues siempre se debe indicar con cuál se desea comunicar el maestro (Grusin, 2013).

- I2C. Es un protocolo que requiere de una línea de reloj y otra para datos, por lo que es síncrona. La ventaja respecto al protocolo SPI es que sólo requiere de dos líneas para que la red funcione. Siempre hay un dispositivo maestro y puede haber múltiples esclavos. Al momento de transmitir datos el dispositivo maestro activa la línea de reloj y envía dos datos a través de la línea de información: el primer dato corresponde a la dirección del esclavo con el que se desea comunicar y el segundo corresponde a la información que se desea transmitir. El dispositivo maestro puede enviar la información deseada en cualquier momento pero para leer información de un dispositivo esclavo el maestro debe primero indicar que desea solicitarle información, es decir que para obtener información del esclavo primero se le debe enviar data (Robot Electronics, 1999).

  Pueden existir maestros-esclavos, es decir, un dispositivo esclavo puede ser el maestro de una red secundaria de dispositivos. Esto representa mucha utilidad cuando se tienen redes grandes de sensores o dispositivos, teniendo un maestro universal y diversos maestros-esclavos de las redes secundarias. Algunos sensores ya vienen predeterminados para funcionar en protocolo I2C por lo que la dirección de esclavo es fija y no se puede cambiar (Robot Electronics, 1999).

- UART. El nombre viene del acrónimo (Universal Asynchronous Receiver Transmitter), utiliza solamente 2 líneas, una de transmisión y otra de recepción, sin línea de señal de reloj. Como no utiliza señal de reloj entonces la información enviada requiere de señales adicionales para poder delimitar qué bits contienen la información, por esto contiene bits de inicio y de fin. El programador o usuario no se deben de preocupar por ellos, el hardware del protocolo se encarga de agregarlos cada vez que se envía. Todos los bits en medio de estos son la

información que se desea comunicar (Durda, 2014).

El formato en que se envían los datos es en ASCII así que al recibir es necesdrio tener precaución pues si se desea comunicar un número 3 en la recepción se tendrá el ASCII de 3 que es un 33 en sistema hexadecimal o 51 en sistema decimal. La ventaja de este protocolo es que como las líneas de transmisión y recepción son separadas entonces se puede realizar el proceso de envío y recepción simultáneamente, a los protocolos con esta propiedad se les conoce como full duplex (Durda, 2014).

Para que la información sea confiable es necesario que ambos sistemas estén configurados a la misma cantidad de baudios, que define la velocidad a la que se transmite la información (Durda, 2014).

b. Comunicación en Paralelo . Se envían todos los bits a la vez, un bit por línea de comunicación. Sus ventajas respecto a la comunicación serial son que la transferencia de datos es más rápida y no necesita una señal de reloj, pero necesita más líneas físicas.

Algunos de los protocolos de comunicación en paralelo más comunes son:

- SPP (Standard Parallel Port). Introducido en 1987 por IBM. El bus de datos poseía 8 pines dedicados a tierra cuya funciós fue cambiada para poder ser líneas de datos, logrando una comunicación bidireccional simultánea (full-duplex) sin agregar pines adicionales al bus (Tyson, 2008).
- EPP (Enhanced Parallel). Creado por Intel, Xircom y Zenith en 1991. Fue creado con el propósito de funcionar con equipos distintos de impresoras que necesitaran mucha más velocidad de transmisión, como equipos de almacenamiento. Con este protocolo se logró aumentar la velocidad de transmisión de 500 kB/s a 2 MB/s (Tyson, 2008).
- ECP (Extended Compatibility). Anunciado en 1992 por Micronoft y Hewlett Packard, dado que el EPP no estaba hecho para trabajar con impresoras el ECP fue creado para proveer alta velocidad de transmición y funcionalidad mejorada para las impresoras (Tyson, 2008).

4. Comunicación inalámbrica.

a. Radiofrecuencia. La radiocomunicación es una forma de telecomunicación que se realiza por medio de ondas radioeléctricas. Se divide en dos grupos que dependen del medio de transmisión: guiadas y no guiadas (Canga, R.).

Dentro de las comunicaciones no guiadas se encuentra la comunicación por radiofrecuencia, cuya señal se propaga en el rango de 30kHz a 300GHz. Se le llama comunicación no guiada debido a que no hay una línea directa entre emisor y receptor, la señal se propaga por el ambiente sin

dirección específica hasta llegar a su destino. Debido a esto no es necesario que exista una línea de vista entre los dispositivos que se desea comunicar (Canga, R.).

Al no existir línea de vista la comunicación entre los dispositivos se completa por diferentes trayectorias debido a distintos fenómenos cómo difracción, refracción, reflexión y dispersión. La difracción sucede cuando la señal cambia de dirección al encontrarse con el borde de un objeto, a pesar de esto y que provoca pérdidas el cambio de dirección ayuda a la transmición de la señal. La reflexión de la señal se da cuando ésta choca con un objeto de dimensiones mucho mayores a las de su longitud de onda, provocando que un porcentaje sea transmitido y otro sea reflejado. Cuando la señal choca con un material que es un excelente conductor la reflexión es total, es decir, las pérdidas son menores y no hay refracción. La dispersión se da cuando la señal choca con objetos de dimensionos pequeñas pero numerosas entre sí. Al impactar, la señal se refleja en varias direcciones y en posible que se genere un cambio de frecuencia y polarización de la onda electromagnética. La dispersión sólo ocurre cuando la señal impacta con una superficie rugosa (Nocedal, 2006).

5. Comunicación alámbrica.

a. X10 . Es un protocolo de comunicación que utiliza una línea de poder como portador para comunicarse y controlar dispositivos compatibles. Un transmisor que trabaja con X10 envía señales de bajo voltaje superpuestas en las líneas de 120 V AC. Cualquier receptor X10 conectado a la línea de poder recibe todas las señales enviadas pero responde únicamente a aquellas que coincidan con su dirección de receptor (Plumley, 2004).

Este protocolo normalmente funciona una distancia hasta aproximidamente 30 metros de la ubicación del transmisor. Esta distancia de puede ampliar al conectar transmisores en distintos nodos se la red. Los receptores usualmente se conectan a televisores, puertas de garaje, aparatos de sonido y otros dispositivos con el fin de realizar automatización por control remoto (Plumley, 2004).

Comúnmente, un sistema X10 utiliza un transmisor central que activa distintos receptores. La ventaja es que no se necesita volver a cablear la red eléctrica si las líneas de poder son capaces de portar la carga. Los trasmisores y receptores se conectan a los receptáculos o se conectan manualmente a los dispositivos o a interruptores de luz (Plumley, 2004).

El protocolo X10 se utiliza comúnmente para automatización de hogares porque es confiable y barato, además no requiene hacer nuevas conexiones. Además, muchos productos de distintos proveedores ya poseen protocolo X10 incorporado (Plumley, 2004).

B. Seguridad de la información

La seguridad de la información es el área de informática, que se enfoca en la protección de la información contenida o circulante en un sistema tecnológico. Tiene un efecto significativo en base a la privacidad de los usuarios y la obtención de la información de terceros. La información

en su mayoría, está centralizada y tiene un alto valor en base a un contexto determinado; por lo que está con riesgo a modificación y mal uso por personas externas. Además, según Misfud, está definida como un conjunto de medidas técnicas, organizativas y legales, que permite a la empresa u organización asegurar la confiabilidad, integridad y disponibilidad de la información en todo momento. (Misfud, 2012)

Para poder proteger la información de un sistema tecnológico; es necesario que la empresa u organización se plantee un conjunto de políticas para la gestión de la información; el más común a utilizar según Misfud, es el Sistema de la Seguridad de la Información (SGSI). El objetivo de SGSI es proteger la información y los datos importantes, por lo tanto hay que determinar cuál es el valor más relevante o más importante que debe de ser protegido y qué tanta seguridad debe de ser implementada, sin afectar las demás características. (Misfud, 2012)

1. Bases de la seguridad informática

a. Integridad: Según la argumentación del Dr. Ed Gelbstein, la integridad de la información, hace referencia a la fidelidad de la información o recursos. Sirve para prevenir modificaciones no autorizadas de la información. Este término además está relacionado no solo con la integridad de los datos; sino con la integridad del origen; el cual es importante tomar en cuenta, debido a que puede afectar la credibilidad y confianza que las personas dan a la información. La integridad de los datos es la garantía de que nadie pueda acceder a la información sin modificarla con la autorización necesaria. Está igualmente ligado a la integridad personal; como la responsabilidad, confianza, entre otros aspectos. (Gelbstein, 2016)

b. Disponibilidad: Misfud define el concepto de disponibilidad como la accesibilidad en todo momento a personas autorizadas. El principal objetivo es prevenir interrupciones no autorizadas o controladas de los recursos informáticos. Un sistema está disponible cuando su diseño e implementación permite negar el acceso a datos o servicios determinados no autorizados. (Misfud, 2012)

c. Confidencialidad: Este término hace referencia a la necesidad de ocultar o mantener en secreto los recursos del sistema, por lo cual es necesario restringir acceso a la información únicamente por autorización y de forma controlada. Uno de los objetivos claves de la fiabilidad, es prevenir la divulgación no autorizada de la información; al ser información digital, no hay medidas físicas de protección. Una medida de protección tecnológica, es la Criptografía; mecanismos y metodologías de cifrado. A pesar de utilizar este tipo de mecanismos, existe un dato muy importante que hay que proteger: la llave de cifrado. Solo con esta llave se puede cifrar/descrifrar

la información, y la mayoría de veces la llave puede viajar por el internet, pudiendo ser capturada por terceros; comprometiendo la información. (Ávila, 2013)

Dependiendo de donde se estén enfocando las necesidades; se puede dar más prioridad a cualquiera de estas tres bases, tomando en cuenta que siempre, como buenas prácticas, se utilicen las tres, sin importar la prioridad.

2. Funciones Hash Las funciones Hash, se utilizan para asegurar el pilar de la Integridad dentro del envío y recepción de los mensajes a través de un canal inseguro. Según Susan Landau; estas funciones aceptan mensajes de cualquier tamaño como entrada, producir una salida que se modifique si se realiza algún cambio de caracteres dentro del texto de entrada, y debe de ser rápido. (Landau, 2006) Usualmente se utilizan las funciones Hash dentro de propositos de cifrado y descifrado, y tiene propiedades como

- Una función criptográfica Hash, debe de ser únicamente de una vía; esta función solo se utiliza para cambiar de texto claro a un valor hash, y no de hash a texto claro.
- Una función criptográfica Hash, debe de cambiar completamente su valor de salida, si la cadena de caracteres de entrada cambia en cualquier valor.
- Una función criptográfica Hash, debe de ser libre de colisiones; esto significa que debe de ser computacionalmente dificil encontrar dos diferentes textos de entrada que produzcan un mismo Hash. (Landau, 2006)

3. Criptografía La criptografía es una técnica de la seguridad, principalmente en la base de confidencialidad. Este posee dos métodos principales; el cifrado y descrifrado. Según la investigación realizada por Noelia y Jaquelina, el cifrado es una función matemática utilizada para convertir de texto plano o texto original, a texto inaccesible, o al menos más difícil de entender para una persona externa. El descifrado también es una función matemática, pero es contrario al cifrado; convierte de texto cifrado al texto original. Estos dos métodos funcionan bajo una llave o llaves (dependiendo del tipo de cifrado que se utilizará). La criptografía tiene como objetivos cumplir cada una de las bases de la seguridad, siendo estas la confidencialidad y la integridad de la información, según sea el caso. (Desiree,Edit, 2004)

Estos métodos están divididos por cifrados dependientes de la llave criptográfica; el cifrado/descifrado simétrico y el cifrado/descifrado asimétrico.

4. Cifrado simétrico es un método criptográfico en el cuál se utiliza una misma llave (llave privada) para cifrar y descifrar información. Esta es la llave que se utilizará en todo

momento para cifrar y descifrar la información transmitida a través de un canal inseguro. El cifrado simétrico es uno de los más rápidos y más comunes. Un ejemplo de uso, es entre dos usuarios que transmitan información; el remitente debe de enviarla debidamente cifrada con una llave predefinida hacia el receptor. El receptor debe de descifrar con la misma llave que el remitente. (De Luz, 2010)

Una ventaja de los algoritmos simétricos es la velocidad, la seguridad, y lo poco costosos computacionalmente hablando. Son utilizados para el cifrado de grandes cantidades de datos y traspaso de información por cualquier canal inseguro. Otra ventaja de este cifrado, es que implementan un control de acceso; si no se posee la llave, no se puede acceder a la información. Por otra parte, una desventaja de estos algoritmos es el manejo de la llave y distribución; si la distribución está mal gestionada, se podría decir que la comunicación no es segura y se debe de generar otra llave. Este cifrado puede ser elaborado por diferentes algoritmos criptográficos. Estos algoritmos reciben como parámetro de entrada: texto plano o texto cifrado, y la llave criptográfica. Como salida se genera un texto inaccesible o texto plano. Dependiendo de si se va a cifrar o descrifrar, según menciona Yuri y Haider en su comparación de algoritmos simétricos. (Medina & Miranda, 2015)

El cifrado simétrico tiene dos distintos modos de operaciones y manejo de datos. Estos se dividen en Cifrado de Bloques y Cifrado de Flujo.

El cifrado simétrico de bloques es una función que mapea bloques de texto plano de longitud n-bit a bloques de texto cifrado de longitud n-bit. Este puede ser visto como una simple sustitución cifrada de un gran tamaño. En otras palabras, divide el texto en bloques pequeños de un bit o un byte de largo. Se codifica cada una de estas divisiones dependiendo de los anteriores; este utiliza un generador de flujo de llave; donde elabora una llave de codificación diferente cada vez que se codifica.

Para la asignación de bloques, los algoritmos de cifrado simétrico realizan sustituciones y permutaciones en el texto plano, hasta obtener el texto cifrado. La operación de sustitución es el reemplazo de un valor de entrada por otro de los posibles valores de salida. La permutación es un tipo especial de sustitución en el que los bits de un bloque de entrada son reordenados para producir el bloque cifrado. Estos algoritmos pueden ser iterativos, donde funcionan aplicando transformaciones en sucesivas rotaciones a un bloque de texto plano. (Menezes, Oorschot & Vanstone)

Para poder aplicar un algoritmo por bloques, es necesario descomponer el texto de entrada en

bloques de tamaño fijo, realizándolo de dos formas; ECB (Electronic Code Book) o CBC (Cipher Block Chaining).

ECB (Electronic Code Book) es un modo de cifrado donde el mensaje en bloques se divide en k bits, cifrando cada uno. De la misma manera para descifrarlo, se divide el texto cifrado en bloques de los mismos k bits y se descifra cada uno. Es muy vulnerable a ataques, debido a que dos bloques idénticos generan el mismo bloque de salida. Tiene como ventaja que los errores de cifrado por bloques no se propagan. Como desventaja tiene que el mismo bloque de información se cifra de la misma manera, el cual puede determinar patrones de información si se repite el bloque, además es maleable, por lo que puede reordenar los resultados cifrados es reordenar los textos originales. (Fabio, 2010)

CBC (Cipher Block Chaining) es un modo que depende de la salida del cifrado del bloque anterior para utilizar en el bloque actual. (Menezes, Oorschot & Vanstone). Tiene como ventaja que los textos repetidos son mapeados a diferentes datos cifrados, y no específicamente en orden, el reordenamiento afecta el descifrado y el cifrado depende de los bloques de texto plano anteriores. Como desventajas tiene que los errores se propagan de bloque a bloque, es de cifrado secuencial y no puede utilizar hardware paralelo. (Fabio, 2010)

El cifrado de flujo por otra parte, son algoritmos que van realizando el cifrado incrementalmente, convirtiendo texto claro en texto cifrado bit por bit, a través de la operación XOR con el bit correspondiente del flujo de la llave, según menciona José González. La fortaleza de este manejo de datos, es específicamente los flujos de llaves, el cual es un flujo de bits de una longitud tan larga como sea el mensaje, generado a través de un generador pseudoaleatorio. (González, 2012). También este método de cifrado tiene sus distintos modos de operacion, entre los cuales residen; OFB (Output Feedback Mode), CTR (Counter) y CFB (Cipher Feedback Mode). (Fabio, 2010)

OFB (Output Feedback Mode) es un modo criptográfico donde crea un bloque pseudoaleatorio grande como entrada para poder cifrar los bloques de texto original. Crea un cifrado de bloques a un cifrado de flujo auto sincrónico. Incrementa la velocidad de cifrado y descifrado. (Menezes, Oorschot & Vanstone). Tiene como ventajas; el cifrado es aleatorizado, posee cifrado secuencial y tiene limitante para propagar errores. Como desventajas; está sujeto a limitaciones únicamente para los cifrados de flujo. (Fabio, 2010)

CFB (Cipher Feedback Mode) es un modo donde el mensaje utiliza un OR Exclusivo con entrada del cifrado del bloque anterior. Crea un cifrado de bloques a un cifrado de flujo auto sincrónico.

(Menezes, Oorschot & Vanstone) Tiene como ventajas; el descifrado es aleatorizado, los bloques cifrados dependen de todos los bloques de texto claro anteriores y reordenarlo puede afectar el descifrado. Como desventajas tiene que los errores se propagan en diferentes bloques y tiene un cifrado secuencial. (Fabio, 2010)

CTR (Counter) es un modo de operación que igual que OFB (Output Feedback) convierte texto cifrado de bloques hacia un cifrado de flujo. Fue introducido por Diffie y Hellman en 1979 y actualmente se ha convertido en un estándar de uso para cifrar información. CTR tiene distintas ventajas; entre ellas se menciona la eficiencia de software y hardware, preprocesamiento para poder aumentar la velocidad de cifrado, acceso aleatorio y seguridad que puede ser probada y simplicidad según mencionan los investigadores Helger Lipmaa y Phillip Rogaway. De la misma manera mencionan que como desventajas posee; no integridad, tiene propagación de errores, vulnerable al introducir errores de usabilidad y tiene un cifrado en forma de estados (se debe de iniciar nuevamente el cifrado para cifrar y descifrar). (Lipmaa y Rogaway, 2000).

a. *DES (Data Encryption Standard)* : llamado por primera vez Lucifer, fue el estándar de cifrado publicado por NIST (Instituto Nacional de Estándares y Tecnología) diseñado por IBM. Se convirtió en 1977 como un estándar y fue adoptado por diferentes agencias de Estados Unidos. Utiliza una llave de 56 bits; con bloques de salida de 64 bits, realizando 16 rondas ejecutando el algoritmo de cifrado. La seguridad de la llave criptográfica de este algoritmo está catalogada por el tamaño del mismo; al tener 56 bits de tamaño (un texto de cincuenta y seis valores 0 o 1), por lo que se convierte en 2^{56} posibilidades. Existen diferentes ataques y métodos registrados que pueden explotar las debilidades de DES, por lo que actualmente se conoce como un cifrado por bloques inseguro. (Fabio, 2010)

DES ha sido atacado utilizando una metodología de *Linear Cryptoanalysis*, el cuál ha podido comprometer todas las rondas del algoritmo. Se ha probado que para poder comprometerlo, la cantidad de evaluaciones para un atacante debe de ser al menos $2^{39} - 2^{43}$, la cantidad de información conocida requerida debe ser al menos 2^{43}, según el investigador Pascal Junod. (Junod, 2001)

b. *Triple-DES (3DES)* : este algoritmo fue diseñado para resolver las fallas que poseía DES, sin volver a elaborar todo el criptosistema. Básicamente, extiende el tamaño de la llave de DES mediante la sucesión de tres llaves diferentes; por lo que el tamaño de la llave es de 56, 112 o 168 bits (tres veces lo que era DES originalmente) y utilizando 48 rondas. Utiliza cifrado de bloques de tamaño de 64 bits. Este algoritmo aún tiene sospechas sobre sus fallos, debido a que no se han encontrado problemas dentro del mismo; por lo que se ha utilizado en un gran número

de protocolos de Internet. (Hamdan *et al.*, 2010)

Como se menciona dentro de una investigación realizada acerca de 3DES; muestra que Triple DES es tres veces más lento que DES, pero más seguro si es utilizado de manera correcta. Pueden existir diferentes conjuntos de llaves; todas las llaves son independientes y distintas, la llave uno y llave dos son independientes y distintas o si todas las llaves son idénticas. El tamaño de la llave fue incrementada en 3DES, para dar adicional seguridad; por lo que un conjunto de tres llaves como de 56 bits de tamaño, lo convierte en una llave criptográfica de 168 bits. Este algoritmo es aún utilizado por el gobierno de los Estados Unidos. La seguridad de la llave criptográfica de este algoritmo está catalogada de la misma manera calculada que DES, como 2^{168} posibilidades. (Karthik y Muruganandam, 2014)

3DES ha sido atacado utilizando una metodología de *Meet-In-The-Middle Attack*, el cual ha podido comprometer todas las rondas del algoritmo. Se ha probado que para poder comprometerlo, la cantidad de evaluaciones para un atacante debe de ser al menos 2^{113}, la cantidad de información requerida debe ser al menos 2^{32} y es necesario 2^{88} bloques de memoria, según el investigador Stephan Lucks. (Lucks, 1998)

Existen dos distintas formas para cifrar y descifrar dependiendo de la cantidad de llaves a utilizar. Por ejemplo, considerando que se tienen las siguientes llaves; llaves criptográfica *k1*, *k2* y *k3*, con distintos valores, se puede realizar utilizando tres distintas llaves y utilizando dos distintas llaves.

Para utilizar tres distintas llaves, en el cifrado; primero se cifrará con k1, luego se cifrará con k2 y por último se cifrará con k3: *CifrarK3 (CifrarK2 (CifrarK1 (Texto))*. Para descifrarlo; primero se descifrará con k1, luego se descifrará con k2 y por último con k3: *DescifrarK3 (DescifrarK2 (DescifrarK1 (Texto))*. (Noura, 2015)

Para utilizar dos distintas llaves, en el cifrado; primero se cifrará con la llave k1, luego se descifrará con llave k2 y por último se cifrará con llave k1: *CifrarK1 (DescifrarK2 (CifrarK1 (Texto))*. Para descifrarlo; primero se descifrará con la llave k1, luego se cifrará con la llave k2 y por último se descifrará con k1: *DescifrarK1 (CifrarK2 (DescifrarK1 (Texto))*. (Noura, 2015)

c. *AES (Advanced Encryption Standard) Rijndael*: conocido también como Rijndael, es un cifrado por bloques, con bloques de datos de tamaño de 128 bits; utilizando llaves simétricas de tamaño de 128, 192 o 256 bits y utilizando 10, 12 o 14 rondas dependiendo del tamaño

de llave criptográfica, según se muestra en el cuadro 1. Este algoritmo criptográfico fue creado por dos científicos de computación llamados; Vincent Rijmen y Joan Daemen. Se introdujo para ser un algoritmo más fácil de implementar, más eficiente, más flexible y de manera gratuita a los que ya existían actualmente. AES Rijndael empezó a competir contra 3DES, DES e IDEA (International Data Encryption Algorithm); estos algoritmos a comparación de AES Rijndael, son muy lentos, además que IDEA no era gratuito. Está catalogado actualmente como un estándar para el gobierno de los Estados Unidos y numerosas organizaciones. Como menciona Douglas Selent en su investigación sobre ADS; eExisten distintos ataques realizados a este algoritmo, pero todos de ellos requieren computacionalmente más recursos, por lo que son muy improbables. (Douglas, 2010)

Según el informe sobre la comparación de algoritmos simétricos por Bruce Schneier y Doug Whiting, Rijndael mostró ser uno de los algoritmos donde el cifrado y descifrado es dependiente de la llave criptográfica. Si la llave criptográfica es 256 bits, este se calcula que tendrá un 40 % más lentitud a comparación a otros algoritmos; de igual manera con una llave de 192 bits, será un 20 % más lento. (Schneier,Whiting, 2000).

AES ha sido vulnerado utilizando una metodología de *Related-Key Attack*, el cuál ha podido comprometer 9 de las 14 rondas. Se ha probado que para poder vulnerarlo, la cantidad de evaluaciones para un atacante debe de ser al menos 2^{224}, la cantidad de información conocida requerida debe ser al menos 2^{77}, según los investigadores de la Universidad de Mannheim en Alemania. (Kelsey *et al.*, 2000)

En el concurso para elegir el nuevo algoritmo AES, siguiendo sus distintos criterios de eficiencia, seguridad, complejidad y flexibilidad; quedó en primer lugar, debido a su eficiencia de cifrado, flexibilidad en distintas aplicaciones y no posee gran complejidad. Además fue uno de los mayormente elegidos por medio de votación, siendo 86 votos a favor y 10 votos en contra. (Miles, 2000)

Cuadro 1: Comparativa AES Rijndael por tamaño de bloque y posibilidades de llave

Tipo de cifrado	**Tamaño de llave (word)**	**Tamaño de bloque**	**Número de rondas**	**Posibilidad de llaves**
AES-128	4 (128bits)	4 (128bits)	10	$2^{128} = 3{,}4 * 10^{38}$
AES-192	6 (192bits)	4 (128bits)	12	$2^{192} = 6{,}2 * 10^{57}$
AES-256	8 (256bits)	4 (128bits)	14	$2^{256} = 1{,}1 * 10^{77}$

En las presentaciones que muestra Fabio Martignon, denota que este algoritmo utiliza un núme-

ro de rondas; el cuál es una composición uniforme y paralela de cuatro pasos; *SubBytes*, *ShiftRows*, *MixColumns* y *AddRound Key*. Los pasos que realiza AES Rijndael para el descifrado, es la misma manera que el cifrado, solo que de manera contraria.

Para SubBytes, este realiza sustitución de byte utilizando una matriz no lineal e invertible de tamaño 16x16 indexado por bits en hexadecimal. Para ShiftRows, este realiza corrimiento de bits de izquierda a derecha. En MixColumns, cada columna es un estado (4 filas de bytes) de otra columna, obtenida al multiplicarla con la posición de la matriz de hexadecimales. Por último AddRound key, es donde cada byte del estado (16 bits) se combina con la llave criptográfica utilizando la operación XOR. (Fabio, 2010)

d. *RC2 (Rons Code o Riversts Cipher 2)* : RC2 es un algoritmo designado por Ron Rivest para RSA Data Security Inc, en 1987. Es un algoritmo de bloques; donde utiliza bloques de tamaño 64 bits y tamaño de llave criptográfica variable; desde 8 a 1024 bits en pasos de 8 bits y utilizando 18 rondas. Los bloques son divididos en cuatro palabras, cada una de tamaño de 16bits. El texto cifrado es procesado como una función de texto plano y una llave criptográfica en base a cantidad de rondas. Existen dos tipos de cifrado y descifrado en base a ronda; cada una con distinta función: la función Mixing Round y Mashing Round. Esta implementación es rápida, pero no tanto como las siguientes implementaciones de RC. La seguridad de la llave criptográfica de este algoritmo, a pesar de ser variable; es recomendado utilizar los tamaños estándar, como 128 o 256 bits. (IPA, 2003)

RC2 ha sido atacado utilizando una metodología de *Related-key Attack*, el cuál ha podido comprometer todas las rondas. Se ha probado que para poder atacarlo, la cantidad de evaluaciones para un atacante debe de ser al menos 1 consulta de llave relativa y la cantidad de información conocida requerida debe ser al menos 2^{34}, según los investigadores John Kelsey, Bruce Schneier y David Wagner. (Kelsey *et al.*, 1997)

e. *RC4 (Rons Code o Riversts Cipher 4)* : RC4 es un algoritmo creado por Ronald Rivest en 1994 para RSA Data Security Inc. Es un algoritmo de flujo que utiliza una llave criptográfica de tamaño desde 40 hasta 2048 bits y utilizando 1 rondas. Se ha convertido en uno de los cifrados utilizados en llaves WEP (Wired Equivalent Privacy) y SSL(Secure Sockets Layer)/TLS(Transport Layer Security) (Gunasundari,Elangovan, 2014). Este algoritmo se considera como inseguro si no se implementa de una manera correcta. Actualmente organizaciones como Google lo utiliza en algunas de sus herramientas. Como mencionan Sheetal y Sandeep en su investigación acerca de distintos algoritmos simétricos, RC4 fue iniciado como un secreto; pero en Septiembre de 1994 se recibió una descripción de este algoritmo y su implementación tan fa-

mosa, que luego, fue utiliza en distintos protocolos de red y otros servicios. (Sheetal,Sandeep, 2014)

RC4 ha sido atacado utilizando una metodología de *PTW y Related-key Attack*, rompiendo el cifrado de las llaves WiFi WEP. Se ha probado que para poder atacarlo, la cantidad de evaluaciones para un atacante debe de ser al menos 2^{20}, y la cantidad de información sobre las llaves relativas conocidas requeridas, debe ser al menos $2^{16,4}$, según los investigadores Fluhrer, Mantin y Shamir. (Fluhrer *et al.*, 2001)

f. *RC5 (Rons Code o Riversts Cipher 5)* : En la investigación comparativa de los algoritmos simétricos, mostraron que RC5 es un algoritmo de bloque descrito por su simplicidad. Fue creado por Ronald Rivest en 1994; llamado Riverst Cipher o Ron's Code (RC5). Este tiene una variación de tamaño de bloque de 32, 64 o 128 bits. Utiliza una llave de tamaño de 0 a 2040 bits y un número de rondas desde 1 a 255, pero con 12 rondas sugeridas. Los valores recomendados son un tamaño de 64 bits para bloque, 128bits para llave y 12 rondas. Las diferencias con RC2 y RC4, es que RC5 utiliza rotaciones dependientes de la información, utilizando funciones de XOR y adiciones modulares; además de que posee tres subrutinas; expansión de la llave, cifrado y descifrado. Este algoritmo es más lento que sus predecesores. Para la cantidad de posibilidades que tiene una llave criptográfica, esta puede ser calculada como 2^{128}. (Gunasundari,Elangovan, 2014)

RC5 ha sido atacado utilizando una metodología de *Timing Attack*, comprometiendo todas las rondas del algoritmo. Se ha probado que para poder atacarlo, la cantidad de evaluaciones para un atacante debe de ser al menos 2^{28} en el mejor de los casos y al menos 2^{40} en el peor. La cantidad de información cifrada debe ser al menos 2^{20}, según los investigadores Helena Handschuh y Howard M. Heys. (Handschuh,Howard, 1999)

g. *RC6 (Ron's Code o Riverst's Cipher 6)* : RC6 salió como consideración que RC5 se había convertido en un potencial candidato para el concurso de AES. Se realizaron modificaciones para poder ser parte de los requerimientos que exigia AES; por lo que pueda ser una modificación de RC5. Utiliza cualquier tamaño para bloque, número de rondas y llave criptográfica; lo usual es utilizar bloques de tamaño de 128 bits, 20 rondas y llaves criptográficas de 128, 192 o 256 bits. (Rivest *et al.*, 1998) Actualmente está vulnerable a los ataques diferenciales y los ataques lineales. (Rivest *et al.*, 1998)

Además, RC6 está actualmente vulnerable a un fenómeno estadístico no muy práctico, en el cuál se ha probado que para poder vulnerar 15 de las 20 rondas recomendadas; es necesario que la cantidad de evaluaciones para un atacante debe de ser al menos 2^{122}, la cantidad de información

conocida requerida debe ser al menos 2^{118}, y es necesario 2^{112} bloques de memoria, según los investigadores de Gemplus, una empresa enfocada en seguridad digital. (Gilbert *et al.*, 2001)

RC6 dentro del concurso de los nuevos algoritmos de AES, quedó como cuarto dentro de los cinco finalistas. Este algoritmo fue colocado en esa posición debido a que tiene un margen de seguridad relativamente bajo a comparación con los demás finalistas. A pesar de eso, es un algoritmo caracterizado por su velocidad de cifrado y descifrado. (Miles, 2000)

h. *MARS* : Según el artículo elaborado por investigadores de IBM, MARS es un algoritmo simétrico que opera con 128 bits de tamaño de bloque y un tamaño de llave criptográfica variable; tipicamente utilizando llaves de tamaño de 128, 192 y 256 bits, y utilizando 32 rondas. Actualmente está catalogado como uno de los algoritmos incluso más potentes que 3DES, y para poder atacar a este algoritmo, es necesaria mucha información. Este algoritmo utiliza distintas operaciones para poder funcionar; utiliza: OR's exclusivos, adiciones, substracciones, multiplicaciones y entre otros métodos. (Burwick *et al.*, 1999). Otra investigación más profunda, donde se analizan distintos criterios de este algoritmo; como la seguridad y eficiencia, muestra que, este algoritmo ofrece mejor seguridad y velocidad de operaciones que 3DES y DES. Se han encontrado diferentes ataques que contra este algoritmo. Meet-In-The-Middle attack es uno de los que ha podido realizar daño significativo, de igual manera el ataque de *Boomerang Attack*. (**?**)

AES MARS ha sido vulnerado utilizando una metodología de *Meet-In-The-Middle Attack*, el cuál ha podido comprometer 21 de las 32 rondas. Se ha probado que para poder comprometerlo, la cantidad de evaluaciones para un atacante debe de ser al menos 2^{232}, la cantidad de información requerida debe ser al menos ocho textos originales y es necesario 2^{236} bloques de memoria, según los investigadores John Kelsey y Bruce Schneier. (Kelsey,Schneier, 2000).

De igual manera, ha sido vulnerado utilizando el ataque *Boomerang Attack*, el cuál ha podido comprometer 12 de las 32 rondas. Se ha probado que para poder vulnerarlo, la cantidad de evaluaciones para un atacante debe de ser 2^{197}, la cantidad de información requerida debe ser al menos 2^{69} textos originales y es necesario 2^{73} bloques de memoria, según los investigadores John Kelsey y Bruce Schneier. (Kelsey,Schneier, 2000).

En un concurso realizado por AES; MARS fue uno de los 5 candidatos a elegir entre el nuevo algoritmo de cifrado fuerte siguiendo los criterios de AES. MARS quedó como cuarto lugar seguido de Rijndael, Serpent, Twofish y RC6. Es un algoritmo seguro, brinda mucha complejidad en las operaciones y no cifra de manera eficiente como los algoritmos anteriores a el. (Miles, 2000)

i. *CAST* : CAST es un algoritmo simétrico de bloques, desarrollado por Carlisle Adams y Stafford Tavares. Este algoritmo produce bloques de tamaño de 128bits y utiliza una llave criptográfica de tamaño 128, 160, 192, 224 o 256 bits, y utilizando 12, 16, 48 rondas dependiendo de la llave criptográfica. De igual manera que otros algoritmos como DES, CAST es un algoritmo que consiste de una serie de redondeos de sustituciones para hacer confusión y difusión. Para el cifrado, este algoritmo divide el texto de entrada en dos, la primera parte para ser transformado por una funcion F y es aplicada a una funcion XOR bit por bit con la otra mitad L_1; luego las mitades son cambiadas de orden. Conforme a la seguridad de la llave criptográfica, esta llave posee 2^{128} distintas posibilidades. Este algoritmo ofrece seguridad muy fuerte ante ataques lineales y differenciales. (Heys,Tavares, 1994)

CAST ha sido vulnerado utilizando una metodología de *Linear Cryptanalysis*, el cuál ha podido comprometer 24 de las 48 rondas (para la llave de 256 bits). Se ha probado que para poder vulnerarlo, la cantidad de evaluaciones para un atacante debe de ser al menos $2^{156,2}$, la cantidad de información conocida requerida debe ser al menos $2^{124,1}$, según los investigadores de la universidad de Shandong en China. (Wang *et al.*, 2009).

CAST con llave criptográfica de 256, fue uno de los algoritmos criptográficos simétricos que concurso dentro de la selección al nuevo AES (Advanced Encryption Standard), el cual no pudo quedar entre los primeros 5 finalistas, debido a que tenía muchas similitudes que AES Serpent. Al ser Serpent más versatil y tiene menos requerimientos de memoria, se eligió Serpent en vez de CAST-256, según el reporte de la primera ronda sobre la competencia a elegir el nuevo AES. (Nechvatal *et al.*, 1999)

j. *Blowfish* : Blowfish es un algoritmo de bloques que utiliza una llave de tamaño variable desde 32 hasta 448 bits, con bloques de tamaño de 64 bits, y utilizando 16 rondas. Fue elaborado por Bruce Schneider como un algoritmo rapido y gratuito a los algoritmos existentes. (Manku,Vasanth, 2015)

Cada rotación consiste en una permutación dependiente de la llave, y una sustitución dependiente de la llave y datos. Las operaciones son OR exclusivo sobre palabras de tamaño 32 bits. Es considerablemente más rápido que el algoritmo DES y está actualmente considerado como seguro; aunque se han encontrado diversas llaves débiles. Bruce Schneider muestra que es mejor utilizar el sucesor de Blowfish, llamado Twofish. Se han registrado multiples ataques contra este algoritmo, desde el momento que fué publicado. (Talens, 1999)

Blowfish ha sido vulnerado utilizando una metodología de *Differential Cryptanalysis*, el cuál ha podido comprometer 4 de las 16 rondas, según el investigador Vincent Rijmen, uno de los desarrolladores de AES Rijndael. (Rijmen, 1997).

k. *Twofish* : este algoritmo fue uno de los finalistas de la AES. Es una versión adaptada de Blowfish, el cual puede operar en llaves de 128, 192 y 256 bits, en bloques de 128 bits, y utilizando 16 rondas. Este utiliza 16 rondas de procesamiento para el cifrado y descifrado. La cantidad de posibles llaves existentes para este algoritmo son $128 = 2^{128}, 192 = 2^{192} y 256 = 2^{256}$. Las llaves criptográficas que provee este algoritmo no son débiles; por lo que muchos ataques no han podido ser exitosos. (Schneier *et al.*, 1998)

Twofish ha sido vulnerado utilizando una metodología de *Differential Attack*, el cuál ha podido comprometer 7 de las 16 rondas. Se ha probado que para poder vulnerarlo, la cantidad de evaluaciones para un atacante debe de ser al menos 2^{256}, según el investigador Niels Ferguson. (Ferguson, 1999).

Twofish fue uno de los primeros finalistas (3ero en la lista), en un concurso de AES, por mayor número de votaciones de la audiencia calificadora en base a distintos criterios; como flexibilidad en operaciones, seguridad y velocidad en cifrado por hardware. A pesar de todo, Twofish ha sido criticado por su complejidad. (Miles, 2000)

Según muestra Bruce Schneier y Doug Whiting en sus comparación con los cinco finalistas; Twofish puede cifrar y descifrar a la misma velocidad independientemente del tamaño de la llave criptográfica, pero a diferencia que se toma más tiempo generar la llave. Existen implementaciones donde la velociad de cifrado es diferente para diferentes tamaños de llave, pero es un algoritmo muy bueno para cifrar pequeños bloques de texto. (Schneier,Whiting, 2000).

l. *IDEA (International Data Encryption Algorithm)* : En la investigación creada por Mahamir y Arpit, mostraron diferentes características de este algoritmo; mostrando que IDEA es un cifrado de bloques, que fue desarrollado por Xuejia Lai y James L. Massey, en 1991. Este algoritmo opera con bloques de tamaño de 64 bits, con una llave criptográfica de tamaño de 128 bits, y utilizando 8.5 rondas. Existen tres funcionalidades principales, usadas en operaciones para diferentes grupos algebráicos, por la cual IDEA es dieferente a otros algoritmos. Para el cifrado, este algoritmo utiliza ocho rondas o pasos para poder cifrar de manera identica la información; seguida de una transformación final. Para el descifrado, se realiza el mismo mecanismo,

solo que de manera contraria; los bloques de información son descifrados de orden contrario que el proceso de cifrado. Tiene una gran resistencia contra ataques diferenciales en diferentes hipótesis, pero vulnerable contra ataques de colisión; si se utiliza de forma incorrecta reduciendo a 6 rondas, este algoritmo puede ser atacado. Para la posibilidad de llaves criptográficas de este algoritmo, se encuentran 2^{128} posibilidades. (Jain,Agrawal, 2014)

IDEA ha sido vulnerado utilizando una metodología de *Meet-In-The-Middle Attack*, el cuál ha podido comprometer 5 de las 8.5 rondas. Se ha probado que para poder vulnerarlo, la cantidad de evaluaciones para un atacante debe de ser al menos 2^{25}, la cantidad de información conocida requerida debe ser al menos 2^{127}, según los investigadores de la universidad de Bilkent en Turquía. (Demirci *et al.*, 2004).

m. *Serpent* : Serpent es un algoritmo de cifrado simétrico, siendo uno de los mejores finalistas dentro del concurso de AES. Este algoritmo opera con llaves criptográficas de tamaño 256 bits con bloques de 128 bits y realiza 32 rondas de cifrado. Los pasos de este algoritmo es realizar una permutación inicial, y en base a ella realizar operaciones de llave y transformaciones lineales, y así sucesivamente hasta llegar a 32 rondas. Por último se realiza una permutación al resultado de las operaciones. Actualmente existen diferentes tipos de ataque contra este algoritmo, siendo estos Linear Cryptoanalysis, Collision Attacks, Dictionary Attacks; siendo los más utilizados para atacar este cifrado. Este algoritmo en base a la seguridad de la llave criptográfica; esta tiene 2^{256} posibilidades, además que cifra en bloques de tamaño 128 bits, y utilizando 32 rondas. (Anderson,Biham,Knudsen, 2000)

Este algoritmo, dentro del análisis que se realizó por los investigadores del departamento de comercio de los Estados Unidos, mostraron que este posee más rondas de cifrado de lo que un algoritmo usualmente posee y puede resistir actualmente, además de poseer un alto márgen de seguridad. El requerimiento de memoria es moderado y el cifrado/descifrado es muy lento. (Nechvatal *et al.*, 2000).

Serpent ha sido vulnerado utilizando una metodología de *Linear Cryptanalysis*, el cuál ha podido comprometer 11 de las 32 rondas. Se ha probado que para poder comprometerlo, la cantidad de evaluaciones para un atacante debe de ser al menos 2^{187}, la cantidad de información requerida debe ser al menos 2^{118} textos originales, según los investigadores Eli Biham, Orr Dunkelman y Nathan Keller. (Biham *et al.*, 2002).

En el concurso realizado, se presentaron los algoritmos de Rijndael, Serpent, Twofish, MARS y

RC6, ordenado por mayor cantidad de votos. Serpent, fue uno de los mejores algoritmos presentados, llevandose el segundo lugar por medio de votación, sucesivamente de AES Rijndael. Serpent ha sido catalogado como uno de los algoritmos que tiene una estructura simple. (Miles, 2000).

Cuadro 2: Comparativa arquitectura cifrados simétricos

Cifrado	**Fecha**	**Cifrado**	**Llave**	**Bloque**	**Rondas**
DES	1975	Bloque	56 bits	64 bits	16
3DES	1978	Bloque	56, 112, 168 bits	64 bits	48
Rijndael	2000	Bloque	128, 192, 256 bits	128 bits	10, 12, 14
RC2	1987	Bloque	8 - 1024 bits	64 bits	18
RC4	1994	Flujo	40 - 2048 bits	-	1
RC5	1994	Bloque	0 - 2040 bits	64 bits	(Variable) 12
RC6	1998	Bloque	(Variable) 128, 192, 256 bits	(Variable) 128 bits	(Variable) 20
CAST	1996	Bloque	128, 160, 192, 224, 256 bits	128 bits	12, 16, 48
Blowfish	1993	Bloque	32 - 448 bits	64 bits	16
Twofish	2000	Bloque	128, 192, 256 bits	128 bits	16
IDEA	1976	Bloque	128 bits	64 bits	8.5
MARS	1998	Bloque	128, 192 y 256 bits	128 bits	32
Serpent	1998	Bloque	128, 192 y 256 bits	128 bits	32

Cuadro 3: Comparativa seguridad cifrados simétricos

Cifrado	**Posibilidades de llaves**	**Posibles ataques**	**Rondas comprometidas**	**Seguro**
DES	2^{56}	Exhaustive Key Search, Linear Cryptanalysis, Differential Cryptoanalysis y Brute Force Attack	16/16	No
3DES	$2^{56}; 2^{112}; 2^{168}$	Related Key Attack, Differential Attacks y Meet-In-The-Middle-Attack	48/48	No
Rijndael	$2^{128}; 2^{192}; 2^{256}$	Meet-In-The-Middle-Attack, Key recovery attack, Side Channel Attack	9/14	Sí
RC2	$2^8 - 2^{1024}$	Related Key Attack, Differential Cryptanalysis	18/18	No
RC4	$2^{40} - 2^{2048}$	Man-In-The-Middle-Attack, Related Key Attack	1/1	No
RC5	$2^0 - 2^{2040}$	Differential Cryptanalysis y Timing Attack	12/12	No
RC6	$2^{128}; 2^{192}; 2^{256}$	Differential y Lineal Cryptanalysis	15/20	Sí
MARS	$2^{128}; 2^{192}; 2^{256}$	Meet-In-The-Middle Attack, Boomerang Attack	21/32	Sí
CAST	$2^{128}; 2^{192}; 2^{256}$	Collision Attack, Lineal Cryptanalysis y Differential Attacks	24/48	Sí
Blowfish	$2^{32} - 2^{448}$	Differential Cryptanalysis	4/16	Sí
Twofish	$2^{128}; 2^{192}; 2^{256}$	Differential Attacks, Related Key Attack	7/16	Sí
IDEA	2^{128}	Collision Attack, Meet-in-the-Middle Attack y Linear Attack	5/8.5	Sí
Serpent	$2^{128}; 2^{192}; 2^{256}$	Dictionary Attacks, Collision Attacks, Linear Cryptanalysis y Differential Cryptoanalysis	11/32	Sí

En el año 1997, el Instituto Nacional de Estándares y Tecnología (NIST) inició un proceso para seleccionar un algoritmo de cifrado simétrico para proteger información federal sensitiva. En 1998, NIST aceptó a quince candidatos para poder iniciar la competencia en base a los criterios de eficiencia (tiempo y recursos computacionales) y seguridad. El instituto revisó cada una de ellas y eligió a cinco finalistas: MARS, Rijndael, RC6, Serpent y Twofish; donde luego de haberlos revisados, NIST seleccionó a Rijndael como el nuevo AES, según el reporte del desarrollo de la competencia. (Nechvatal *et al.*, 2000).

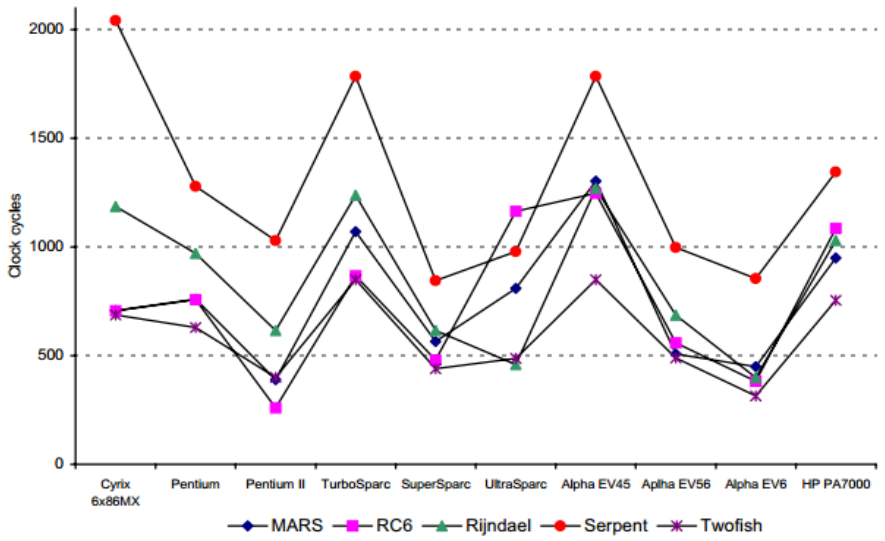
Cuadro 4: Comparativa cifrados simétricos finalistas de la competencia AES

Cifrado	**Rondas comprometidas**	**Votos**	**Rendimiento cifrado/descifrado**	**Rendimiento de creacion llave**
Rijndael	9/16	76	Alto	Alto
RC6	15/20	-14	Alto	Medio
MARS	8/14	-70	Medio	Medio
Twofish	9/32	10	Medio	Bajo
Serpent	6/16	52	Bajo	Medio

(Schneier,Whiting, 2000), (Miles, 2000), (Nechvatal *et al.*, 2000)

En una comparación de eficiencia para los cinco finalistas dentro del concurso de NIST a elegir el nuevo AES; Bruce Schneier y Doug Whiting, se realizaron análisis conforme a diferentes criterios; eficiencia en tiempo y memoria del cifrado, descifrado y la generación de llave criptográfica, y la seguridad para los algoritmos. Ellos mencionan que la elección del lenguaje de programación para su implementación no debería de importar. Las implementaciones fueron en diferentes ambientes, resultaron ser relativamente parecidos, brindando uno de los muchos resultados. (Schneier,Whiting, 2000).

Figura 1: Comparación realizada por investigadores sobre los 5 finalistas del concurso AES.



Como se aprecia en la figura anterior, Serpent ha sido uno de los más tardados en todos los sistemas probados. Por el contrario RC6 ha sido uno de los más rápidos. (Schneier,Whiting, 2000)

5. Cifrado asimétrico para este cifrado criptográfico, se utiliza un par de llaves para el envío de mensajes. Estas dos llaves deben de pertenecer a la misma persona que haya enviado el mensaje. Las llaves son diferentes, una es pública, la cual se entrega a cualquier usuario, y la otra llave es privada; donde el propietario debe guardarla para que nadie tenga acceso a ella. Este método criptográfico garantiza que se generará solo una vez esta pareja de llaves; por lo que se puede asumir que es muy difícil que un usuario tercero obtenga la misma pareja de llave.

Este cifrado salió para que los usuarios no intercambiaran las mismas llaves para su sistema de información; sino independientemente ellos puedan utilizar la llave ya sea pública o privada. Si el propietario utiliza la llave privada para cifrar el mensaje; cualquiera puede descifrarlo utilizando la llave pública; por lo que se determina la identificación y autentificación del remitente; es por eso el fundamento de la firma electrónica. (Medina & Miranda, 2015)

Las ventajas para este tipo de cifrado, es que existen dos llaves, una pública y otra privada; ofrece confidencialidad y privacidad en todo momento, brinda un control de acceso, también brinda el no repudio, donde un sistema no puede negar la creación o acceso de un sistema. La principal diferencia es que la llave pública que posee el remitente (o viceversa) es la que cualquier persona podrá tenerlo, pero no podra descrifar o leer la información debido a que está unida con la llave pri-

vada; la cual sólo la posee el receptor (o viceversa). Pero como desventaja, es que estos algoritmos son mucho más lentos que los algoritmos simétricos, y requieren mucha más computación, debido a la utilización de grandes cantidades de cálculos matemáticos, que por lo tanto la complejidades son muy altas, según los investigadores de la India; Sasi, Dixon y Wilson. (Sasi,Dixon y Wilson, 2014)

Este cifrado puede ser elaborado por distintos algoritmos, entre ellos se pueden mencionar:

a. *RSA* : es uno de los algoritmos más usados, más sencillo de entender y más fácil de implementar, en el cifrado asimétrico. Fue creado por Ronal Rivest, Adi Shamir y Leonard Addleman en 1977. Este algoritmo opera con una llave criptográfica mayor a 1024, mayor que la de los otros algoritmos, siendo esta incrementada por múltiplos de 256 bits; es recomendado una llave de tamaño 2048 bits. Es utilizado por servidores web y navegadores para asegurar el tráfico de red, asegura la privacidad y autenticidad de un correo, asegura también inicio de sesiones remotas, además de sistemas de pago por medio de tarjetas de crédito. Este se basa en la dificultad de la factorización de números grandes; por lo que las llaves públicas y privadas, se calculan a partir de un número proveniente de un producto de dos primos grandes. Dan Boneh menciona que un atacante deberá de enfrentar el problema de factorización para poder encontrar la llave. Para RSA existen distintos intentos de ataques a la implementación de este algoritmo; *Timing Attack* y *Random Faults*. A pesar que han pasado muchos años que han intentado atacar y romper el algoritmo RSA, no han habido problemas serios hasta el momento, según el investigador Dan Boneh. (Boneh, 1998).

Para poder generar las llave pública y privada de este algoritmo, según los creadores de RSA (Rivest, Shamir y Adleman) mencionan que para generarlo, este lo hace en base a estos pasos. Se eligen dos números primos distintos p y q principalmente aleatorios. Se realiza la operación de multiplicación de los mismos obteniendo un resultado $n = pq$ al cual se le aplica la función de Euler totient. Se elige un número entero e que esté dentro de 1 hasta el valor de la función de Euler que sea coprimo, y finalmente se obtiene un número entero d, obtenido del módulo inverso multiplicativo de e. Para los valores e y d son el exponente público y el exponente privado respectivamente. Para poder obtener la llave pública, esta se realiza un módulo del exponente e. Para la llave privada, se realiza el módulo del exponente d. (Rivest *et al.*, 2001).

De la misma manera, ellos explicaron el proceso de cifrado; el mensaje M se convierte en un entero m donde este se encuentre entre el valor 0 y n, siendo estos valores coprimos, y su máximo común divisor sea 1. Luego el opera $c \equiv m^e \ (mod\ n)$ utilizando el texto cifrado c y la llave pública e. Para el descifrado, se obtiene el valor m del texto cifrado c utilizando su exponente privado d realizando la operación $c^d \equiv (m^e)^d \equiv m \ (mod\ n)$; con m se puede recuperar el mensaje M realizando de la forma inversa. (Boneh, 1998)

.

Un grupo de estudiantes del Instituto de Administración y tecnologías de la información, elaboró una comparación de algoritmos de RSA con ElGamal. Brindaron distintos resultados, y ambos fueron igualmente prometedores al momento de comparar entre distintos criterios; como: tamaño de llave, velocidad y ataques conocidos. No hay mucha diferencia con estos algoritmos conforme a la seguridad, ambos ofrecen seguridad fuerte. RSA puede tener una llave criptográfica de mayor tamaño y no ofrece escalabilidad. (Shetty *et al.*, 2014).

Otra investigación, determina que la desventaja de utilizar este algoritmo, es la velocidad de cifrado. (Gurujara *et al.*, 2014).

b. *ECC (Elliptic Curve Cryptography)* Elliptic Curve Cryptography (ECC) fué creado en 1985 por Neil Koblitz y Victor Miller. Es un algoritmo de llave pública que provee los mismos mecanismos que los esquemas de RSA, pero utilizando llaves criptográficas con tamaño máximo de 571 bits. Su seguridad está basada en problemas de alta complejidad; utilizando el algoritmo Elliptic Curve Discrete Logarithmic Problem (ECDLP). Este algoritmo utiliza una curva elíptica en el cual las variables y los coeficientes son restringidas a elementos con campos finitos; utiliza dos familias de curvas elipticas para poder realizar aplicaciones criptográficas: *Prime Curves* y *Binary Curves*. Otra diferencia con RSA que brinda ECC, es la habilidad de poder realizar las mismas funcionalidades de RSA pero con una llave más pequeña. (Kumar *et al.*, 2012).

En un documento realizado por dos investigadores, Neal Koblitz y Alfred J. Menezes, muestran las diferentes especulaciones que brindaba la agencia Nacional de Seguridad de los Estados Unidos (NSA), al intentar colocar a ECC como el nuevo estándar de seguridad en los últimos años. Muchas de estas especulaciones brindan dudas a diferentes agencias y cripto analistas, acerca de si NSA quiere poner a ECC como estándar, debido a que puede descifrarlo fácilmente. En el documento ellos explican que han apoyado este algoritmo debido a su eficiencia y seguridad comparada con los otros dos algoritmos, RSA y ElGamal. Muchos de estos comentarios no son oficiales, entonces aún no se puede determinar con seguridad si es recomendado utilizar este algoritmo. (Shetty *et al.*, 2014).

Una investigación realizada por Swadeep, Anupriya y AnshulSachdeva; mostró una comparación en tiempo de cifrado/descifrado y las diferencias en tamaño de llaves criptográficas. Los resultados determinan que las operaciones del algoritmo RSA son más rápidos que el de ECC; con la diferencia que ECC ofrece más seguridad que RSA, a pesar que no se ha determinado el nivel de complejidad que esta tiene conforme a seguridad. (Singh *et al.*, 2013).

Otra investigación, determina que ECC es muy dificil de implementar que los demás algoritmos, como RSA; el cuál incrementa la posibilidad de implementar errores, reduciendo su seguridad. (Gurujara *et al.*, 2014).

c. *ElGamal* : ElGamal es un criptosistema creado por Taher Elgamal en 1984; en el cual está basado en el problema matemático del logaritmo discreto, basado en el algoritmo de Diffie-Hellman, según Andreas Meier. Este es utilizado en firmas digitales para cifrar y/o descifrar información. Según la publicación de los investigadores de los laboratorios GTE, ElGamal utiliza una llave criptográfica de tamaño recomendado de 2048 bits, y ofrece una buena escalabilidad, además de la velocidad de cifrado y descifrado que toma. También mostraron que ElGamal ofrece mayor seguridad que RSA, debido a que realiza más rondas de cifrado, por lo tanto es más lento para el cifrado y descifrado. (Tsiounis y Young, 2001). Esta seguridad se basa en la suposición que la función es utilizada únicamente en un sentido, debido a la dificultad de calcular el algoritmo discreto. Para el cifrado, este algoritmo primero debe de hacer algunos pasos.

Según Andreas Meier, la generación de la llave pública y privada de el algoritmo ElGamal se realizan distintos pasos. Se debe de generar aleatoriamente un número primo muy grande p y un generador g de un grupo multiplicativo Z_p^* de enteros módulo p. (Meier, 2005). Para poder obtener el grupo multiplicativo, según el libro aplicado de criptografía, se debe de seleccionar aleatoriamente un $(k-1)$bit primo q, donde k es el tamaño de la llave criptográfica, y realizar la operación $p = 2q + 1$ hasta comprobar que p sea un número primo. (Menezes, Oorschot & Vanstone). Para seleccionar la llave privada, se elige un número entero b del grupo Z de forma aleatoria con la condición que se encuentre en el rango de $1 \leq b \leq p - 2$ para ser el exponente privado. Luego se genera la llave pública $g^b mod p$. La llave pública será (p, g, g^b) y la privada b. (Meier, 2005).

Meier menciona que para poder cifrar se necesita el siguiente procedimiento; obtener la llave pública (p, g, g^b), preparar el mensaje M para cifrarse convirtiendolo a un grupo de enteros $(m_1, m_2, ...)$. Luego se debe de generar un exponente verdaderamente aleatorio crítico para seguridad k, y computar en base a la llave pública realizando la operación $g^k \ mod \ p$ y combinarla con el texto cifrado para enviar. Luego se cifra el mensaje M con el texto cifrado C, se iterará sobre el grupo de enteros calculandolo con la operación $c_i = m_i * (g^b)^k$. Ahora, para poder descifrar se deben de hacer los siguientes pasos. El texto cifrado contiene el exponente privado b y el exponente aleatorio k generado y se obtiene la llave compartida por medio de $(g^k)^{p-1-b} = (g^k)^{-b} = b^{-bk}$. Luego de haber encontrado la llave compartida se obtendrá el texto cifrado dividido en grupos por medio de la operación $m_i = (g^k)^{-b} * c_i \ mod \ p$, se agrupan todos los valores y se obtiene el texto descifrado. (Meier, 2005).

En una investigación realizada acerca de cifrado asimétrico ElGamal, muestra que su implementación es muy fácil, debido a las operaciones que utiliza; como la multiplicación y exponenciación en un algoritmo discreto. (Koscielny, 2004)

Cuadro 5: Comparativa algunos cifrados asimétricos

Cifrado	**Fecha**	**Llave**	**Implementación**
RSA	1977	<> 2048 bits	Fácil
ECC	1985	< 571 bits	Difícil
ElGamal	1984	<> 2048 bits	Fácil

6. Cifrado homomórfico Según los investigadores matemáticos Santi Martines, Victor Mateu, Rosana Tomás y Magda Valls; existe otro tipo de cifrado, el llamado Cifrado Homomórfico, en donde realizar una operación algebráica sobre un texto plano el que equivale a realizar una operación sobre el mismo texto cifrado. Si se realizan operaciones sobre los datos cifrados y luego se descifra el resultado, se obtendrá el mismo resultado que si se realizan las mismas operaciones sobre los textos originales. Este cifrado es utilizado para poder encadenar servicios sin exponer la información cifrandola y descifrandola, además de ser más eficiente para el sistema al no realizar estas operaciones (Martínez *et al*, 2012). En el 2009, dos investigadores dieron a conocer una aplicación práctica sobre este cifrado para una red de comunicaciones híbrida con arquitectura en forma de sensores y malla. Este sistema permite el envío de información hacia distintos servicios de manera transparente y son capaces de realizar análisis estadísticos con diferentes parámetros utilizando la información cifrada y su debida autenticación con cada nodo o servicio, según Roberto Riggio y Sabrina Sicari; investigadores italianos. (Riggio & Sicari, 2015).

7. Algoritmos de cifrado más utilizados Existen distintas empresas tecnológicas, y administradoras de servicios en internet u conexiones con ella; similares al enfoque del proyecto. Esto servirá para conocer el algoritmo que utilizan para cifrar sus datos y brindar una buena seguridad a sus clientes.

a. Dropbox : Dropbox es un servicio de almacenamiento de archivos, que ofrece almacenamiento desde la nube, sincronización de archivos, nubes personales y software para clientes. Este servicio en el ámbito de seguridad, almacena los datos en bloques discretos; el cual se fragmentan y cifran mediante el algoritmo de AES (Advanced Encryption Standard). Además utiliza un protocolo de SSL/TLS para la transferencia de estos archivos, el cuál crea un túnel seguro

protegido por el cifrado de AES de 128 bits o más.

b. Intel : Intel es una compañia multinacional de tecnología; el cuál está catalogado como uno de los más valorados y grandes creadores de chips, microprocesadores y procesadores dentro de computadoras personales. En el ámbito de la seguridad; la familia de procesadores de Intel Core del 2010; brinda un grupo de instrucciones para poder utilizar el algoritmo AES. Utilizan AES para proteger el tráfico de red, información personal y la infraestructura de los sistemas de tecnología en una corporación. Esta guia tiene como nombre: *Intel Advanced Encryption Standard (AES) New Instructions (AES-NI)*. Al momento de utilizarla, Intel asegura diferentes mejoramiento a algunas características dentro del sistema; unas de ellas son el mejoramiento de la eficiencia y mejoramiento de la seguridad.

c. Cisco : Cisco es una compañía multinacional de tecnología que diseña, manufactura y vende equipo de red a nivel mundial. En el ambiente de seguridad, Cisco recomienda diferentes algoritmos criptográficos para el cifrado de información, brindando los parámetros recomendados para su correcta implementación. Para los algoritmos de cifrado simétrico; Cisco menciona cuatro de ellos: DES, 3DES, RC4 y AES. Para DES y RC4, recomiendan no utilizarlo. Para 3DES ellos determinan que puede ser una buena opción, pero existen otros mejores. Finalmente ellos colocan como aceptable utilizar el algoritmo de AES utilizando el modo de CBC, con parámetros de tamaño de llave de 256 bits.

Dentro de sus recomendaciones de uso para algoritmos criptográficos asimétricos, la empresa CISCO, mostró que utilizar el algoritmo RSA con llave criptográfica de tamaño 3072 bits, muestra una seguridad aceptable para intercambio de llaves.

d. IBM, SoftLayer y eperi : IBM es una compañía multinacional de tecnología que manufactura y comercializa hardware, software y otros servicios de computadoras. SoftLayer es un servidor dedicado, almcenamiento administrado y proveedores de procesamiento dentro de la nube. eperi es una empresa alemana desarrolladora de productos de software de seguridad. Estas empresas poseen una alianza, y ellas están convencidas que la información y la seguridad tiene una gran prioridad. Ellos recomiendan utilizar el cifrado AES con llave criptográfica de 256 bits, o alguno de los algoritmos de la familia RC.

e. Google (Google Cloud Platform) : Google es una compañía multinacional de tecnología especializada en servicios relacionados con el internet; como busqueda, computabilidad dentro de la nube y desarrollo software. Google utiliza un algoritmo de cifrado para los datos

de usuario almacenados dentro de su plataforma de la nube llamada Google Cloud Platform; utilizando AES con llave criptográfica de tamaño 256 o 128 bits.

Google, además ara poder guardar y mantener segura la llave criptográfica que utiliza en cifrado/descifrado de datos (AES), la empresa utiliza un certificado de llave criptográfica pública del algoritmo RSA. Dentro de los servicios que ofrece su Plataforma de la Nube (Google Cloud Platform) utilizan el certificado de RSA.

f. NASA : La NASA es una agencia independiente del gobierno de los estados unidos; el cuál es el responsable de los programas espaciales civiles, además de las investigaciones aeronautica y aeroespaciales. NASA recomienda utilizar GPG (Gnu Privacy Guard) para poder cifrar la información importante; utilizando como algoritmo de cifrado AES con llave criptográfica de 256 bits para el cifrado/descifrado de los datos.

8. Llaves criptográficas La protección de la información cifrada es fuerte hasta que se encuentra su punto débil, la llave criptográfica. Una llave criptográfica, es una variable de valor cualquiera, la cual puede aplicarse utilizando un algoritmo, para cifrar o descifrar una sección de texto. La fuerza o dificultad de la llave criptográfica tiene que estar relacionado con el valor de la información; si la llave es comprometida, toda la información estará de igual manera comprometida, es por eso que se debe de elegir una buena llave criptográfica. (Gregory, 2015). Existe un ciclo de vida de las llaves criptográficas, en las cuales se denotan como:

a. Creación de llaves criptográficas La creación de llaves criptográficas aleatorias debe de ser elaborado dentro de un servidor seguro; para que el atacante no pueda elaborar u observar el proceso de generación de llaves. Existen dos distintos métodos o formas para poder elaborar llaves criptográficas. Aleatoridad, una llave debe de ser verdaderamente aleatoria. No Predictivas, en un sistema donde las llaves criptográficas son creadas con frecuencia; es necesario de asegurar que los valores de estas no sean predictivas. No debe estar basada en cualquier condición conocida, incluyendo los valores de llaves anteriores.

Un aspecto importante para la generación de las llaves, es el tamaño. Este determina la facilidad o dificultad para poder quebrarlo. Notese que si la llave es extremadamente larga, esta puede ser tambíen fácil de encontrar. La idea general es encontrar un tamaño de llave grande, pero no excesivo. Este tamańo es medido en bits o bytes. (Harris, 2013)

b. Protección y custodia de llaves criptográficas Acceso a llaves criptográficas debe ser controladas y estar protegidas contra la modificación o usuarios no autorizados (Harris,

2013). En algunas organizaciones que almacenan información sensible, utilizan una metodología de *división de custodia*; este determina que dos o más personas son requeridas para poder obtener la llave criptográfica, particionando un pedazo de llave para cada una. Ayuda a prevenir que una sola persona poseea la llave, y brinde acceso, alteración e incluso destrucción de la información protegida. (Gregory, 2015).

Existen métodos para proteger la llave; entre ellos se pueden mencionar: Módulos de seguridad en hardware, únicamente los usuarios con hardware determinado podrán utilizar las llaves; por ejemplo tarjetas de acceso; autenticación por medio de usuario y contraseña, por medio de autenticación del administrador; almacenamiento de las llaves en diferentes localidades físicas, las llaves criptográficas están en otra parte del mundo; fecha de expiración de llave criptográfica, expiración de llave automáticamente. (Harris, 2013)

c. Rotación de llaves criptográficas Cuando se cifra información sensible, la organización o el personal de seguridad debe de tener procedimientos y políticas formales a seguir por si una llave criptográfica está comprometida. (Gregory, 2015).

d. Destrucción de llaves criptográficas Cuando una llave criptográfica ya no es necesitada para poder cifrar; esta debe de ser eliminada de forma segura. Esta llave debe de ser eliminada de forma permanente en la misma localidad donde se creó. La información cifrada y protegida por la llave a eliminar, queda sin funcioncionalidad y es mejor eliminarla de igual manera. (Gregory, 2015).

e. Depósito de llaves criptográficas El proposito de depositar las llaves utilizando servicios terceros; es para poder tener una gran certeza que la información pueden ser recuperados. Si la organización que cifró los datos tenga un desastre con las llaves y se pierdan todas, o si la organización destruye la información incluyendo las llaves; esta pueda ser recuperada. (Gregory, 2015).

9. Métodos de ataque o criptoanálisis

a. *Exhaustive Key Search* Es uno de los ataques más simples para los criptosistemas, pero a veces es uno de los más realistas. Está designado para cifrados de bloque en cuando el criptoanálisis no pueden ser aplicados, según menciona Quisquarter y Standaert. (Quisquarter y Standaert, 1998). La estrategia básica de la busqueda de llave exhaustiva, es que a pesar de ser un algoritmo algebraico o criptoanalítico, este lo utiliza como una transformación de caja negra, donde se desconoce la información o arquitectura del algoritmo. (Kilian y Rogaway, 2000).

b. *Linear Cryptanalysis* Criptoanálisis lineal es un método de ataque que intenta conseguir probabilidades altas de ocurrencias de expresiones lineales, involucrando textos planos, bits de texto cifrado y bits de la llave. Para este algoritmo, un atacante debe de tener información de un grupo de textos originales con sus respectivos textos cifrados; pero el atacante no puede seleccionar que textos originales puede tener. La idea principal, según el ingeniero Howard es, aproximar la operacion de una parte del cifrado con una expresión que es lineal, siendo esta una operación de MOD-2 bit. Es conocido también como *Known-Plaintext attack*. (Heys, 2004).

c. *Differential Cryptanalysis* Criptoanálisis diferencial es un método de ataque que explota la gran probabilidad de obtener diferencias en las ocurrencias, de la misma manera para la encontrar diferencias en la última ronda del algoritmo criptográfico. (Heys, 2004). Es muy similar al ataque de criptoanálisis lineal. Es necesario que tenga pares de textos originales con su diferencia de entrada. Como menciona James McLaughlin, este ataque también es conocido como *Chosen-plaintext attack*, y es muy difícil de elaborar; debido a que este criptoanálisis tiene que poder obtener textos cifrados correspondientes a los pares específicos de texto original, en vez de solo textos originales aleatorios. (McLaughlin, 2015).

d. *Brute Force Attack y Dictionary Attack* Un ataque de fuerza bruta es un intento de encontrar contraseñas para usuarios conocidos, intentando desde cualquier combinación de letras, números o simbolos. Esta metodología se ha vuelto muy popular hoy en día, con la ayuda de computadoras potentes y eficientes, probandolo con una velocidad rápida, diferentes combinaciones. Con sistemas poderosos, una contraseña de 14 caracteres o menos, puede ser encontrada en menos de una semana (7 días) utilizando esta metodología. Mientras más grande es la contraseña, implica un costo y tiempo computacional más elevado, de la misma manera, cuando el número de posibilidades incrementa, incrementa de la misma manera.

Un ataque de diccionario, es un intento de encontrar cualquier posible contraseña de una lista predefinida de contraseñas conocidas o de contraseñas esperadas. Este ataque es llamada diccionario, debido a que se buscará en una gran lista de contraseñas cada una de las palabras, para poder encontrar la contraseña. De igual manera, se puede utilizar dos tipos de métodos de ataque para poder encontrar contraseñas; por ejemplo, se puede intentar un Ataque de Fuerza Bruta, utilizando un Diccionario. (Conrad *et al.*, 2012).

e. *Man-in-the-Middle Attack* Este es un ataque que permite a un intruso o personal no autorizado de obtener información a través de un canal de red de forma secreta. Este

ataque toma como ventaja la debilidad de los protocolos de autenticación dentro de los canales de comunicación. Un ejemplo que menciona Subodh Gangan, es que dos personas A y B quieren tener una comunicación a través de la red, una persona C no conocida, puede interceptar la información y obtener datos sensibles. (Gangan, 2015).

f. *Meet-in-the-Middle attack* Este es un ataque criptográfico desarrollado por Diffie y Hellman que emplea una compensación de espacio-tiempo que reduce la complejidad de craqueo a un sistema de cifrado múltiple, menciona Stephane Moore. (Moore, 2010). Meet-in-the-Middle attack es un ataque criptográfico parecido al ataque de cumpleaños; trata de encontrar un valor en el rango del dominio de composición de dos funciones, de tal manera que la imágen de la primera función dé lo mismo que la imágen inversa de la segunda función. (Ahmad *et al.*, 2010).

g. *Ciphertext only attacks (COA)* Según menciona Biryukov y Kushilevitz en su investigación, COA es un ataque donde el atacante tiene acceso a una gran cantidad de textos cifrados. No se tiene acceso a los textos originales, pero en base análisis de estos paquetes, la llave criptográfica puede ser obtenida. (Biryukov y Kushilevitz, 2007).

h. *Known plaintext attack (KPA)* Según menciona Shamir, uno de los creadores del algoritmo RSA y Zinger, KPA es un método de ataque, donde el atacante conoce los textos claros en algunas partes del texto cifrado; por lo que intenta descifrar el resto de texto cifrado con el texto original. Es también conocido como Criptoanálisis Lineal contra los algoritmos de cifrado de bloques. (Shamir y Zinger, 2012).

i. *Chosen plaintext attack (CPA)* CPA es un método de ataque donde el atacante puede obtener un texto cifrado de su elección, y en base a este buscar pares y obtener comparaciones para obtener la llave criptográfica. Es conocido también como Criptoanálisis diferencial, contra algoritmos de cifrado de bloque o funciones Hash. (Damgard y Nielsen, 2001).

j. *Side channel attack* Este ataque es uno de los más fáciles de implementar y muy poderosos para atacar a implementaciones criptográficas. Atacan principalmente a los protocolos, módulos, primitivos e incluso dispositivos dentro del sistema. La idea principal de los ataques de Side Channel Attack (SCA), es de buscar cómo el algoritmo criptográfico está implementado, que en cómo es la arquitectura del algoritmo, según mencionan los ingenieros de la academia de ciencia China. Estos ataques se llevan acabo cuando hay una correlación entre las mediciones físicas que son tomadas durante la computación: por ejemplo ola radiación, consumo de energía, consumo de tiempo y entre otro; y el estado interno del dispositivo que procesa, el cuál está relacionado con

la llave secreta. (Zhou y Feng, 2005).

k. *Timing attack* Los Timing Attack, son usualmente ataques contra dispositivos débiles como las tarjetas inteligentes. Un ataque de tiempo puede ser aplicado también a sistemas de software según mencionan los investigadores de la universidad de Stanford. Estos ataques permiten al atacante extraer secretos mantenidos dentro de un sistema de seguridad; únicamente observando el tiempo que toma el sistema para responder a diferentes consultas. Hasta ahora, estos ataques fueron solo aplicados a los tokens de seguridad físicos. Estos ataques son dificiles de aplicar dentro de servidores, debido a que los tiempos de descifrado son enmascarado por procesos concurrentes ejecutandose dentro del sistema, mencionaron los investigadores. (Brumley y Boneh, 2010).

l. *Boomerang attack* Este ataque es una derivación del de differencial attack. En el ataque diferencial, el atacante encuentra las diferencias en el texto original, el cual pueden afectar el resultado de la diferencia de la salida; texto cifrado. Ahora, el ataque de Boomerang, intenta generar una estructura de tipo cuarteto dentro de un punto en medio del camino a través del sistema de cifrado, mencionó David Wagner. (Wagner, 1999).

m. *Collision attack* Estos ataques utilizan el análisis de Side Channel Attack, para detectar colisiones internas y son generalmente no restringidas a algoritmos criptográficos particulares. Un ejemplo, según los investigadores alemanes, fue cuando se realizó un ataque de colisión contra el algoritmo criptográfico simétrico DES, el cual combinó colisiones internas con obtención no autorizada de informacion. Para los Hash criptográficos, colisión se produce cuando dos entradas de texto plano producen el mismo valor Hash de salida. (Schramm *et al.*, 2004).

n. *Birthday attack* (Ataque de Cumpleaños) Previamente llamado paradoja de cumpeaños. Este ataque fue creado bajo la suposición matemática de: en un cuarto con 23 personas, las probabilidades de que dos personas compartan el mismo día de cumplea nos, son más grandes que el 50 %. Esta puede ser explicada como; si uno está presente dentro de un cuarto con 22 personas más, hay 1/365 de probabilidad de que se comparta mi cumplea´ nos, para cada una de las 22 personas restantes dentro de la habitación, haciendo un total de 22/365 posibilidades. Ahora si se no se cumple la compartición, se deja la habitación; dejando así un compañero con 21/365 posibilidaddes de compartirlo con las personas restantes. De la misma manera para cada una de las personas dentro de la habitación. Ahora si se suma cada una de las posibilidades como $22/365 + 21/365 + 20/365 + ... + 1/365$, es más del 50 % de probabilidades. (Conrad *et al.*, 2012).

El ataque de cumpleaños es utilizado para poder crear colisiones de **hash** o **hash collision**. Este método conforme a la explicación previa, es difícil de conseguir mi entrada que colisione con otra. Pero si se compara con el método de cumpleaños, es fácil encontrarlo. Se puede deducir que encontrar una entrada cualquiera, de todas; es relativamente fácil encontrar una colisión con otra entrada. (Conrad *et al.*, 2012).

ñ. *Replay attack* Este ataque es de interpretación, y son posibles a través de capturar tráfico de red, por medio eavesdropping. Estos intentan restablecer comunicación a través de retransmitir trafico capturado, para poder seguir realizando la comunicación a través del sistema. Este puede ser prevenido utilizando autenticación una a la vez, y utilizando identificación de sesiones de secuencia. (Bruschi *et al.*, 2010).

o. *ARP spoof* El protocolo encargado de enviar cada paquete a su destino, es el protocolo Address Resolution Protocol (ARP). Este puede ser vulnerado, por lo que cualquier atacante puede monitorear o incluso modificar la información. Este ataque es utilizado para poder emplear *Man-in-the-Middle*, el cual permite interceptar mensajes entre dos victimas, permitiendo el acceso que usualmente debería estar restringido, según mencionan los investigadores de la universidad de Federico Santa María. (Pérez, 2014). Spoofing es un termino que se utiliza cuando se menciona a la suplantación de la dirección o identidad de un ordenador ajeno; el atacante puede hacerse pasar por otro dispositivo, obteniendo acceso que en condiciones normales estaría restringido. Según los investigadores, existen dos tipos de Spoofing; Spoofing Activo: el intruso interfiere con el tráfico legítimo que fluye a través de la red, y el Spoofing Pasivo: el intruso monitorea el tráfico de red. (Pérez, 2014).

10. Principios de seguridad: Seguridad en servidores Existen diferentes principios para poder brindar una buena seguridad de software. Cada uno de estos principios es un rol vital para la seguridad de las aplicaciones dentro de un servidor. La institución de estándares de los Estados Unidos menciona los siguientes.

a. Asegurar el enlace más débil la seguridad del software es fuerte, si los enlaces son fuertes. Es critico al momento de crear sistemas de software para poder brindar una seguridad fuerte y asegurarse que no existen agujeros dentro. Un ejemplo, según menciona Eugene Lebanidze es: si la información tiene una buena seguridad al momento de transferencias utilizando algoritmos de cifrado fuertes, pero dentro del repositorio o donde se almacena la información, estos son de texto claro; es más propenso que el atacante vaya por la información que está almacenada dentro del repositorio. Los atacantes intentarán comprometer cualquier posible técnica para poder

debilitar y vulnerar el enlace más débil del sistema de software. (Lebanidze, 2011)

b. Asegurar defensa en profundidad es una estrategia de defensa en capas que es requerida por si una de ellas es comprometida, otra capa pueda estar disponible para poder mitigar el ataque y prevenir de gran manera el sistema. Un claro ejemplo es en la comunicación de datos entre servidores está cifrada y el lugar de almacenamiento está también cifrado, adicionalmente se podría agregar cortafuegos para la aplicación para prevenir otros ataques. (Lebanidze, 2011).

c. Fallo en la seguridad a pesar que los fallos siempre pueden ser prevenidos, la habilidad que un sistema de software pueda manejar de forma correcta los fallos es crítica y puede prevenir distintos ataques a la aplicación de la empresa. Este principio determina que el registro de fallas del sistema no debe de estar presente para cualquier usuario, por lo que un atacante puede obtener esta información y realizar ataques en base a la falla o fallas encontradas. (Lebanidze, 2011).

d. Menor privilegio este es un principio donde un usuario o proceso solo debe de tener acceso a los privilegios mínimos necesarios para realizar las acciones determinadas. Este puede ayudar al sistema al momento de un ataque debido a que el atacante al tener acceso a este usuario con privilegios mínimos, no tiene acceso a funciones críticas que impacten al negocio, o incluso solo a funciones limitadas y no en su totalidad. (Lebanidze, 2011).

e. Simplicidad en el diseño un diseño e implementación muy compleja innecesaria puede causar distintos problemas serios de seguridad. Una buena práctica de programación y de ingeniería es crear sistemas que realicen lo que tienen que realizar de una forma simple y sin funcionalidades extras. La complejidad puede ser dificil de evaluar en el campo de la seguridad y puede brindar una alta posibilidad de ingresar errores de programación o bugs. (Lebanidze, 2011).

f. Privacidad el sistema debe de almacenar la menor información confidencial posible. Como por ejemplo no se debe de almacenar los números de tarjeta de crédito haciendo que los usuarios lo ingresen nuevamente. Mientras menos información confidencial exista dentro del servidor; menos propenso que atacantes puedan obtenerla y hacer un mal uso de ella. (Lebanidze, 2011).

11. Principios de seguridad: Seguridad en dispositivos Para poder brindar una buena seguridad dentro de los dispositivos, el instituto de estándares de los Estados Unidos elaboró una serie de principios de seguridad en el momento del desarrollo.

a. Arquitectura El diseño de la arquitectura incluye la selección de un servidor de administrador de dispositivos móviles y los clientes de software para cada uno de ellos, además de redes privadas virtuales para asegurar de mejor manera la conexión entre dos dispositivos. (Souppaya & Scarfone, 2016).

b. Autenticación La autenticación involucra seleccionar los dispositivos y los métodos de autenticación con su debidas políticas de creación, modificación y eliminación de usuarios con su debida autenticación. (Souppaya & Scarfone, 2016).

c. Criptografía Para determinar el uso de la criptografía dentro de und ispositivo, es necesario incluir la selección de los algoritmos de cifrado y protección de integridad en la comunicación entre dispositivos. Adicionalmente es necesario crear políticas de las llaves criptográficas como la creación, rotación y eliminación de las mismas, además del conocimiento de los requerimientos mínimos para los algoritmos criptográficos. (Souppaya & Scarfone, 2016).

d. Configuración Para este principio, es necesario establecer estándares de seguridad mínimos para los dispositivos móviles; como la aplicación de parches y actualizaciones para mitigar vulnerabilidades, además de controles adicionales como redes privadas virtuales y certificados de seguridad. (Souppaya & Scarfone, 2016).

12. Herramientas de ataque

a. Wireshark : Wireshark es un analizador de protocolos *open-source* diseñado por Gerald Combs y que actualmente está disponible para plataformas Windows y Unix. Conocido originalmente como Ethereal, su principal objetivo es el análisis de tráfico, además de ser una excelente aplicación didactica para el estudio de las comunicaciones y para la resolución de problemas de red. Wireshark implementa una amplia gama de filtros que facilitan la definición de criterios de búsqueda por más de 1100 protocolos actualmente, según menciona Borja Merino Febrero. (Merino, 2011). Wireshark tiene la habilidad de poder realizar ataques de ARP Spoofing, donde un atacante puede interponerse entre una o varias máquinas para poder interceptar, modificar o capturar paquetes. Es una técnica muy intrusiva, pero puede ser detectada al momento de la cantidad de tráfico que se ingresa dentro de la red. (Merino, 2011).

b. Arpspoof ARPSpoof es una técnica usada comúnmente por atacantes en redes internas para ataques de Man-in-the-Middle o para explotar algún fallo en la victima para tener acceso al equipo. Address Resolution Protocol (ARP) es un protocolo de red encargado para resolver direcciones IP y MAC, según menciona Ignacio Pérez. (Pérez, 2014). Esta herramienta puede encontrarse dentro del sistema operativo Kali Linux. (Valentino, 2013).

c. FindMyHash Findmyhash es una herramienta desarrollada en python, y utiliza un total de 48 servicios online de crackeo de contraseñas para intentar averiguar texto cifrado a partir de un archivo o texto Hash que hayamos podido recuperar. Actualmente soporta distintos algoritmos de Hash: MD4, MD5, SHA1, SHA256, RMD160, LM, NTLM, MYSQL, CISCO7, JUNIPER. Según las pruebas que han realizado los desarrolladores, el éxito de encontrar la contraseña a partir de un hash LM/NTLM es de un 60 % a 70 %, según menciona Juan Antonio. (Antonio, 2011).

C. Almacenamiento de información y servicios web

1. Base de datos Una base de datos se puede definir como una gran cantidad de datos almacenados en una o varias computadoras. El software que maneja esta información es conocido como sistema gestor de base de datos. Para los propósitos de este proyecto es necesario contar con uno o más gestores de bases de datos que cumplan con los siguientes requerimientos

- manejo de volumen de información
- velocidad de consulta de información
- integridad de información

(Abiteboul, 1995)

Los principales modelos de interés de este trabajo son SQL y NoSQL. SQL (Structured Query Language) nació de la necesidad de establecer un lenguaje estándar para hacer consultas sobre la información almacenada. Para trabajar sobre una base de datos SQL es necesario primero definir su estructura utilizando el lenguaje DDL (Data Definition Language) y luego manipular su contenido con el lenguaje DML (Data Manipulation Language). Las características distintivas de las bases de datos SQL son las siguientes:(Structured Query Language, 2016)

- Cumple con las características ACID (Atomicidad, Consistencia, Persistencia, Aislamiento).
- Escalabilidad vertical, aunque utilizando Hadoop puede comportarse de acuerdo a una escalabilidad horizontal.
- Poca flexibilidad

Los DBMS (Sistema gestor de base de datos por sus siglas en ingles) SQL más populares son Oracle, MySQL, Microsoft SQL server y PostgreSQL. A continuación se analizan las ventajas y desventajas de cada uno. (DB-Engines, 2016)

Cuadro 6: Ventajas y desventajas del DBMS MySQL

MySQL	
Ventajas	**Desventajas**
Facilidad de uso	Problemas de estabilidad: corrupción tablas, consumo desmedido de CPU
Soporte por parte de la comunidad	Problemas con alta concurrencia
Open Source	Oracle no está interesado en continuar su desarrollo
Bajo costo	Depende de extensiones para llenar los requerimientos de ACID
Es un estándar de la industria	No cumple al 100 % con el estándar SQL
Portabilidad	

(Russ, 2014)

Cuadro 7: Ventajas y desventajas del DBMS Oracle

Oracle	
Ventajas	**Desventajas**
Cumple con los requerimientos ACID	Precios elevados de licenciamiento
Manejo de múltiples bases de datos utilizando el 2 phase commit	Complejidad técnica alta
Soporte por parte del vendedor	Consumo alto de recursos, orientado a servidores dedicados
Informes de actualizaciones y nuevas funcionalidades	
Portabilidad	
Buen rendimiento bajo grandes cantidades de datos	

(System Properties Comparison, 2016)

Cuadro 8: Ventajas y desventajas del DBMS SQL Server

SQL Server	
Ventajas	**Desventajas**
Rendimiento	Las actualizaciones pueden requerir grandes cambios en el sistema.
Con respecto a oracle, tiene una mayor comunidad	Por el momento solo está disponible para servidores con sistema operativo Windows
Soporte por parte del vendedor	Costos de licenciamiento altos
Cumple con los requerimientos ACID	

(Jones, 2012)

Cuadro 9: Ventajas y desventajas del DBMS PostgreSQL

PostgreSQL	
Ventajas	**Desventajas**
Tipos de datos flexibles	Replicación de base de datos relativamente nueva, puede tener errores.
Permite el uso de otros lenguajes dentro de la base de datos: python, ruby, R, v8	El soporte por parte de los desarrolladores y la comunidad no es tan amplio como en los otros DBMS analizados.
Utilizando hstore se tiene acceso a algunas de las ventajas que ofrecen algunas bases de datos NoSQL.	La mayoría de aplicaciones open source no soportan el uso de postgreSQL.

(System Properties Comparison, 2016)

El término NoSQL, se utiliza generalmente para referirse a bases de datos no relacionales. Existe una amplia variedad de tipos de bases de datos NoSQL, a continuación se listan las más importantes: (A Comparison, 2014)

- **Bases de datos de documentos:** se asocia una llave con una estructura de datos compleja conocida como documento. La complejidad del documento facilita la transparencia entre objeto y datos bajo el paradigma de programación orientada a objetos. Un ejemplo es Mongodb. (A Comparison, 2014)
- **Bases de datos utilizando llave/valor:** este tipo de base de datos se basa en asociar una llave con un valor. El funcionamiento es similar al de un diccionario y en su mayoría se utiliza para guardar información básica. Un ejemplo es Redis. (A Comparison, 2014)
- **Bases de datos basada en columnas:** tiene un funcionamiento similar al de una base de

datos llave/valor, pero existen varias parejas llave/valor para recrear lo que sería una fila. También se compara frecuentemente una base de datos basada en columnas con una arreglo de dos dimensiones donde cada elemento del arreglo contiene una lista de valores. Algunos ejemplos son Cassandra y HBase. (A Comparison, 2014)

- **Bases de datos de grafos o redes:** utilizadas en contextos donde los sistemas de información presentan entidades, y las relaciones entre ellas, que están mejor representadas por grafos. Algunos ejemplos son Neo4J y Giraph. (A Comparison, 2014)

Las características distintivas de las bases de datos NoSQL son:

- Escalabilidad horizontal
- Flexibilidad
- Transparencia en paradigmas de programación orientada a objetos.

A continuación se mencionan los casos de uso más populares para los distintos tipos de DBMS NoSQL.

Cuadro 10: Casos de uso de los distintos tipos de Base de datos

Caso de uso	**Descripción**
Llave/valor	▪ Guardar valores por un tiempo limitado y para uso futuro frecuente, en otras palabras **caching**. ▪ Guardar listas de espera y conjuntos de datos. ▪ Guardar información del estado de una aplicación.
Basado en columnas	▪ Guardar colecciones grandes de información por un tiempo prolongado. ▪ Escalabilidad al manejar grandes volúmenes de información no estructurada.
Orientada a Documentos	▪ Guardar información anidada. ▪ Facilidad de uso con documentos Json.
Grafos	▪ Manejo de información relacional compleja. ▪ Trabajo sobre grados de separación entre nodos de datos. ▪ Clasificación de información.

(Use cases, 2011)

Para proyectos simples es común que la información de la base de datos este almacenada en una computadora accesible dentro de una red. Pero conforme la cantidad de información a almacenar crece la capacidad de una computadora para almacenarla se ve limitada. Para ello se crearon los manejadores de bases de datos distribuidos (DDBS), estos se encargan de distribuir la información dentro de una red a manera de repartir la carga del almacenamiento, además de replicar la información a manera de reducir el riesgo de pérdida de información. (Phanouriou, 1995) Una de las herramientas más populares para distribuir información dentro de una red es Hadoop. Hadoop es un framework de código abierto escrito utilizando el lenguaje de programación Java. Hadoop tiene la capacidad de escalar desde una sola máquina hasta miles de servidores. Existen muchos proyectos relacionados con Hadoop, dentro de estos existen varios manejadores de bases de datos distribuidos. (Apache, 2016)

Uno de los DDBS utilizados para manejo de grandes volúmenes de datos es HBase. HBase es un manejador de base de datos NoSQL de código abierto. Tiene la capacidad para trabajar con

billones de filas y millones de columnas mientras su rendimiento se mantiene rápido. Una de las mayores ventajas de HBase es que es tolerante a fallos gracias a que replica la información en distintos servidores de la red. Esto permite reponerse a un fallo en un servidor ya que HBase puede traer la información almacenada de réplicas en otros servidores. (Horton works, 2016)

Antes de configurar las tablas, nodos o documentos, que lleva una base de datos es necesario modelar el sistema de información. Una práctica común es utilizar un Diagrama Entidad Relación. El Diagrama Entidad Relación representa a la realidad a través de un esquema grafico empleando: (El modelo Entidad-Relación, 2001)

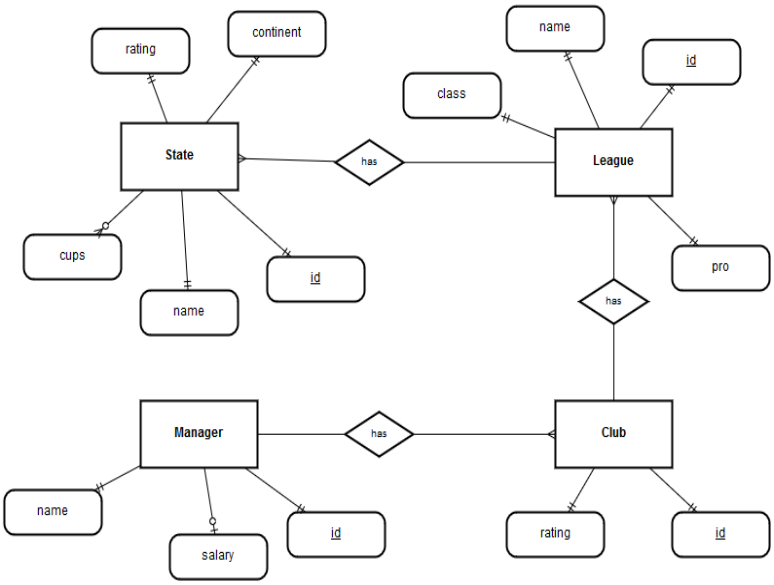
- Entidades: Las entidades son conjuntos de elementos presentes o no en el contexto del sistema de información que se desea representar.
- Relaciones: Las relaciones son vínculos que permiten definir una dependencia entre una o más entidades.
- Atributos: Son los rasgos, propiedades y características del elemento que representa la entidad.
- Claves: Campo o atributo de una entidad que tiene como objetivo distinguir cada registro de la entidad.
    - Claves primarias: Permiten identificar cada registro de una entidad como único.
    - Claves externas: Es un campo clave conformado por el valor de una clave primaria de otra entidad. Permite establecer una relación entre entidades.

En la siguiente figura se muestra un ejemplo de diagrama ER:

El diagrama entidad relación generalmente se utiliza para sistemas de información que utilizaran una base de datos SQL. En el caso de las bases de datos NoSQL no existe una metodología general para modelar los sistemas de información. Para modelar un sistema que utilizara NoSQL se vuelve necesario utilizar esquemas específicos para el tipo de NoSQL que se está utilizando. En algunos casos se utilizan diagramas entidad relación, a pesar de no estar trabajando con bases de datos relacionales, a manera de representar la relación entre las colecciones de información. (Katsov, 2012)

2. Herramientas de desarrollo Para el acceso a los datos y su manipulación comúnmente se utiliza una Interfaz de programación de aplicaciones (API). Esto permite ofrecer servicios y recursos en forma de funciones para facilitar la comunicación con otros sistemas o módulos independientes de un mismo sistema. Para desarrollar un sistema no es necesario construir un API, pero el no hacerlo implica tener que proporcionar acceso directo a la base de datos para cada sistema con el que se desee trabajar. (IBM, 2015) Para facilitar la construcción de una API se puede utilizar un framework para acortar el tiempo de desarrollo. Un framework es una

Figura 2: Ejemplo de diagrama entidad relación



estructura conceptual y tecnológica de soporte definido, normalmente contiene módulos concretos de software que sirven como base para el desarrollo de un software específico. Antes de analizar algunos frameworks es necesario definir algunos conceptos:

- ORM: Es un mecanismo que hace posible el acceso y manejo de objetos sin tener que preocuparse de cómo estos objetos se relacionan con una base de datos. (Hibernate - ORM Overview, 2016)
- Microframework: Es un término utilizado para referirse a frameworks minimalistas. Los frameworks minimalistas no ofrecen todas las funcionalidades que normalmente se esperarían de un framework, funcionalidades como: ORM, validación de entradas, etc. (Eguiluz, 2015)
- REST (Representational State Transfer): Es un conjunto de principios que definen la interacción entre distintos componentes. A continuación se listan las restricciones de la arquitectura REST:
  - Arquitectura cliente-servidor: Consiste en una separación clara y concisa entre los 2 agentes básicos en un intercambio de información. Estos agentes son el cliente (el que solicita la información) y el servidor (el que provee la información).
  - Protocolo sin estado: Toda la información necesaria para responder a una petición se encuentra dentro de la misma solicitud.
  - Cacheable: El servidor debe definir alguna forma de cachear las respuestas de manera que se pueda aumentar el rendimiento, escalabilidad, etc.

- Sistema por capas: El cliente no conoce si quien está ejecutando su petición es el servidor o un intermediario.
- Interfaz uniforme

(Fernández, 2013)

- RPC (Remote Procedure Call): Antes de que la arquitectura REST la mayoría de APIs eran construidas utilizando la arquitectura RPC. RPC es similar en algunos aspectos a REST, como en el hecho de que ambos utilizan una arquitectura cliente-servidor, pero donde más difieren es en el formato de peticiones al servidor. En REST se utiliza una combinación de URL y métodos de la petición http (GET, POST, PUT, DELETE, etc.) para determinar que método se ejecutará, en cambio en RPC el método a ejecutar se determina únicamente por la URL. Otra diferencia importante es que RPC recibe los parámetros para la ejecución del método en el cuerpo de la petición, en cambio REST recibe algunos parámetros de la URL de la petición, como identificadores, y otros en el cuerpo de la petición. (Sturgeon, 2016)

A continuación se listan las ventajas y desventajas de los frameworks de desarrollo más populares.

Cuadro 11: Ventajas y desventajas del Framework Flask (python)

Flask	
Ventajas	**Desventajas**
Microframework	Generalmente se requiere más trabajo, en relación a frameworks de desarrollo web completos, para implementar las distintas funcionalidades.
Rendimiento mayor en relación a otros frameworks, como Django.	Comunidad de desarrollo pequeña
Menor cantidad de código en relación a otros frameworks.	

(Parker, 2013)

Cuadro 12: Ventajas y desventajas del Framework Bottle (python)

Bottle	
Ventajas	**Desventajas**
Microframework	Generalmente se requiere más trabajo, en relación a frameworks de desarrollo web completos, para implementar las distintas funcionalidades.
Rendimiento mayor en relación a otros frameworks, como Django.	Comunidad de desarrollo pequeña
Flexibilidad	
Facilidad en la creación de API Web	

(Parker, 2013)

Cuadro 13: Ventajas y desventajas del Framework Jersey

Jersey	
Ventajas	**Desventajas**
Microframework	Generalmente se requiere más trabajo, en relación a frameworks de desarrollo web completos, para implementar las distintas funcionalidades.
Soporte de modelo vista controlador.	Si es necesario utilizar software de terceros para facilitar el desarrollo es posible encontrarse con que no es compatible con la versión actual.
Soporta conexiones asíncronas.	
Rápido en comparación de los frameworks python mencionados.	
Documentación extensa	

(Top 8 Java RESTful, 2015)

Cuadro 14: Ventajas y desventajas del Framework Restlet

Restlet	
Ventajas	**Desventajas**
Microframework	Generalmente se requiere más trabajo, en relación a frameworks de desarrollo web completos, para implementar las distintas funcionalidades.
Fue desarrollado con el objetivo específico de crear REST apis.	Perdió popularidad por el uso de Jersey y Play Framework
Modularidad	Curva de aprendizaje alta
Todavía se encuentra en desarrollo.	

(Top 8 Java RESTful, 2015)

Cuadro 15: Ventajas y desventajas del Framework Express

Express	
Ventajas	**Desventajas**
Curva de aprendizaje baja	Es necesario repetir código que podría estar incluido en el framework.
Documentación extensa	Refactorizar el código es complicado.
Se ajusta a las necesidades de cada proyecto	Las pruebas deben ser extensas ya que no se provee ninguna facilidad para realizarlas.

(Gorbatchev, 2014)

Cuadro 16: Ventajas y desventajas del Framework Restify

Restify	
Ventajas	**Desventajas**
No tiene funcionalidades que no sean necesarias.	Es necesario repetir código que podría estar incluido en el framework.
Regulación de tráfico de consultas a la web API	Refactorizar el código es complicado.
	Las pruebas deben ser extensas ya que no se provee ninguna facilidad para realizarlas.

(Gorbatchev, 2014)

Cuadro 17: Ventajas y desventajas del Framework loopback

loopback	
Ventajas	**Desventajas**
Documentación generada de forma automática.	Comunidad pequeña.
Framework enfocado en el desarrollo de REST APIs.	El soporte en su mayoría es de tipo comercial.
Facilidad para modificar el esquema de la base de datos, también facilita el cambio de DBMS.	Curva de aprendizaje alta.

(Gorbatchev, 2014)

3. Herramientas para realización de pruebas Para poder medir cómo se comporta un servicio web bajo distintos niveles de demanda se utilizan las pruebas de carga. El objetivo de las pruebas de carga es obtener el tiempo de respuesta, la cantidad máxima de usuarios y la utilización de recursos bajo un nivel definido de demanda. Para la realización de pruebas de carga la herramienta JMeter es una de las más populares. JMeter es una herramienta de software de código abierto desarrollada utilizando Java. Una de las facilidades que provee JMeter es el diseño de pruebas de rendimiento y carga utilizando una interfaz gráfica. Otra característica importante de JMeter es que permite exportar los resultados de las pruebas en un archivo separado por comas (csv) compatible con herramientas de análisis. (Stevens, 2010) Existen varias alternativas para pruebas de carga y rendimiento, una de ellas es BlazeMeter. BlazeMeter además de ofrecer servicios de pruebas de carga y rendimiento ofrece servicios de análisis de datos gratuitos. Estos servicios pueden utilizar los archivos csv que JMeter crea como salida para realizar su análisis. (BlazeMeter, 2016)

The Grinder es otra herramienta para la realización de pruebas de carga. Al igual que JMeter, The Grinder es una herramienta de código abierto escrita en Java. Posee muchas de las características que JMeter ofrece pero una de sus especialidades es la creación de información de prueba a partir de archivos y/o bases de datos. Esta herramienta también puede realizar pruebas con múltiples protocolos, como TCP, UDP, HTTP, etc. (Open Source Load Testing Tools, 2015)

Aunque muchas de las herramientas de prueba fueron desarrolladas utilizando el lenguaje de programación Java existen alternativas, una de ellas es Tsung. Tsung es una herramienta de código abierto, escrita en el lenguaje de programación Erlang, utilizada en pruebas de rendimiento y carga. El principal beneficio de esta herramienta es su capacidad de proveer métricas de desempeño tanto para el cliente como el servidor. Esto permite realizar distintos análisis útiles, como: le uso de CPU del servidor bajo una cantidad específica de transacciones por segundo.(Open Source Load Testing Tools, 2015)

D. Integración

Integrar está definido por la Real Academia Española como: “Aunar, fusionar dos o más conceptos, corrientes, entre otros., divergentes entre sí, en una sola que las sintetice” (Real Academia Espñola, 2014). Esto da la base para la definición de integración de un sistema. La integración de un sistema es unir nuevos o existentes sistemas y tecnologías para crear un sistema más con mejores o más funcionalidades que el sistema actual (Sage & Rouse, 1999).

La integración tiene dos objetivos principales: lograr que todas las aplicaciones trabajen con la misma información y mejorar la eficiencia al evitar procesos innecesarios y errores. Al realizar la integración de sistemas e intentar de cumplir estos objetivos se presentan distintas ventajas y desventajas. Entre las ventajas se encuentran: acceso a información entre diferentes sistemas o módulos, aumento de eficiencia en términos de recursos y tiempo, integridad de la información y control de acceso a la información. Entre las desventajas esta el agregar complejidad al proyecto y en algunas circunstancias resulta difícil hacer el sistema flexible y con facilidad de crecimiento (s.a., 2011).

La integración se puede realizar cuando se tiene un sistema terminado o se puede realizar de manera iterativa conforme se esta desarrollando el sistema. Existe una metodología que se puede utilizar como guía para este proceso interativo. Esta metodología inicia al dividir cada subsistema en partes más pequeñas. Entre estas partes puede existir dependencia entre ellas, aunque lo ideal es que sean independientes. Si existe dependencia entre estas partes, se puede realizar una definición que dicta como estas partes trabajan juntas.

Esta metodología requiere que durante el proceso de creación del producto final se tome en cuenta la integración. El objetivo de tomar en cuenta la integración desde el inicio del producto es hacer la integración del sistema predecible y más sencilla (Mellon, 2009).

Existen diversas prácticas utilizadas en la integración iterativa. Una de ellas son los lenguajes de interfaz. En esta metodología se utilizan lenguajes que permiten definir una interfaz entre el cliente y el servidor (McKinnon, s.f.). Aqui se definen el nombre de los métodos y sus parámetros. Esto se puede utilizar para separar el desarrollo de la integración, ya que se puede hacer siguiendo lo que indica la interfaz. Wrapping es otra metodología que crea un programa que va a ser intermediario entre la interfaz del componente que se espera y la interfaz que usa el componente (Mellon, 2009). Por último, middleware se define como una capa en la que se provee una manera de comunicarse entre interfaces de métodos con repositorios o generadores de datos. Esto conecta información y una interfaz (Bakken, s.f.).

Cuando se piensa en la integración de un sistema o proyecto, utilizando o no la metodología iterativa, se deben tomar en cuenta los desafíos que se presentan realizando la integración.

Uno de los desafíos más comunes de usar redes es que estas no son fiables. Comúnmente la integración suele suceder de una computadora hacia otra a través de redes. Esto crea un problema

ya que esta comunicación puede crear retrasos o interrupciones. Diseñar una solución distribuida en varias redes causa que las llamadas a métodos sea lenta, porque la red en que se encuentra esta solución es lenta. La integración debe ser capaz de transmitir información entre diferentes lenguajes de programación, sistemas operativos y formatos de información. Esto trae consigo el problema que los sistemas constantemente cambian, es por esto que la integración debe minimizar las dependencias entre sistemas. Existen diversas soluciones a estos desafíos.

Una solución es la transferencia de archivos. Esto significa que la aplicación escribe un archivo que es leído por otra aplicación. Se debe establecer donde se guarda el archvo y el format del archivo. Esto propone compartir información por archivos. Otro acercamiento es base de datos compartida, la cual propone que las aplicaciones compartan un esquema de base de datos y esté centralizada.

Para compartir información también existe mensajería que consiste en publicar un mensaje en un canal para que sea recibido por la otra aplicación. Esto se enfoca en compartir información. Invocar procedimientos remotamente permite compartir información y realizar proceso de otra aplicación. Este procedimiento implica que una aplicación muestra sus funcionalidades para que puedan ser accedidas remotamente (Hohpe & Woolf, 2004).

Integración remota se le llama cuando se transporta la información a través de internet. Esta integración se puede realizar a través de distintas técnicas. Entre las técnicas más relevantes hoy en día se encuentran: CORBA, XML-RPC y SOAP.

Common Object Request Broker Architecture, conocido como CORBA, esta diseñada en torno a objetos que pueden ser utilizados en diferentes ambientes. Uno de los mayores beneficios de esta técnica es que permite ser independiente del lenguajede programación o plataforma que se este utilizando. Esto permite envolver el objeto con una serie de reglas e instrucciones de ejecución escritas en CORBA, para que se ejecuten en otro ambiente. El problema con CORBA es que puede llegar a tener reglas muy complejas, por lo que su utilización puede ser compleja (Liu, s.f.).

XML-RPC es un protocolo que utiliza XML (Extensible Markup Language). En este protocolo se describen métodos a ejecutar, sus resultados y cómo deben ser implementados. Esto permite ejecutar métodos en una máquina haciendo una solicitud por internet. XML-RPC define en un XML los métodos. Luego se crea una solicitud HTTP (Hypertext Transfer Protocol) utilizando el método POST, la otra máquina reciba la solicitud, busca el método lo ejecuta con la con la información recibida. Por último se envía el resultado del método (Slonneger, 2006).

SOAP (Simple Object Access Protocol) es una técnica que se utiliza mucho para la utilización de servicios web. Esta técnica también utiliza HTTP y XML para transmitir información entre máquinas. Entre sus beneficios se encuentra que puede utilizar diferentes métodos de HTTP, haciéndolo un poco más flexible; pero es considerado más lento que los anteriores mencionados (Papazoglou, 2008).

Para transmitir información de manera estructurada existen distintos tipos de formatos. Coma ya se mencionaron anteriormente, XML es uno de estos formatos. XML es un formato basado en texto que se utiliza para representar información estructurada. Este formato permite la creación de representaciones más complejas pero conlleva más carácteres y es más lento en términos de transmición de datos comparado a JSON (Quin, 2015).

Uno de los formatos más comunes es JSON (JavaScript Object Notation). JSON es un formato liviano para intercambio de datos. Este formato esta basado en el lenguaje de programación Javascript. Este formato tiene dos estructuras una colección de nombres y valores, llamados objetos; una lista de diversos valores, estos puede ser objetos o solo valores como enteros, cadenas o decimales. Este formato es conocido por ser más rápido que XML y utiliza menos recursos para procesar (Nurseitov *et al*, s.f.). Un formato que ya no se utiliza tanto, por las limitantes que tiene, es CSV (Comma Separated Values). Este formato de archivo representa de manera sencilla datos en forma de tabla. En este formato se tienen las columnas separadas por comas o punto y coma, y las filas separadas por salto de líneas (Shafranovich, 2005).

Ya se mencionó qué es la integración remota y cómo se puede llevar a cabo, IBIDS es un marco de referencia para diseñar estos sistemas de integración remota. Con IBIDS se deben tomar en cuenta tres dimensiones cuando se quiere realizar la integración de un sistema. Estas dimensiones son: control, información e interfaz del usuario. La dimensión de control se encarga de la habilidad de comunicación entre las diferentes partes a integrar. La dimensión de la información se encarga de manejar la información que cada componente crea o que cada componente tiene. La dimensión de interfaz de usuario es como se van a presentar las diferentes herramientas a los usuarios (Usha *et al*, s.f.). La integración remota es un tipo de integración para sistemas que se comunican por internet. Mensajería es otro tipo de integración que puede llegar a utilizar integración remota. La mensajería en la integración se encarga del envío de mensajes, así como de la comunicación de aplicaciones. Un ejemplo de mensajería son los mensajes de voz. Alguien deja un mensaje de voz y el receptor escucha el mensaje cuando tenga la oportunidad. Este tipo de comunicación permite que quien envía el mensaje puede despreocuparse del mensaje, ya que solo envía el mensaje y el sistema que guarda el mensaje de voz se encarga de mantenerlo ahí hasta que el receptor lo escuche.

A un nivel más técnico, la mensajería es una tecnología que permite la comunicación entre programas con entrega confiable. Esta comunicación se realiza a través de mensajes, y estos mensajes se envían por canales. Estos canales son caminos que comunican programas y llevan mensajes. Este canal es el encargado de guardar los mensajes y compartirlos a través de varias aplicaciones. En la mensajería también se tiene un remitente y receptor, que son quienes envían, reciben y leen el mensaje.

La mensajería usualmente es manejada por message-oriented middleware (MOM). Para que la mensajería funcione se debe tener configurado los canales que definen el camino de los mensajes.

MOM se encarga de administrar los mensajes dentro del sistema y distribuirlos dentro del canal.

Su tarea principal es asegurar que se transmitan correctamente los mensajes, manejando los diferentes casos correctamente. La transmisión de un mensaje se puede dividir en los siguientes pasoso.

- El remitente crea el mensaje, llena el mensaje con información
- El remitente agrega el mensaje al canal
- El sistema de mensajería transmite el mensaje desde el remitente al receptor
- El receptor lee el mensaje desde el canal
- El receptor procesa la información encontrada en el canal

En estos pasos se debe tomar en cuenta que los mensajes deben ser enviados y olvidados. El remitente debe agregar a una cola el mensaje. El canal obtiene el mensaje de la cola. Para leer el mensaje el receptor se debe suscribir al canal y una vez entra el mensaje al canal, el receptor que esta escuchando el canal puede proceder a leer el mensaje. Este proceso permite que el remitente y receptor sean independientes. El remitente puede enviar el mensaje y olvidarse de él porque el canal mantendrá el mensaje hasta que el receptor lo lea; por otro lado, el receptor no esta esperando que el remitente deposite el mensaje, él lo lee cuando pueda.

Es importante que el mensaje pase por el canal para independizar el remitente del receptor. El canal permite que si el remitente o receptor cambia, el canal sigue siendo el mismo, lo que implica que el otro lado de la comunicación sigue siendo igual. Por ejemplo, si el remitente cambia, el receptor sigue recibiendo el mensaje del canal para luego leer y procesar información.

Se ha mencionado mucho la mensajería y como funciona, pero falta por abarcar el por qué. La mensajería es más inmediata que la transferencia de archivos, mejor encapsulada que las bases de datos compartidas y es más confiable que invocación remota de procedimientos. La mensajería permite comunicar diferentes lenguajes de programación haciendo más flexible y más sencillo su crecimiento entre aplicaciones. Esto lo hace a través de ser un traductor para las diferentes aplicaciones. También permite enviar la información y olvidarse del mensaje, independizando las aplicaciones y permitiendo que siga trabajando la aplicación del remitente en otras tareas (Hohpe & Woolf, 2004).

La mensajería se puede utilizar a través de patrones de integración. Un patrón es una solución que se da constantemente a un problema en un ambiente determinado. En el ámbito de la integración existen diferentes patrones que se pueden utilizar para llevar a cabo la integración. Un patrón no es necesariamente una solución que siempre funciona, es más una guía de lo que se puede hacer (Universidad de Sevilla, 2012).

Entre los patrones de integración se encuentra el canal de mensaje. Este es el lugar donde se almacena el mensaje antes de que un componente lea el mensaje (Universidad de Sevilla, 2012). Estos canales se conectan a algún componente. Un componente es un elemento del sistema, el cual tiene interacción con la información que se transmite (Hohpe & Gregor, 2015).

Aunque usualmente los canales se conectan a componentes, pueden conectarse a un extremo de mensajería (Endpoint). Un endpoint es un elemento el cual conoce poco sobre la aplicación en la que se encuentra. Este elemento solo sabe que espera un mensaje y/o debe enviar un mensaje a algún otro elemento (Universidad de Sevilla, 2012). Mensaje es la información que se transmite (Universidad de Sevilla, 2012). En muchas ocasiones los mensajes pasan por un Router. El router no modifica el mensaje, solo se encarga de conectar distintos elementos, por ejemplo, mensaje con un canal, mensaje con un endpoint, entre otros (Hohpe & Gregor, 2015).

En distintos escenarios el router envía un mensaje con un formato a otro lugar donde se espera otro formato, para esto existe el traductor. Este elemento es el encargado de convertir la información de un formato a otro para que otro contexto o ambiente lo pueda utilizar (Universidad de Sevilla, 2012). También existe el conector que son flechas son utilizadas para mostrar la dirección del flujo de la información. Es un elemento que une dos elementos e indica hacía donde va la información (Hohpe & Gregor, 2015). En el anexo A se muestran las figuras con la representación gráfica de los patrones mencionados anteriormente. Para el desarrollo de una plataforma de integración que tome en consideración los patrones se tomaron en cuenta diversas herramientas.

Apache Camel es una plataforma de integración que aplica o incluye dentro de su ambiente patrones de integración para empresas. Con esta plataforma se pueden definir reglas para el ruteo y reglas de mediación en una gran variedad. Unas de las mayores ventajas de esta plataforma es su capacidad de trabajar con otras tecnologías, por ejemplo, puede trabajar directamente con ActiveMQ, JMS, JBi, entre otras. Esto se debe a que es una plataforma open-source. Esto permite que tenga una fuerte comunidad y tenga varias herramientas ya desarrolladas. A pesar de poder trabajar con estas tecnologías también se le pueden agregar otros componentes para abarcar más formatos de información (The Apache Software Foundation, 2016).

Otra plataforma que se puede utilizar para integración es Spring Integration. Spring Integration se puede ver como una extensión del modelo de programación de spring. Spring ayuda a los desarrolladores construir código simple, rápido y utilizando JVM (Java Virtual Machine) para los sistemas y aplicaciones. El objetivo de Spring Integration es proveer un modelo con el cual se puedan implementar soluciones que sigan los patrones de integración de empresas. Spring provee este modelo y también toma en consideración la parte de las pruebas y que el mantenimiento del código sea factible (Pivotal, 2016). Mule es una plataforma de mensajería, es un Bus de Servicio Empresarial (en inglés Enterprise Service Bus ESB). Mule permite la comunicación de tecnlogías a través de objetos. Estos objetos se comunican entre tecnlogías a través de servicios y por eso Mule

es un bus de servicios. Este bus de servicio esta basado en Java. Un ESB puede comunicar diferentes plataformas y también puede orquestar eventos de diferentes plataformas. Una de las ventajas de Mule es que permite la comunicación de plataformas o tecnologías siendo nada más un sistema de transporte. Las principales capacidades de Mule son: creación de servicios y hosting, mediación de servicios, ruteo de mensajes y transformación de datos (Mulesoft Inc, 2016). Aparte de estas fuertes herramientas se tomó en consideración una herramienta enfocada a mensajería, ActiveMq. Apache ActiveMq es un servidor de mensajería y para patrones de integración. ActiveMq es popular y es compatible con diversos lenguajes de programación. Esto lo hace una muy fuerte a utilizar cuando se piensa en integración (The Apache Software Foundation, 2016).

E. Análisis de datos

El análisis de datos se puede entender como el proceso de aplicar de forma sistemática técnicas que involucran áreas como la estadística y la lógica, con el fin de describir, ilustrar, condensar, resumir y evaluar un conjunto de datos. (Center, 2005) De igual manera se podría decir que se considera el examinar los datos de maneras que permiten revelar la relación, patrones y tendencias dentro de estos. Por ello se tendría que someter estos datos a operaciones estadísticas con las cuales se pueden determinar las relaciones que parecen existir entre las variables analizadas, además de saber cuál es el nivel de confianza de la respuesta que se está obteniendo (Development, 2015).

Dentro de esta área diversos lenguajes de programación son usados de forma común, esto se debe no sólo a la cantidad de librerías y soporte de parte de la comunidad sino también a las facilidades que ofrece el lenguaje para llevar a cabo las tareas relacionadas con el análizar los datos. En este grupo se pueden mencionar a lenguajes como Julia, R y Python (Nicolaou, 2014), siendo estos los preferidos a la fecha en que se ha desarrollado esta investigación.

- **Julia:** Fue creado en 2009 por Jeff Bezanson, Stefan Karspinki y Viral B Shash, este cuenta con una forma de exploración de datos interactiva y además es bastante útil para el desarrollo de prototipos de algoritmos, cuenta además con la ventaja de poder escalar los prototipos realizados en el lenguaje para que puedan manejar una mayor cantidad de dato; dado que este es un lenguaje de programación relativamente nuevo su comunidad ha ido creciendo de buena manera aunque aún necesita de mejoras en sus diveros paquetes (Kelman, 2015). Se ha dicho que Julia puede producir código tan eficiente como el de C, esto soalmente si el código a compilar está hecho pensando en el funcionamiento (White, 2013).

- **R:** Este nacio en 1997 siendo una alternativa para lenguajes como SAS o Matlab. Al pasar el tiempo fue ganando popularidad en áreas estadísticas, esto puede deberse gracias a con este se puede trabajar con una gran cantidad de datos complejos de forma sencilla, además que permite la manipulación de los datos usando funciones sofisticadas, elegantes y además tiene la posibilidad de crear gráficos, todo esto en pocas líneas. Cuenta con una gran comunidad

que ha colaborado en el desarrollo de un ecosistema bastante completo, llegando a tener ás de 2 millones de personas que emplean R, siendo también uno de los más populares para trabajar con datos, seguido por Python (Nicolaou, 2014).

- **Python:** Nació en 1991 creado por Guido van Rossum. Este ha ganando popularidad en el ánalisis de datos pues es bastante sencillo de aprender además que permite la manipulación de datos rápida y sus capacidad para trabajar con muchos datos. Cuenta con una gran comunidad que ha hecho que gane popularidad para el análisis estadístico, quitando parte de su poder en el mercado a R. De igual manera, en este se pueden crear productos que no sólo son sofisticados sino también escalables (Nicolaou, 2014). Una de las más grandes fortalezas de Python es la comunidad que los respalda, haciendo herramientas y librerías que ayudan en las diferentes tareas que envuelve el trabajar con datos. Una de estas librerías es Scikit Learn la cual fue desarrollada en 2007 por David Cournapeau y que para 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort y Vincent Michel de INRIA, tomaron el proyecto e hicieron el primer lanzamiento público en ese año; dicha librería cuenta con herramientas simples y eficientes para el trabajo realizado con datos (Scikit-Learn, 2016), además esta provee algoritmos del tipo supervisado y no supervisado, incluyendo librerías como SciPy, NumPy, Matplotlib, IPython, Sympy y Pandas; enfocándose esta libreria en la facilidad de uso, la calidad de código, colaboración, documentación y rendimiento, de igual modo esta se preocupa por el uso eficiente de funciones, a través de usar librerías-C para mejorar su rendimiento (Brownlee, 2014).

Dentro del área de análisis de datos se deben considerar pasos previos a la aplicación de los métodos a usar, esto es importante no sólo para mejorar el rendimiento de los algoritmos a utilizar sino también para reducir las posibilidades de error de los mismos. Este proceso es conocido como la preparación de datos. Este proceso es crítico dado que para que los diferentes algoritmos puedan aprender de la forma correcta, cuando es necesario, se necesita que se les proporcionen los datos correctos para el problema que se desea resolver (Brownlee, 2013). Este involucra diferentes pasos.

- **Selección de datos**: Consiste en la selección de subconjuntos de datos de todos los disponibles con los que se puede trabajar. Se busca que se considere que datos realmente son necesarios para poder resolver el problema en el que se está trabajando (Brownlee, 2013). Dentro de esta se considera la selección de características o variables, dada la gran influencia sobre el desempeño de los métodos a desarrollar que estas tienen, llegando a afectar negativamente cuando se trabaja con características irrelevantes. Nótese que la selección de características es donde se eligen aquellas que contribuyen mayormente a la salida en la que se está interesado. Si se eligen adecuadamente las características del modelo se puede reducir el overfitting, pues al tener menos datos redudantes existe menor oportunidad de hacer de-

cisiones en base al ruido de los datos; también se mejora la exactitud de los resultados, pues al tener menor datos que producen ruido la precisión del modelo mejora; de igual modo se reduce el tiempo de entrenamiento, dado que se tienen menor cantidad de datos por ende los algoritmos aprenden más rápido (Brownlee, 2016).

- **Preprocesar los datos**: En este se considera como se usarán los datos seleccionados previamente, con la finalidad de poner los datos en una forma en la cual se puedan trabajar (Brownlee, 2013). Se consideran tres pasos:

  - Formatear: Este se da pues algunas veces los datos no se encuentran en el formato que se desea, pudiendo incluirse el proceso de pasar la forma original de guardado de los datos a una nueva, como el pasar de una base de datos relacional hacia un archivo de texto plano; además se considera cuando se desea dar un nuevo formato a los datos como tal, siendo por ejemplo la conversión de una cadena a un tipo de dato como una fecha o bien un número flotante (Brownlee, 2013).
  - Limpiar: Este consiste en el borrado o arreglo de los datos que hacen falta dentro del conjunto de datos. Esto se da cuando existen instancias de datos que estan incompletos y que no tenian los datos que se creían importantes para el problema, en tal caso se recomienda eliminar estos datos. De igual modo, cuando existen datos sensibles a algunos atributos, estos atributos necesitan ser disacociados o bien eliminarse (Brownlee, 2013).
  - Muestreo: Este es necesario cuando hay muchos más datos disponibles que los que se necesita para trabajar, en cuyo caso el elegir muchos datos pueda traer como resultado tiempos de ejecución prolongados y más requerimientos computacionales como en la memoria. Por ello es recomendable tomar pequeñas muestras de datos para ser exploradas previo a considerar todos los datos (Brownlee, 2013).

- **Transformar los datos**: Este paso se ve influenciado por el tipo de algoritmo con el que se esté trabajando (Brownlee, 2013). En este se consideran tres pasos:

  - Escalar: Los datos procesados previamente pueden tener atributes con una mezcla en las escalas para diferentes cantidades, llegando a provocar que varios métodos no devuelvan los resultados esperados, por ello es necesario tener la posibilidad de convertir los datos a una escala entre 0 y 1, y en algunas ocasiones buscar que los datos muestren una distribución específica con la finalidad que los métodos se comporten lo mejor posible (Brownlee, 2013).
  - Descomponer: Este es necesario cuando existen características o variables que representan un concepto complejo que pueden ser más útiles cuando esta divido en partes, como

es el caso en el uso de una fecha, la cual puede tener componentes como día, hora y otros, y puede que solamente día sea relevante para el modelo (Brownlee, 2013).

- Agregar: Este se da cuando hay variables o características que tienen más significado dentro del modelo al estar unidas, para ello se deben considerar diferentes métodos para agreación que se pueden llevar a cabo y cúal tiene más significado para el problema que se desea resolver (Brownlee, 2013).

Según el estudio de Eichinger *et al.*, las diferentes técnicas para la predicción de series de tiempo llegan a diferir no solo en las técnicas que se utilizan sino también en el período de tiempo para el cual se desea implementar la predicción ya sea para un tiempo por hora, por día, por mes o por año. De igual manera en dicho estudio se catalogan las técnicas de predicción de la siguiente manera:

- **Auto regresión**: Este es un grupo de técnicas que se basan en el uso de modelos matemáticos los cuales emplean los valores previos de la serie de tiempo con la finalidad de realizar predicciones. Esta tiene que trabajar con problemas que trae el efecto de la estacionalidad de los datos, para lo cual se han desarrollado diversas técnicas. (Eichinger & Pathmaperuma, 2013)

- **Ténicas de Suavizado Exponecial:** Las cuales son moving-average, es decir que pertenece al grupo de las series de tiempo que son construidas a través de tomar varios promedios de diversas secuencias de valores provenientes de otra serie de tiempo. (Hyndman, 2009). Siendo estas un subgrupo que usan pesos con un factor de decaimiento exponencial a través del tiempo.(Eichinger & Pathmaperuma, 2013)

- **Técnicas de Aprendizaje de Máquina**: Muchas de las técnicas de esta área han sido bien adaptadas para realizar predicciones sobre el tiempo, dentro de las cuales se pueden mencionar las Redes Neuronales o bien la técnica de Máquinas de vectores de soporte conocidas en inglés como Suport Vector Machines. (Eichinger & Pathmaperuma, 2013)

Una aproximación para realizar el análisis de datos es el aplicar algoritmos del área de **aprendizaje de máquina**, este se entiende como el estudio de algoritmos computacionales para aprender a hacer ciertas acciones como el aprender a completar tareas específicas o bien el realizar predicciones más acertadas. El tipo de aprendizaje que se lleve a cabo debe estar siempre basado en un grupo de observaciones o bien de datos. Entonces se podría decir que el aprendizaje de máquina es sobre aprender a tener un mejor desempeño en el futuro basado en lo que se ha experimentado en el pasado. (Schapire, 2013)

En el área del aprendizaje de máquina existen dos tipos de aprendizaje conocidos como aprendizaje supervisado y no supervisado. En el aprendizaje **no supervisado** se estudia una clase de

problemas en los cuales se busca determinar cómo los datos están organizados. Es decir, estudia cómo los sistemas pueden aprender a representar un ingreso en particular en términos de un patrón, de tal manera que refleje la estructura estadística sobre la totalidad del conjunto de patrones de ingreso. (Dayan, 2009)

Por otro lado el **aprendizaje supervisado** se basa en deducir una función de un grupo de datos de entrenamiento. Dicho conjunto de datos consiste en pares de vectores de ingreso, que son típicamente vectores, y sus salidas deseadas.La salida de la función puede ser un valor continuo, que se le conoce como regresión; o bien puede predecir una etiqueta de la clase del objeto de entrada, es conocida como clasificación. La tarea primordial del aprendizaje supervisado es el predecir el valor de la función para cualquier objeto de ingreso válido basado en un número de ejemplos de entrenamiento; pero para lograr esto, el agente que está aprendiendo tiene que generalizar de los datos actuales a una situación desconocida de una forma razonable.

1. Predicción. En el área de consumo de energía eléctrica se han realizado diversos estudios para analizar el consumo de hogares y empresas, de los cuales debe notarse que la mayoría de estos estudios se han enfocado en la predicción de consumo de energía eléctrica por hora, semana y mes. Para el caso de los estudios centrados en hogares se han enfocado en el consumo mensual, esto dado que por ser un hogar las empresas eléctricas solamente proporcionan datos por mes; mientras que para el estudio de empresas se han encontrado sensores que han proporcionado datos por hora, semana y mes. (Edwards, New, & Parker, 2012) A pesar de esto, existe un estudio realizado por Edwards. New y Parker de la universidad de Tennessee, en el cual han intentado aplicar diversos algoritmos de aprendizaje de máquina que han sido usados para realizar predicciones por hora y semana para empresas pero esta vez aplicados a hogares usando un conjunto de datos que contenían información sobre el consumo de energía de hogares por hora. Durante el estudio de Edwards y sus colegas, se verificó que para la predicción de consumo de energía en empresas un método basado en redes neuronales se desempeña de mejor manera. Por otro lado, para el área hogares se mostró que la mayoría de modelados para empresas funcionaban de manera decadente, pero el método de Least Squares Support Vector Machines lo hizo de mejor manera.

Entre las técnicas que se han usado en diversos estudios relacionados con el análisis de datos de consumo de energía, en el área de predicción, se encuentran las siguientes:

- **Regresión Lineal:** Esta es una de las técnicas más simples, y esta basada en el ajuste de una función lineal con la forma $y = \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \beta$. Donde y es el valor objetivo y x son los valores de entrada y los βs representan el peso de la función. Debe notarse que a pesar de que este es un modelo simplista, es bastante usado para lograr predicciones con una medida de rendimiento lineal en el área de predicción de consumo eléctrico.(Edwards *et al.*, 2012)

- **Red Neuronal con Feed Forward**: También conocida como FFNN por sus siglas en inglés. Edwars *et al*, menciona que en estudios previos como los realizados por MacKay *et al*, han mostrado que las Feed Forward Neural Network son capaces de predecir el consumo de energía eléctrica de forma eficiente, con lo cual se ha hecho claro que este tipo de método puede ser bastante útil para poder aproximar funciones no lineales, lo cual quiere decir que una FFNN puede aproximar una función que mapea de $\Re^m \rightarrow \Re$ sin necesidad de hacer supuestos sobre la relación que existe en las entradas y salidas. Pero este sí requiere que se defina la estructura del modelo, incluyendo el número de capas y unidades escondidas dentro de la red, así como de cualquier otro parámetro asociado. Con respecto a los pesos de este método son aprendidos a través de usar métodos basados en gradiente descendiente como el de Newton Raphson, a través de minimizar una función de error especificada por el usuario. Pueden usarse diversas funciones para el error, pero bajo el estudio de Edwards y sus colegas, para este tipo de problemas de consumo de energía es mejor usar el **Sum Squared Error**, el cual es la suma de las diferencias al cuadrado de cada observación y la media de su conjunto total de datos (Huguenard, 2015); pero con el uso de este vienen dos problemas como el over-fitting, el cual es cuando el algoritmo o modelo utilizado captura y trabaja con el ruido que contienen los datos, es decir, que el método se ajusta demasiado bien a los datos (Cai, 2014); pero la FFNN podría ajustar sus pesos de manera tal que pueda actuar de manera adecuada en la fase de entrenamiento, pero esto trae consigo el problema que no podría proceder adecuadamente para nuevas entradas de datos. Este tipo de problema puede ser mitigado a través de separar el conjunto de entrenamiento en dos partes, uno para entrenamiento y otro para validación. (Edwards *et al.*, 2012)

- **Máquina de vectores de soporte para regresión**: Llamado en inglés como Support Vector Regression o SVR por sus siglas en inglés. Este tiene como objetivo el disminuir el riesgo estructural, es decir, minimizar la probabilidad de que un modelo construido de un conjunto de datos de entrenamiento vaya a cometer errores en los nuevos datos por buscar la mejor solución para la generalización de los datos de entrenamiento. La mejor solución puede ser encontrada a través de minimizar la ecuación convexa de criterio (Edwards *et al.*, 2012):

$$\frac{1}{2}||w|| + C\sum_{i=1}^{l} \xi_i + \xi_i^*$$

Esta ecuación tiene como restricciones:

$$y_i - w^T\varphi(\vec{x}) - b \leq \in + \xi_i$$

$$w^T\varphi(\vec{x}) + b - y_i \leq \in + \xi_i^*$$

Donde $\in$, define el rango deseado de error para los puntos, xi_i y ξ_i^* son variables de holgura, llamadas así a aquellas variables que son agregadas a la restricción de una inecuación con la finalidad de convertirla en una ecuación(Boyd & Vandenberghe, 2004); para este caso son usadas para garantizar que la solución existe para todo $\in$, C es la penalización para balancear entre el ajuste de los datos y la suavidad. w son los pesos de la regresión. φ, representa una función kernel para lograr el mapeo del espacio de ingreso a una dimensión más alta de espacio de las características.

Una de las ventajas que este método ofrece es que hay una única solución que minimiza una función convexa, pero dicha solución es dependiente del ingreso C, $\in$ y de los parámetros necesarios para la función kernel seleccionada. Para lo que refiere a la selección de parámetros existen técnicas como la de **rejilla de búsqueda con validación cruzada** el cual consiste en producir una validación que estima el rendimiento estadístico para cada punto dentro de una cuadrícula (Krstajic, Buturovic, Leahy, & Thomas, 2014); **validación cruzada de dejar-una-fuera** (leave-one-out cross-validation en inglés) la cual consiste que para un número igual a la cantidad de datos la función aproximadora es entrenada sobre todos los datos a excepción de uno y la predicción es hecha para ese punto (Schneider, n.d.); de igual manera se podrían usar otros métodos de este tipo. El problema que existe con este método es la escalabilidad, pues la función convexa de criterio mencionada enteriormente, es optimizada de algoritmos de optmización de programación cuadrática, los cuales suelen requerir altos niveles de memoria y velocidad cuando los datos usados son demasiados. (Edwards *et al.*, 2012)

Cabe decirse que este método puede variar su complejidad en términos de tiempo, dado que cuenta con diversas variaciones para la funcion kernel o núcleo. Entre los más populares se encuentran:

- **Lineal:** Este es el más simple, pues es dado por el producto interno de las componentes del núclo más una constante C. Este suele representarse como $< x, y >$ (Souza, 2010), pero tiene esta definido por:

$$k(x, y) = x^T y + c$$

- **Polinomial:** Este es un tipo de núcleo no estacionario, siendo bien usados para aquellos problmeas con los datos de entrenamiento están normalizados (Souza, 2010). Este suele representarse como $(\gamma < x, y > +r)^d$, estando definido por:

$$k(x, y) = (\alpha x^T y + c)^d$$

- **RBF:** Siendo las siglas para *radial basis function*, aunque también se le suele llamar

como el kernel Gaussiano (Souza, 2010). Se le suele representar como $exp(-\gamma||x-y||^2)$ pero su representación está dada por

$$k(x,y) = exp(-\frac{||x-y||^2}{2\sigma^2})$$

De la cual su valor signma juega un papel importnate para el desempeño de este, pues si se sobre estima, el exponencial se comportara casi de forma lineal y la proyección dimensional perderá su característica no lineal; por otro lado si se subestima, la función carecerá de regularización y los límites de decisión se verá muy afectado por el ruido de los datos en la fase de entrenamiento (Souza, 2010).

- **Sigmoid:** Este es conocido como el kernel de Tangete hiperbólica. Viene del campo de las redes neuronales, donde se usa una función sigmoid como función de activación (Souza, 2010). Usualmente se le representa como $tanh(\gamma < x, y > +r)$, pero su función está definida por:

$$k(x,y) = tanh(\alpha x^T y + c)$$

Algo interesante de notar que al usar esta función como kernel, este método se asemeja a una red de percetron de dos capas. Se ha encontrado que es bastante popular gracias a su buen desemepeño en la práctica (Souza, 2010).

- **Mezcla jerárquica de expertos**: Este es un tipo de red neuronal la cual aprende a particionar un espacio de ingresos a través de un conjunto de expertos, lo cual para este caso, el ingreso sería los valores directamente del sensor. Dichos expertos se especializarán en una región en particular, o bien se asistirán entre ellos en el aprendizaje sobre la región. Este tipo de modelados encuentra su aplicación en conjuntos de datos donde estos se pueden dividir en sub poblaciones, lo cual va a acorde para poder analizar el consumo de energía de hogares a través de diferentes estaciones, por lo cual dichas variaciones dependientes de la estación puede que sea mejor analizar por separado. Entonces, este modelo intenta descubrir cada diferente tipo de sub modelado de forma automática y designar a un experto al área (Edwards *et al.*, 2012).

- **Least Squares Support Vector Machines**: Conocido como LV-SVM por sus siglas en inglés, es muy similar al conocido como Support Vector Regression, este último fue desarrollado para minimizar el riesgo estructural, lo cual quiere decir, que busca minimizar la probabilidad de que el modelo construido a partir de la fase de entrenamiento realizará ciertos errores en las nuevas verificaciones a través de buscar una solución que mejor generalice la fase de prueba. El método LS-SVM difiere del SVR dado que la función de criterio es diferente, puesto que esta está basada en los mínimos cuadrados, además de que las restricciones

del problema son cambiadas de una inecuación a una ecuación, (Edwards *et al.*, 2012) lo cual lleva a tener una función de optimización formulada como sigue:

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i^2$$

Con la restricción:

$$w^T\varphi(\vec{x}) + b + \xi_i = y_i$$

Este algoritmo tiene como ventaja el hecho de modificar la función de criterio no requiere programación cuadrática para solventar el problema de optimización; lo cual lleva a este algoritmo a poder resolver de una manera mucho más rápida. Por otro lado, este método emplea todos los datos para definir su solución, lo cual lleva a este a perder la propiedad de escasez, lo que puede afectar a la solución para poder ser generalizada. Pero existen estudios como los de Suykens, Brabanter, Lukas y Vandewalle de la Universidad de Leuven; además del realizado por Hoegaerts, Suykens, Vandewalle y De Moor también de la Universidad de Leuven, que sobrellevan dicho problema haciendo uso de prunning o dando pesos.(Edwards *et al.*, 2012)

- **Fuzzy C-Means con Redes Neuronales con Feed Forward :** Este tiene como base la separación del proceso de aprendizaje en dos fases. La primera es una fase de aprendizaje no supervisado la cual emplea clustering para aproximar $P(Z|X)$. Mientras la segunda fase emplea cada cluster para entrenar a los expertos. Es decir, que este es una modificación al método de Mezcla Jerárquica de Expertos. (Edwards *et al.*, 2012) Para realizar la parte de clustering se podría usar cualquier tipo de algoritmo realizado como K-Means, que es explicado más adelante, **Self-Organizing Maps** que consiste en reducir el tamaño del conjunto de datos, que logra al usar redes neuronales auto organizadas (Germano, 1999); o bien Clustering Jerárquico. Pero debe notarse que un algoritmo que no permita que las observaciones puedan pertenecer a varios clusters producirá una aproximación demasiado rígida, lo cual provocará que los Expertos ignoren grandes conjuntos de datos de observaciones y así se produzcan modelos demasiado pobres. En el trabajo de Edwards *et al*, se emplea Fuzzy-C Means para controlar y evitar las aproximaciones rígidas. Este está basado en minimizar la función de criterio:

$$\sum_{i=1}^{N}\sum_{j=1}^{C}u_{ij}^m||\vec{x}_i - \vec{c}|j||^2$$

Donde u_{ij} representa la probabilidad de que $\vec{x}_i$ sea un miembro de un cluster $\vec{c}_j$, y m es un parámetro definido por el usuario que controla cuanto una observación pertenece a múltiples clusters. Para minimizar la función anterior a través de un proceso iterativo se emplean las

ecuaciones:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \vec{x}_i}{\sum_{i=1}^{N} u_{ij}^m}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \frac{||\vec{x}_i - \vec{c}_j||^{\frac{2}{m-1}}}{||\vec{x}_i - \vec{c}_k||}}$$

A pesar que este método pareciera ser mejor que el de Mezcla Jerárquica de Expertos, tiene un gran defecto al basarse en la idea de asumir que la relación espacial entre las observaciones puede aproximar $P(Z|X)$, mientras que el otro método aproxima $P(Z|X)$ al maximizar $P(Y|X, \theta)$ (Edwards *et al.*, 2012).

- **Dependencias temporales:** Este método es basado en la idea de que cada respuesta del objetivo yt es dependiente de las observaciones pasadas xt-1 así como de las observaciones actuales xt. Dichas dependencias siguen un **orden de Markov** o bien están dispersos. Si el orden de Markov es seguido, la respuesta es dependiente de conjunto previo de observaciones. Por otro lado si están dispersos esto indica que yt puede ser dependiente de una combinación de observaciones pasadas en lugar de un conjunto completo, el explorar todas las combinaciones de las dependencias esparcidas crece de una manera exponencial, lo cual puede llegar a ser contradictorio. En el trabajo de Edwards *et al*, se indica que es recomendable usar las observaciones x_{t-1}, x_{t-2}, etcétera para predecir y, pues de no seguirse esta condición se usaría información futura para predecir yt, lo cual indica que es mejor seguir un orden de Markov. Cabe notarse que el uso de un orden esparcido puede ser útil para cuando el conjunto de datos es muy pequeño dado que el crecimiento exponencial de este tipo de algoritmos puede llegar a ser demandante en el área de recursos cuando el conjunto de datos utilizado es demasiado grande.(Edwards *et al.*, 2012)

- **Árboles de decisión:** Los árboles de decisión son dados por algoritmos que son capaces de identificar diversas maneras de dividir un conjunto de datos con el fin de crear segmentos como ramas (deVille, 2006). De igual forma, se puede decir que un árbol de decisión es una estructura de árbol en la cual cada una de sus ramas representa una decisión entre diversas alternativas, y cada una de sus hojas tiene como representanción una decisión, estos tienen como ventaja el ser bastante rápidos en la parte de aprendizaje (Wilson, 2015). Este tiene diferentes acercamientos para la construcción del árbol, cada uno de estos algoritmos tienen diferentes áreas de aplicación, siendo los más populares los siguientes:

  - **ID3**: Llamado así por sus siglas en ingés *Iterative Dichotomiser 3* fue desarrollado en 1986 por Ross Quinlan. Este busca construir el árbol de decisión con un acercamiento .ªrriba-abajo", con una búsqueda codiciosa a través del conjunto dado y probar cada atributo en cada nodo (Peng, Chen & Zhou, 2009)

- **C4.5** Este construye un árbol de decisión con una estrategia de "divide y vencerás". Para cada nodo en el árbol se le adjunta un conjunto de casos, además se les asigna pesos para tomar en cuenta valores desconocidos (Ruggieri, 2002). Este algoritmo agrega el concepto de proporción de información ganada y atributos continuos; este elige un atributo basado en el radio de información obtenida, además puede trabajar con atributos continuos en una forma discreta y procesar datos deficientes (Dai, Zhang & Wu, 2016).
- **CART:** Llamado así por sus siglas en inglés *Classification and Regression Trees*. Fue propuesto por Breiman *et al* en 1984. Este tipo de árbol es uno del tipo árbol de decisión binario, que se construye a través de dividir cada nodo en dos nodos hijos de forma repetitiva, comenzando por una raíz que tiene toda la muestra de entrenamiento. Nótese que se busca dividir lo más posible cada nodo para que al final cada nodo hijo sea "lo más puro"posible, así mismo cada división depende unicamente del valor de una sola variable predictora (Breiman *et al*, 1984).

Además de las herramientas que ofrece el área de aprendizaje de máquina, también existen algoritmos que se basan en procesos puramente estadísticos para estimar la relación entre variables; usualmente se busca el encontrar una relación causal entre las variables. Este tipo de análisis incluye diversas técnicas para modelar y analizar numerosas variables entre las cuales se encuentran los modelos auto regresivo (**AR** por sus siglas en inglés), modelos en movimiento promedio (**MA** por sus siglas en inglés), modelos en movimiento promedio autorregresivo (**ARMA** por sus siglas en inglés) y modelos auto regresivo integrados de promedio móvil (**ARIMA** por sus siglas en inglés).(Sykes, 2007)

Un modelo auto regresivo se da cuando un valor de la serie temporal, entiéndase esta como una secuencia de medidas de una misma variable en un tiempo determinado, es retrocedido a valores previos dentro de la misma serie temporal, como ejemplo se puede mencionar retrocedida en la ecuación . Cabe decirse que el orden de una auto regresión es el número de valores inmediatamente anteriores de la serie que se emplean para calcular el valor en el presente.(Romer, 2016a)

Para el caso de un modelo de movimiento promedio, este es usado de manera común para modelar series temporales invariantes. Este tipo de modelado en lugar de usar valores anteriores para la predicción, usa los errores de las predicciones pasadas en un modelo de regresión. (Romer, 2016b) Este modelo se puede representar con la ecuación $y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \ldots + \theta_q e_{t-q}$ donde e_t es lo que se conoce como ruido blanco. Además que q representa el grado del modelado. De igual manera cabe decirse que cada valor de y_t puede ser considerado como un promedio en movimiento con peso de algunos errores de unas predicciones pasadas.(Hyndman & Athanasopoulos, 2016b)

En lo que respecta a modelos en movimiento promedio autorregresivo son modelos matemáticos

de persistencia o autocorrelación en una serie temporal. Estos modelos son altamente usados en diferentes campos como la hidrología, econometría entre otros. Este tipo son usados para predecir el comportamiento de una serie de tiempo dados los valores pasados (Meko, 2015). De igual manera la necesidad de uso de este tipo de regresión viene dada porque algunas veces las series temporales necesitan un orden alto de modelado autorregresivo o bien de movimiento promedio, esto para poder modelar de manera correcta el proceso; es para estos casos que se emplea este tipo de modelado. Nótese que este modelo expresa la media condicional de un valor y_t como función de $y_{t-1}, ..., y_{t-p}$ y de los errores $e_{t-1}, ..., e_{t-q}$. El número de observaciones pasadas de las cuales llega a depender y_t depende de p, el cual es el grado de auto regresión; por otro lado, el número de errores pasados del cual y_t depende se da por q, el cual es el grado de movimiento promedio, lo cual lleva a representar el modelo como $ARMA(p, q)$. Además la metodología ARMA, se basa en usar tanto análisis de auto regresión y métodos basados en promedio para obtener el comportamiento de una serie de datos. Este asume que los datos son estacionariamente fluctuantes de una forma un tanto uniforme alrededor de una media de tiempo invariante. Este se considera de poco uso para el análisis de impacto o bien para aquellos donde se incorpore aleatoriedad.(MathWorks, n.d.)

Por otro lado, el modelado autorregresivo integrado de promedio móvil predice un valor en base a una combinación lineal de sus propios valores y errores pasados y el valor actual además de los valores pasados de otras series temporales. Este procedimiento provee un conjunto de herramientas comprensivas para una serie temporal invariante con un modelo de identificación, parámetros de estimación y predicción.(Institute Inc., 1999) El método conocido como Autoregressive Integrated Moving Average (ARIMA), es una generalización del modelo conocido como Autoregressive Moving Average (ARMA). Ambos modelos son alimentados con un conjunto de datos que son usados para comprender o bien para predecir puntos futuros en la serie, de igual manera encuentran su aplicación en aquellos conjuntos de datos donde no se muestra un comportamiento estacionario. Para el caso específico de ARIMA este es una técnica para realizar predicciones que se basa en la inercia de los datos. (Bossche, Wets, & Brijs, 2004) De igual manera su principal aplicación es en aquellas predicciones cortas pues este solamente requiere 40 puntos de datos históricos por lo menos; además este funciona mejor cuando los datos muestran un patrón estable o consistente sobre el tiempo con una cantidad pequeña de datos atípicos. ARIMA es superior usualmente para cuando los datos son razonablemente extensos y la correlación entre las observaciones pasadas es estable (Hyndman & Athanasopoulos, 2016a).

Para el área de predicción de energía se han usado diversos métodos estadísticos como los ya mencionados. En el estudio realizado por Chujai, Kerdprasop y Kerdprasop en Hong Kong, se ha demostrado que para el análisis de consumo de energía y predicción de la misma, es preferible usar el método de ARIMA para hacer predicciones a nivel mensual y por cada 4 meses, además del ARMA para períodos diarios y semanales. De la misma manera se han encontrado los mismos

resultados de parte de Saab y sus colegas en El Líbano. Además en el estudio de dado por Zhu Go y Feng donde se compararon los métodos de BVAR y ARIMA para la predicción de datos, resultó que el modelado en base a ARIMA resultó tener menor error. (Chujai, Kerdprasop, & Kerdprasop, 2013)

2. Agrupación. En lo que respecta al área de clasificación de datos, la mayoría de estudios realizados orientados a esta área como el de Riveiro et. al, así como el Eichinger *et al*, o bien el de Phrahastono *et al*, y el de Chicco *et al*, han usado algoritmos del área de aprendizaje de máquina teniendo diversas variaciones entre los mismos pero siempre con el mismo objetivo, el clasificar el consumo de energía.

En lo referente a clustering, esta es una técnica no supervisada para agrupar datos basados en su similitud mutua, por ello los algoritmos que se emplean para este pueda detectar datos similares y con ello separar pares de datos no similiares en diferentes grupos. Dentro de esta sección se han propuesto diversos métodos como **clusters basados en densidad**, este lleva a cabo la agrupación de los puntos que se encuentran cercanos, marcándose como forasteros a aquellos datos que se encuentran en lugares con poca densidad de puntos; y también **clústers espectrales** el cual consiste en realizar los cluster con los datos que están conectados pero no necesariamente están compactados o dentro de un cluster con límites convexos, para este se hace uso del espectro o eigenvalores de la matriz de similitud de los datos para realizar la reducción de dimensionalidad antes de realizar los clusters en menos dimensiones. Todas estas formas difieren en la misma definición que tiene de clusters, pero como área común se tiene que todas las salidas un conjunto de clusters, es decir una partición de los datos bien definidos (Eichinger & Pathmaperuma, 2013).

Por ello Prahastono *et al.* proveen una revisión de algunos métodos de clustering que han sido usualmente usados en el área de electricidad. Cada uno de los métodos que menciona Prahastono *et al* tienen como parte inicial el derivar una matriz de caracteristicas para cada perfil de conjunto de datos, luego cada uno de ellos sigue un procedimiento descrito a continuación.

- **Jerárquico**: Este agrupa los datos y de forma simultánea lo hace sobre varias escalas, esto a través de crear un árbol de clusters, dicho árbol es de jerarquía de multi-nivel, donde cada uno de los clusters son unidos entre los diferentes niveles. (Prahastono, King, & Özveren, 2007) Para llevar a cabo este método es necesario encontrar similitudes o bien disimilitudes entre cada uno de los pares de perfiles cargados dentro de los datos basados en la matriz de similitudes computada anteriormente. Una de las ventajas que ofrece este método es que los datos originales permanecen intactos en la raíz del árbol de clusters.(Prahastono, King, & Özveren, 2007) El determinar la similitud o bien distancia entre los datos puede ser llevada a cabo de diversas maneras que pueden incluir **distancia Euclidiana**, comúnmente la más utilizada. Esta plantea que la distancia entre dos puntos (x, y) y (a, b) esta dada por $\sqrt[2]{(x-a)^2+(y-b)^2}$ (Bogomonly, 2016), **distancia Mahalanobis** esta es también es cono-

cida como distancia cuadrática, al igual que la anterior sirve para medir la distancia entre dos puntos, esta se puede definir con la formula $d_{ij} = [(\vec{x}_i - \vec{x}_j)^T S^{-1}(\vec{x}_i - \vec{x}_j)]^{0,5}$, nótese que los grupos de datos deben tener el mismo número de variables pero no es necesario que tengan el mismo número de observaciones, es decir el mismo número de columnas pero no asi el mismo número de filas (Teknomo, 2015); también existe la conocida como **métrica de bloque de ciudad** en la cual la distancia entre dos puentos es la suma del valor absoluto de la diferencia de sus coordenadas, es decir $|x_1 - x_2| + |y_1 - y_2|$ (Pieterse & Black, 2006). Así mismo está la **métrica Minkowski** haciendo que la distancia entre dos puntos en un espacio vectorial normal este dada por $d(x, y) = (\sum_{i=0}^{n-1} |x_i - y_i|^p)^{\frac{1}{p}}$, **distancia Hamming**, este es un número para denotar la distancia entre dos cadenas binarias a través de la fórmula $\sum |A_i - B_i|$ (Pieterse& Black, 2001), este tiene como peculiaridad que permite a las computadoras detectar y corregir errores por su cuenta.(McKenzie, n.d.) En lo que refiere a la agrupación esta puede ser procesada a través de enlazar los pares de perfiles de carga que están en una proximidad usando un criterio de enlace como **distancia corta** el cual consiste en elegir la menor distancia a disposición, **distancia promedio** en el que se considera como la distancia a alegir la que se genera del promedio de las distancias a posibles, **distancia de centroide** donde se toma la distancia desde y hasta los centroides de los puntos, o bien **distancia Ward** el cual es un algoritmo de jerarquía para clusters que minimiza la inercia en cada paso, se entiende inercia como la suma de la resta de los cuadrados, la señal residual y la inicial. (Murtagh & Legendre, 2011). Dado que los datos son pareados en un nuevo cluster binario, los clusters más recientemente formados son agrupados en un cluster más grande hasta que se forma un árbol jerárquico. Este método es recomendable cuando el número de grupos no está predeterminado, de hecho la posición de corte determinará el número de clusters. (Prahastono, King, & Özveren, 2007)

Se debe notar que para este método existen dos acercamientos, uno de ellos es llamado **método de división**, en la cual se empieza con un sólo grupo, para luego ser divido en dos nuevos grupos que comparten similitudes, así se procede de forma recursiva hasta que solamente queda un clúster para cada observación o bien hasta que se alcanza un número determinado de grupos, a este también se le conoce como una estrategia .ªrriba-abajo"(Tibshiriani, 2013). El otro acercamiento se le conoce como **método aglomerativo**, el cual consite en iniciar con todas las observaciones en su proprio clúster y luego ir uniendo dos de estos grupos, hasta que exista un sólo grupo o bien hasta que se alcance un límite dado, a esta estrategia se le conoce como .ªbajo-arriba"(Tibshiriani, 2013).

- **K-means:** Este agrupa los perfiles de carga al determinar cierto número de clusters y un punto central para cada cluster. Tras haber sido determinado el punto central para cada grupo, cada conjunto de datos debería ser asignado al punto central más cercano y luego

realizar una recalculación de un nuevo punto central, esto se hará de forma iterativa hasta que las posiciones de cada punto central sean estables. La asiganción de los datos a un punto central es evaluado mediante el uso de la distancia Euclidiana, Bloque de Cidudad, y Hamming. También se considera **Cosine,** el cual mide la distancia entre dos vectores y un espacio interno del producto, tomando la medida del coseno del ángulo entre ellos (Perone, 2013). Además está la **Correlación** , este es análogo a la covarianza del producto del momento; pero a diferencia de la definición tradicional esta es solamente cero si los vectores aleatorios son independientes. A diferencia del método anterior, este no crea un árbol o tabla para describir los grupos de los datos pero en su lugar crea un solo nivel de clusters. Este método usa las observaciones actuales de los datos por lo cual es más aplicable para realizar clustering de grandes conjuntos de datos. (Prahastono, King, & Özveren, 2007)

Uno de los algoritmos más famosos para realizar este método es conocido como el **algoritmo de Lloyd**, el cual fue propuesto en 1957, debe notarse que este trabaja de forma iterativa mejorando el centroide de los clústeres, se tiene que hacer énfasis en el hecho que la forma en que empiezan los centroides no es parte del algoritmo y algunas veces es necesario que se proveea como parámetro; entonces este algoritmo trabaja basado en dos pasos, el primero todos los puntos son asigandos a uno de los centroides que tengan más carca, y en el segundo paso todos los centroides son actualizados a través de calcular la media de todos los puntos en el grupo (Eliasson & Rosén, 2013). Se debe notar que dichos pasos son realizados hasta que algún criterio sea dado, además que una de las métricas más comunes para medir la distancia es a través de la distancia Euclidiana (Eliasson & Rosén, 2013).

- **Fuzzy K-means**: Este método es muy similar al anteriormente mencionado, teniendo como diferencia que cada conjunto de datos tiene un grado de pertenencia a cada cluster inicial, es decir que todo conjunto de datos pertenece a todos los clusters con un grado diferente. Cabe decirse que cada grado de pertenecia para un dato debe sumar uno. Este inicia con determinar el número de clusters y así adivinar el punto central de cada cluster, el cual pretende marcar la localización para cada cluster y luego asignar un grado de pertenencia a cada conjunto de datos; luego de ello se actualizan los puntos centrales del cluster y así mismo los grados de pertenencia. Este métdo no genera límites entre los conjuntos de datos para la primera iteración dado que el proceso de clustering envuelve todos los datos, los límites evolucionarán de forma automática cuando el proceso de clustering esté completo. En comparación con el k-means, este método es más largo dado que cada iteración no solamente es la actualización del punto central sino también el grado de pertenecía de cada conjunto de datos (Prahastono, King, & Özveren, 2007).

- **"Sigan al líder"**: Este método fue descrito por Yu *et al*, así como por Chicco *et al*. En dicho método se usa un proceso iterativo para computar los centros de los clusters y no es necesario el predeterminar el número de clusters, pues este es automáticamente derivado de la determinación de la distancia límite. El proceso de este método se detiene cuando el centro del cluster es estable. Al realizar la primera iteración del algoritmo se determina el número de clusters y la pertenencia de los perfiles de carga, lo cual crea límites entre los conjuntos de datos; el siguiente paso es ajustar los patrones de datos al cluster más cercano y actualizar el centro del cluster (Prahastono, King, & Özveren, 2007).
- **Fuzzy Relation**: Este método es basado en un proceso complejo de forma iterativa que se puede describir de manera simple como primero determinar la similitud de los perfiles de carga usando el método de amplitud Cosine, luego de ello agrupar los perfiles de carga al usar el método de composición max-min, que consiste en la combinación de dos relaciones del tipo fuzzy para determinar la cuán similares son un grupo de observaciones (Prahastono, King, & Özveren, 2007), y tras esto se determina el número de clusters al usar lambda-cuts para el método de relación fuzzy y así finalmente obtener el número de clusters. Este método usa la relación fuzzy para evaluar las similitudes y agrupar los conjuntos de datos, por ello los límites entre los grupos de datos es creado en la iteración final tras haber evaluado el límite. Esta técnica es recomendada para manejar grandes grupos de datos difusos con iteraciones complejas. El número de clases es decidido por el Lambda-Cuts. (Prahastono, King, & Özveren, 2007)

Por otro lado Nizar *et al* proponen como método k-Means, COBWEB y EM. Mencionando que el método EM es un método probabilístico que calcula los conjuntos más parecidos de clusters. El método de **EM** es uno basado en probabilidades con el objetivo de encontrar el conjunto al que más se pueda llegar a parecer un dato, este método calcula la probabilidad del cluster y los parámetros de distribución (Nizar, Member, Dong, Zhao, & Member, 2006). Por otro lado el método **COBWEB** es uno de clustering incremental donde se forma un árbol en cualquier fase, donde cada hoja y raíz representa el conjunto de datos completo(Nizar, Member, Dong, Zhao, & Member, 2006). El árbol consistirá en una raíz al inicio y luego diferentes instancias se irán agregando hasta que se haya actualizado de forma apropiada en cada fase y se usa una evaluación heurística para medir a que categoría puede pertenecer. En dicho estudio terminaron concluyendo que el método de k-Means era el mejor en términos de rapidez. (Riveiro, Johansson, & Karlsson, 2011)

De igual manera Chicco *et al.*, comparan diferentes métodos, entre los cuales están "sigan al líder de forma modificada, clustering jerárquico, k-means y fuzzy K-means. El **método modificado de "sigan al líder"** consiste en seguir el mismo principio ya mencionado anteriormente pero en

este estudio se ha modificado a través de tomar en cuenta la dispersión que tienen los datos, por lo cual se ha modificado la métrica Euclidiana que se usa en el algoritmo original al introducir para cada índice un peso con la forma $\frac{\sigma_h^2}{\bar{\sigma}^2}$, donde σ_h^2 es la varianza del $h - ava$ característica computado de todos los patrones en la población inicial, y $\bar{\sigma}^2$ es el promedio de los valores de la varianza. Teniendo como resultado en dicho estudio, que la modificación de sigan al líder y el clustering jerárquico parecían ser los más efectivos con respecto a los indicadores de validez(Chicco, Napoli, & Piglione, 2006)

Un aspecto que es importante notar de este tipo de métodos es que algunas veces es necesario predeterminar la cantidad de clústeres con la que se trabajará dado que métodos como el de K-Means necesita saber cuandos clústeres son deseados. Es por ello que existen diferentes métods que ayudan a saber cuántos clústeres deberían ser usados en los conjuntos de datos, entre estos vale la pena mencionara a:

- **Regla del pulgar:** Este es el método que puede ser aplicado a cualquier conjunto de datos, dado que se basa solamente en dividir por dos la cantidad de datos y a ello aplicarle la raíz cuadrada (Kodinariya & Makwana, 2013).
- **Método del codo:** Este es un método visual, teniendo como idea básica el empezar con 2 clústeres y luego ir incrementado 1 clúster a cada paso, calculando los nuevos grupos y el costo que viene con el entrenamiento, puesto que en algunos valores de la cantidad de clústeres el costo decae dramáticamente, y luego de eso alcanza un plano constante, dicho valor es el que se busca con este método (Kodinariya & Makwana, 2013).
- **Acercamiento de criterio de información:** Este es usado para seleccionar uno modelo dentro de varios que tienen diferentes parámetros, este busca balancear el incremento en la probabilidad dado por el agregar parámetros a través de introducir un término de penalidad a cada uno. El uso de técnicas de selección se emplea como base para saber el número de grupos basado en modelos mezclados. De igual manera, la selección es dada en dos fases, donde la primera de ellas se obtiene el conjunto candidato de modelos a través de un principio de aprendizaje para un rango de modelos, mientras que en la segunda fase se seleccionada el modelo apropiado basado en un criterio de selección (Kodinariya & Makwana, 2013).
- **Acercamiento de criterio de teórico:** Este es un acercamiento basado en una motivación fuertemente teórica, usando ideas del campo de la teoría de tipos de distorción, teniendo como ventaja que es fácil de entender y computar, además de ser bastante efectiva en variedad de problemas. Se basa en la "distorción"que es la medida de disperción dentro del grupo (Kodinariya & Makwana, 2013).
- **Método de la silueta:** Fue introducido por Kaufman y Rousseeuw. Este es considerado un coeficiente bien balanceado, que ha mostrado además un buen desempeño en diversos

experimentos. El concepto de este envuelve la diferencia entre los miembros de un grupo y la separación con los demás grupos (Kodinariya & Makwana, 2013). Este se define con la ecuación:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

Donde $a(i)$ es la distancia promedio entre i y los demás miembros del grupo al cual pertenece i, y $b(i)$ es el mímino del promedio de las distancias entre i y los demás datos en otros grupos (Kodinariya & Makwana, 2013).

- **Validación cruzada:** Este se basa en la estabilidad de un clúster, para lo cual divide en dos o más partes el conjunto de datos, una de las partes es usada para realizar la agrupación y la otra es usada para validar. La idea en la que se basa la .^estabilidad"de un grupo es que un algoritmo bueno tiende a repetir grupos similares de datos que vienen del mismo lugar, es decir que un algoritmo es estable con respecto de la aleatoridad de su ingreso (Kodinariya & Makwana, 2013).

- **Método de Pham *et al*:** Este método fue propuesto por Pham y su grupo en 2004, en el cual se introduce una función $f(K)$ con el fin de evaluar la calidad de un grupo resultante y para ayudar a decidir un valor óptimo para K para cada conjunto, esto a través de decir que al variar en diferentes números de K es posible evaluar el resultado de las agrupaciones producidas. Este tiene como objetivo el identificar regiones en las cuales los punto están concentrados, teniendo en cuenta que es importante analizar la distribución interna de cada grupo así como su relación a otros clústeres en el conjunto de datos. Dentro de este se define la distorción de un grupo como una medida de la distancia entre puntos y su centroide (Pham *et al*, 2004), dada por

$$I_j = \sum_{x_i \in C_j} ||x_i - \mu_j||^2$$

(Pham *et al*, 2004)

Así mismo se dice que el impacto global de todas las distorciones de los grupos esta dado por:

$$S_k = \sum_{j=1}^{K} I_j$$

(Pham *et al*, 2004)

Dentro de este método también se define una función de búsqueda $f(K)$, la cual debe verificar ciertos casos para estar informada sobre el problema de la selección de la cantidad de grupos

(Pham *et al*, 2004), con lo cual se llega a la definición siguiente:

$$f(K) = \begin{cases} 1 & \text{if } K = 1 \\ \frac{S_K}{\alpha_K S_{K-1}} & \text{if } S_{K-1} \neq 0, \forall K > 1 \\ 1 & \text{if } S_{K-1} = 0, \forall K > 1 \end{cases}$$

$$\alpha_K = \begin{cases} 1 - \frac{3}{4N_d} & \text{if } K = 2, N_d > 1 \\ \alpha_{K-1} + \frac{1-\alpha_{K-1}}{6} & \text{if } K > 2, N_d > 1 \end{cases}$$

(Pham *et al*, 2004)

Donde N_d es el número de atributos de los datos y α_K es el factor de peso. $f(K)$ es la relación de la distorción real con la estimada, esta disminuye cuando existen sectores de alta concentración en la distribución de los datos. Por esto el valor de K está determinado en el valor más pequeño de $f(K)$, pues son considerados grupos bien definidos (Pham et al, 2004).

F. Interfaz de usuario

Una **interfaz de usuario** se define como el espacio donde se da la interacción entre los humanos y las computadoras. El objetivo de esta interacción es permitir el funcionamiento y control efectivo de la máquina desde el usuario, mientras el equipo es capaz de retroalimentar y ayudar simultáneamente al usuario en la toma de decisiones (Christensson, 2012). Estas interfaces pueden ser de diferentes tipos como interfaces gráficas, pantallas táctiles o interfaces físicas. Típicamente, el objetivo del diseño de una interfaz de usuario es producir una interfaz que haga fácil (auto explicativa), eficiente y agradable (amigable al usuario) el operar un sistema (Raymond & Landley, 2004).

1. Diseño de interfaz Para maximizar la posibilidad de éxito en este objetivo de interacción, la interfaz de usuario debe de cumplir con las características de usabilidad y utilidad.

La **usabilidad** es un atributo de calidad de la interfaz de usuario e indica qué tan fácil es de usar (Nielsen, 2012). Para cumplir con esta característica la usabilidad define 5 componentes de calidad:

- **Facilidad de aprendizaje:** ¿Qué tan fácil es para los usuarios realizar las tareas básicas la primera vez que se encuentran con el diseño?
- **Eficiencia:** Una vez que los usuarios han aprendido el diseño de la aplicación, ¿qué tan rápido pueden realizar tareas.
- **Memorabilidad:** Después de un tiempo sin interactuar con el diseño, ¿qué tan fácil es para el usuario re acomodarse?

- **Errores:** ¿Cuántos errores comete el usuario, qué tan severos son estos y qué tan fácil pueden recuperarse?
- **Satisfacción:** ¿Qué tan agradable es el uso del diseño?

La **utilidad** es otro atributo de gran importancia en la interacción con el usuario. Esta se define como la funcionalidad del diseño, es decir, si el diseño permite al usuario hacer lo que necesita (Nielsen, 2012). Algunas herramientas genéricas propias de un sistema deberían estar presentes en la navegación principal (Farrell, 2015), como:

- Contacto
- Información del sistema
- Herramientas de idioma
- Inicio de sesión y registro
- Imprimir
- Guardar
- Buscar
- Herramientas de visualización de texto

Existen otras herramientas que dependen exclusivamente del tipo de sistema que se esté presentando. Al enlistar todas las funcionalidades del sistema, cada una adquiere una clasificación de importancia primaria o secundaria. Esta clasificación representa las herramientas que son más útiles y utilizadas por el usuario a través de la interfaz. Funcionalidades que no van a estar disponibles para el usuario directamente a través de la interfaz, no deben ser listadas a pesar de jugar un papel fundamental en el sistema. Al emplear esta clasificación, se cuenta con una serie de requerimientos que deben incluirse en la interfaz y se harán más fácilmente visibles según su grado de importancia. La posición de cada elemento de esta lista en la interfaz determina si el usuario será capaz de encontrarla fácilmente.

a. Importancia de la usabilidad y utilidad La usabilidad como la utilidad son igualmente importantes, y juntas determinan si algo es útil: es irrelevante que algo sea fácil de usar si no hace lo que se quiere. Así como algo que hipotéticamente hace lo que se quiere, pero no se es capaz de lograrlo debido a la dificultad de la interfaz. Tanto la usabilidad como la utilidad pueden estudiarse con los mismos métodos de interacción con usuarios (Nielsen, 2012).

- Utilidad = se proveen las **funcionalidades necesarias**.

- Usabilidad = qué tan **fáciles y agradables** de usar son estas funcionalidades.
- Útil = usabilidad + utilidad.

La importancia de usabilidad y utilidad en una interfaz de usuario es significativa, porque los usuarios tienden a abandonar una interfaz que encuentran difícil de utilizar. Por ejemplo, si el usuario no encuentra lo que necesita rápidamente, si se pierde, si la información escrita es muy difícil de leer o no se responden las preguntas clave del usuario. Esto significa que si al usuario no le agrada la interfaz o no la puede utilizar, las funcionalidades brindadas por el sistema se ignoran por completo. Entonces la interfaz termina siendo el punto fundamental de conexión entre el cliente y el sistema. Si falla en su objetivo, también falla todo el sistema.

b. Interfaz y experiencia de usuario Curiosamente, estas características se relacionan con dos términos muy conocidos en el área de diseño gráfico y mercadeo, **Interfaz de usuario** y **experiencia de usuario** (UI y UX, respectivamente por sus siglas en inglés), que buscan mantener la relación entre los usuarios y sus productos.

El diseño de una interfaz de usuario se relaciona con la característica de utilidad; permitir al usuario la interacción con las funcionalidades del sistema. La experiencia de usuario busca, además de cumplir con las necesidades y expectativas del usuario, una experiencia agradable en el uso del sistema (Norman & Nielsen, 2012), lo cual se relaciona con el concepto de usabilidad. Aunque las características de una interfaz útil no abarquen estos conceptos en su totalidad, es importante establecer su relación, y posteriormente encontrar metodologías e implementaciones en común.

2. Diseñar una interfaz de usuario útil Existen diversas metodologías para cumplir con las características de una interfaz de usuario útil. Es lógico pensar en la necesidad de incluir al usuario final en el proceso de diseño, porque de esta forma se puede evaluar el rendimiento de la interfaz en un ámbito real.

a. Metodología de pruebas con usuarios Esta metodología es la más básica y propone tres componentes principales (Nielsen, 2012):

- Contar con algunos usuarios representativos, como clientes o empleados (de preferencia fuera del departamento).
- Solicitar a los usuarios realizar tareas representativas con el diseño.
- Observar lo que hace el usuario, dónde tienen éxito, y dónde encuentran dificultades con la interfaz de usuario. No hablar y dejar al usuario expresarse.

Es importante realizar pruebas individuales con los usuarios y dejarlos resolver cualquier problema por su propia cuenta. Si se le brinda ayuda o se dirige su atención a una parte específica de la pantalla, se contaminan los resultados de las pruebas.

b. Estudio de usabilidad propuesto por NN/g (Nielsen Norman Group) La usabilidad juega un rol para cada etapa del proceso de diseño. La necesidad de realizar varios estudios es la razón por la cual este grupo recomienda realizar estudios individuales que sean rápidos y baratos de ejecutar. A continuación se describen los pasos a seguir (Nielsen, 2012):

1. Antes de empezar el nuevo diseño, probar el antiguo para identificar las partes buenas que deben permanecer o incluso enfatizar, y las partes malas que causan problemas al usuario.

2. A menos que se trabaje en una red interna, probar los diseños de los competidores para obtener información barata sobre el rango de interfaces alternativas que tienen funcionalidades similares al sistema propio.

3. Conducir un estudio de campo para ver cómo se comportan los usuarios en su hábitat natural.

4. Hacer prototipos en papel de una o más ideas de diseño y probarlas. Mientras menos tiempo se invierta formulando estas ideas mejor, porque se necesitarán cambiar basadas en los resultados de las pruebas.

5. Refinar las ideas de diseño para probarlas mejor en múltiples iteraciones, empezando por prototipos de baja fidelidad hasta llegar a representaciones altamente fieles ejecutadas desde una computadora. Probar cada iteración.

6. Inspeccionar el diseño a partir de *lineamientos establecidos de usabilidad*, basados en estudios previos o investigaciones.

7. Cuando se ha tomado una decisión y se implementa el diseño final, realizar una última prueba. Problemas sutiles de usabilidad ocurren durante la implementación.

c. Diseño centrado en el usuario Una forma de cumplir con estos dos requerimientos de usabilidad y utilidad es a través del **diseño centrado en el usuario** (*User Centered Design*). Existe un estándar ISO que indica los requerimientos de un diseño centrado en el humano, y que además utiliza factores humanos y de ergonomía, técnicas y conocimientos de usabilidad. Este estándar incluye un conjunto de principios del diseño centrado en el usuario, una planificación

para cumplir con estos principios en un sistema interactivo y un conjunto de actividades que deben llevarse a cabo con el usuario (ISO, 2010). Se basa en el manejo de hardware y diseño de software y sus principios son:

- El diseño está basado en un entendimiento explícito de los usuarios, tareas y ambientes.
- Los usuarios están involucrados durante el diseño y desarrollo.
- El diseño está encaminado y es refinado por las evaluaciones centradas en el usuario.
- El proceso es iterativo.
- El diseño incluye toda la experiencia de usuario
- El equipo de diseño incluye habilidades y perspectivas multidisciplinarias.

3. Tipos de interfaces de usuario Como se mencionó en la definición de interfaz de usuario, esta puede ser **virtual** (basada en software) y **física** (basada en hardware). Debido a la naturaleza del sistema que se desea implementar, en esta sección se analizan los tipos de interfaces de usuario virtuales. En este tipo de interfaz, generalmente se tiene una pantalla donde el usuario puede visualizar la información proporcionada por el sistema y se puede interactuar por medio de un componente físico (teclado, mouse, control remoto o la misma pantalla táctil), aunque en dispositivos modernos, ya se puede interactuar a través del movimiento o impulsos cerebrales. A continuación se describen los tipos de interfaces más comunes.

a. Interfaz gráfica de usuario La **interfaz gráfica de usuario** (*Graphical User Interface* - GUI, en inglés), es definida como el estilo de interacción humano-computador, basado en cuatro elementos fundamentales (Sakal, 2009):

- Ventanas
- Iconos
- Menús
- Punteros

También conocidos como WIMP (*windows, icons, menus, pointers,* en inglés). Estos elementos hacen una asociación semántica y funcional con la **manipulación directa**. Esta es, probablemente, la característica más importante de la GUI, y permite al usuario la interacción con objetos utilizando un dispositivo como puntero (un ejemplo típico incluye arrastrar-y-soltar, seleccionar objetos y texto con el mouse, crear objetos en un ambiente gráfico, etc). Muchas operaciones disponibles en el menú, también se pueden realizar a través de manipulación directa y gracias a esto,

el usuario experimenta al sistema como una extensión del mundo real (Sakal, 2009). Los objetos con los que el usuario interactúa generalmente existen en el mundo real y se encuentran visibles constantemente.

La GUI es propia de interfaces nativas, es decir, que es diseñada exclusivamente para el dispositivo que se está utilizando. Por lo tanto, se adapta al dispositivo sin problemas y muestra sus funcionalidades disponibles.

b. Interfaz web de usuario La interfaz web de usuario (*Web User Interface* - WUI, en inglés), es el tipo de interfaz que es diseñada exclusivamente para la navegación y presentación de información a través de una página web, donde la atención es prestada al contenido (Sakal, 2009). A diferencia de una GUI, en este tipo de interfaces el usuario no puede navegar entre ventanas o aplicaciones, sino desde un sitio hacia otro. El objetivo principal de una WUI es presentar información útil, pero actualmente se ha utilizado para muchos otros propósitos, como aplicaciones y portales web.

Debido al crecimiento en popularidad del internet, las WUI han sido la forma estándar de presentación de contenido. El diseño básico utilizado en la navegación de un sitio web, es la estructura de menú jerárquica (utilizada en entornos no gráficos), cuyo objetivo es la facilidad de uso y familiaridad con el usuario. Sin embargo, algunos diseñadores crean ambientes agnósticos donde (Sakal, 2009):

- Las aplicaciones presentan información a través del uso de varias ventanas del explorador.
- No hay íconos y componentes convencionales, sino que distintas aplicaciones utilizan distintos iconos y animaciones según el tipo de navegación o por estética.
- El puntero soporta únicamente la acción de selección por click y desplazamiento; la funcionalidad de arrastrar-y-soltar no está incorporada sistemáticamente.

Actualmente, la tendencia en el diseño de WUIs se ha desplazado hacia la creación de interfaces parecidas a una GUI, donde el usuario tenga la sensación de estar utilizando un sistema completo, relacionable a un ambiente de la vida real. A partir de este pensamiento, han surgido varios tipos de WUI, según el tipo de sitio. Estos se describen a continuación (Sakal, 2009):

- **Sitios orientados a la información:** su intención es la presentación de información, sin necesidad de procesar transacciones. La interfaz de estos sitios equilibra la necesidad de presentar información útil con la de permitir a nuevos visitantes aprender y navegar fácilmente.

- **Aplicaciones web:** su objetivo es proporcionar al usuario un sistema completo con el cual interactuar para satisfacer sus necesidades. Su interfaz busca parecerse a la de una GUI donde el usuario sienta control del ambiente, a pesar de acceder a través de un navegador.

- **Portales web:** este tipo de sitios buscan crear una conexión entre el usuario y otras secciones del mismo sitio o sitios externos, al proporcionarle un enlace directo. Su interfaz debe ser sencilla, y funciona como un catálogo en donde el usuario puede elegir un enlace dependiendo de la búsqueda que haya realizado o de la conFiguración de un filtro. Hoy en día, se les conoce como Portales de búsqueda o navegación.

c. Interfaz móvil Este tipo de interfaces incluyen todas las interfaces diseñadas especialmente para dispositivos móviles, como teléfonos inteligentes (smartphones) y tabletas. Hoy en día, el uso de smartphones se ha convertido en una necesidad, más que en una tendencia. Las personas lo utilizan para tareas diarias, como comunicación, agenda, pagos, etc. Por lo tanto, el diseño de interfaces móviles ha adquirido gran importancia en el mundo de la interacción humano-computador. A pesar de que sus principios buscan familiaridad con el usuario (como en una GUI), la interfaz móvil cuenta con varias diferencias de interacción, por la singularidad del dispositivo (Web Design, 2012):

Cuadro 18: Características en tipos de dispositivos

Dispositivo móvil (Smartphone, tableta)	**Dispositivo de escritorio**
Pantalla pequeña	Pantalla grande
Conectividad intermitente	Conectividad confiable
Ancho de banda bajo	Ancho de banda alto
Energía por batería	Conexión directa

A pesar de las limitantes descritas en el Cuadro 18, los dispositivos móviles poseen ciertas características únicas y útiles. Son sumamente personales y transportables, usualmente conectados y direccionables. Además poseen sensores de localización, movimiento, aceleración, orientación, proximidad, condiciones ambientales, entre otros (Web Design, 2012). Estas características proporcionan una mayor variedad de posibilidades en el diseño de interfaces, así como limitantes.

Cuando se diseñan interfaces móviles se toma en cuenta el tipo de sistema de software que se desea presentar, generalmente web o aplicación.

- **WUI adaptable:** En el mundo de las WUIs, resulta necesario diseñar interfaces capaces de adaptarse según el tamaño del dispositivo en cual se navega. A esta adaptabilidad se le conoce como diseño adaptable (*responsive*). Se podría decir que están especialmente diseñadas para la visualización en dispositivos móviles. Aunque son una excelente alternativa para mostrar

la web en un tamaño apropiado, este tipo de interfaces no sustituyen la convención del diseño nativo.

- **Interfaz móvil nativa:** este tipo de diseño abarca la implementación de aplicaciones nativas (propias de la plataforma), para mostrar la interfaz del sistema. Sus mayores ventajas son la rapidez y las convenciones propias de la plataforma que permiten mayor familiaridad para el usuario. Su mayor desventaja es el desarrollo multiplataforma, que implica un gasto mayor de recursos.

Cada una de los tipos de interfaces tiene sus características, ventajas y desventajas. En el Cuadro 19 se resumen las más importantes.

4. Plataformas de desarrollo Debido a las características del sistema que se desea implementar (multiplataforma, conectividad a internet, presentación de información y red social) a continuación se describen y analizan plataformas de desarrollo web para el desarrollo de la interfaz de usuario. Cada plataforma debe tener la capacidad de proporcionar las herramientas y componentes para cumplir con los objetivos de la interfaz de usuario útil: usabilidad y utilidad. Las plataformas analizadas tienen como objetivo el desarrollo *frontend*, que abarca el lado del sistema que interactúa con el cliente y son de código abierto (*open source*).

a. AngularJS Nació en 2009, como parte de un proyecto más grande. Fue convertido a código abierto y actualmente es patrocinado por Google. Entre sus funcionalidades principales y únicas están enlace de datos de doble canal (two-way data binding), inyección directa de dependencias, código fácil de probar y la extensión de HTML (HyperText Markup Language) a través de directivas (Shaked, 2013). Su comunidad es la más extensa entre las plataformas del lado del cliente, pues cuenta con la mayor cantidad de estrellas y contribuyentes en Github, la mayor cantidad de módulos de terceros, tutoriales, extensiones y preguntas resueltas.

b. Backbone.js Es una plataforma MVC (Modelo, Vista, Controlador) de bajo peso. Nació en 2010 y rápidamente creció en popularidad como una alternativa a otras plataformas pesadas del momento (Shaked, 2013). Su popularidad ha disminuido desde que aparecieron plataformas con mayores funcionalidades e igualmente sencillas de utilizar.

c. Ember.js Sus raíces se remontan hacia 2007, a través de la plataforma SproutCore MVC, de Apple (Shaked, 2013). Sin embargo, su lanzamiento oficial se dió en diciembre de 2011. La comunidad de Ember.js no es la más grande (por ser relativamente nuevo), pero ha ido creciendo constantemente.

Cuadro 19: Ventajas y desventajas para cada tipo de interfaz de usuario

Tipo de interfaz	Ventajas	Desventajas
GUI	▪ Reduce la necesidad de percibir y recordar cosas en la interacción y aumenta la velocidad de transferencia de información (Web Design, 2012). ▪ Es predecible, familiar y provee una sensación de control. ▪ Disminuye el rango de errores cometidos por el usuario debido a sus múltiples formas de completar tareas.	▪ La cantidad de elementos en pantalla, como iconos y ventanas puede resultar abrumador para algunos usuarios. ▪ El diseño es propio para cada plataforma o sistema operativo. Por lo tanto, varía para cada sistema y no se sigue una convención unificada. ▪ Es necesario actualizar el programa para obtener la última versión de la interfaz.
WUI	▪ Es el tipo de interfaz más popular, por lo tanto abunda la cantidad de información, herramientas y componentes. ▪ El diseño es multiplataforma, es decir, se visualiza el mismo tipo de interfaz en cualquier dispositivo. (Si se utiliza un diseño responsive, también en móviles) ▪ Flexibilidad en el diseño. Es posible imitar las características de cualquier otro tipo de interfaz. ▪ No es necesario actualizar cada dispositivo para obtener la versión más reciente de la interfaz.	▪ Generalmente depende de una conexión a internet. Además, esta condición puede hacer más lenta la interacción. ▪ Ha sido adaptada para la creación de aplicaciones, pero no es su función principal. Por lo tanto, se deben incluir herramientas adicionales. ▪ Se necesita un navegador para ser desplegada, y la visualización varía entre cada navegador.
Interfaz móvil	▪ Diseño nativo, por lo tanto se siguen convenciones según la plataforma. ▪ Diseño especialmente creado para dispositivos móviles, por lo cual se maximiza su visualización y rendimiento. ▪ Disponible como una aplicación dentro del dispositivo y no como un sitio web.	▪ Las convenciones también presentan la desventaja de poca flexibilidad. ▪ Costo elevado en el desarrollo individual para cada plataforma móvil. ▪ La aplicación debe ser autorizada por la tienda oficial de la plataforma.

5. Comparativa entre plataformas de desarrollo web Las plataformas tienen características similares pero es necesario establecer la que mejor se adapta a las necesidades del sistema. Por esta razón, se hará una comparativa entre ellas.

a. Comunidad La comunidad de desarrollo es un factor importante a considerar, porque significa más información para aprender, desarrollar y resolver problemas. Las plataformas han tenido un impacto global en la comunidad de desarrollo. Y una de las formas de medir este

impacto es a través de la cantidad de usuarios activos en línea. El Cuadro 20 muestra estadísticas sobre el impacto que la plataforma ha tenido en la comunidad de desarrollo, hasta junio de 2015.

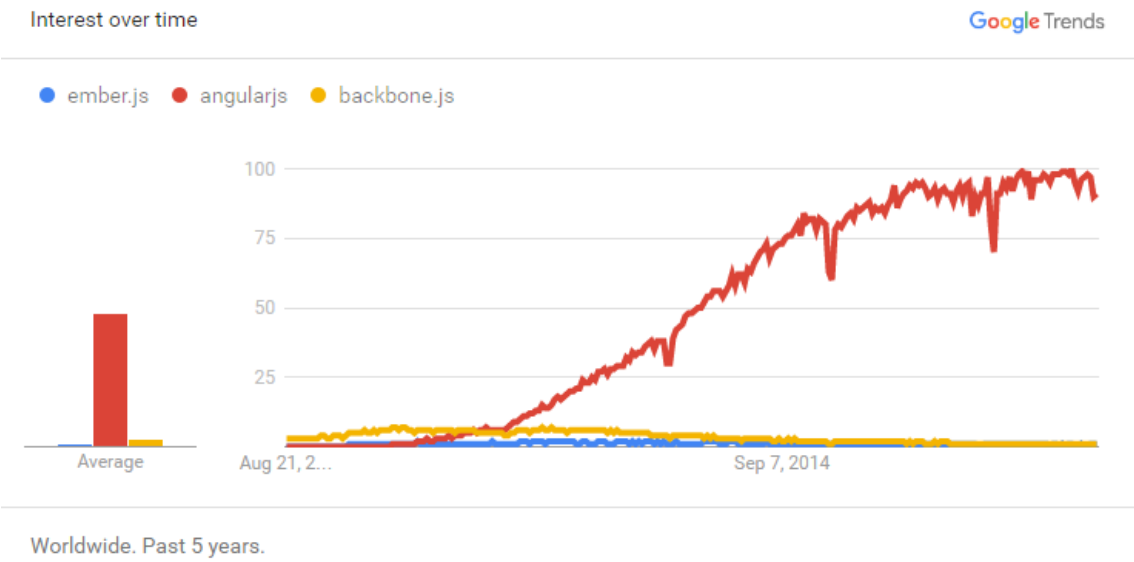
Cuadro 20: Comparativa plataformas de desarrollo web. Comunidad

Métrica	**AngularJS**	**Backbone.js**	**Ember.js**
Estrellas en Github	40.2k	18.8k	14.1k
Módulos de terceros	1488	256	1155
Preguntas en StackOverflow	104k	18.2k	15.7k
Resultados en YouTube	93k	10.6k	9.1k
Colaboradores en Github	96	265	501
Problemas abiertos en Github	922	13	413
Problemas cerrados en Github	5,520	2,062	3,350

(Shaked, 2013)

Esta comparación hace notar la presencia en línea para cada una de las plataformas y la comunidad de desarrollo detrás de ella. La Figura 3 muestra una gráfica que indica el grado de crecimiento en popularidad en las búsquedas de Google para cada una de las plataformas.

Figura 3: Comparativa plataformas de desarrollo web. Tendencia en las búsquedas de Google



(Google Trends, 2016)

A la hora de seleccionar una plataforma, sus características y funcionalidades básicas son de las

primeras cosas a analizar. En el Cuadro 21 se hace una descripción de las mismas. Es importante porque ilustra si las herramientas con las que cuenta son suficientes para cumplir con los objetivos del módulo.

Cuadro 21: Comparativa plataformas de desarrollo web. Características

Característica	**Backbone.js**	**Ember.js**	**AngularJS**
Tipo de plataforma	MV + VC (*Model, View + View Controller*)	MVC (*Model, View, Controller*) puro	MVW (*Model, View, Whatever*)
Plantilla de Apoyo (*Template Support*)	Utiliza underscore.js (permite incluir lógica adentro del código de plantilla)	Utiliza Handlebars.js (admite lógica pero de forma limitada)	Si proporciona una plantilla de apoyo incluida, sin requerir ni permitir soporte para otras.
Plantilla de Apoyo Incluida	No	Si, pero solo con Handlebars.js	Si
Injección automática de código (*Auto Binding*)	No	Si, pero solo con Handlebars.js	Si
Enrutamiento (Routing)	Si	Si	Si
Dependencias	underscore.js	Handlebars.js y Jquery 1.7	No
Compatible con otras plataformas	Si	Si	Si
Funcionalidades adicionales		Propiedades preprocesadas, formateo de datos	DI (*Dependency injection*), directivas, observadores de expresiones y valores (*watchers*), validaciones en HTML5.

(Shan, 2013)

También es importante comparar las plataformas de acuerdo a sus ventajas y desventajas, porque aunque tengan las características deseadas es probable que funcionen de manera diferente entre sí. Y aunque no sea posible encontrar las características perfectas, debe seleccionarse la que mejor se ajuste a los objetivos del sistema. En el Cuadro 22 se describen las características que más destacan y que más se esperan en cada plataforma.

Despues de seleccionar una herramienta de desarrollo para interfaces de usuario, es el momento de seleccionar un tipo de diseño en la cual se presentará la información. Esta selección es fundamental para cumplir con los objetivos de usabilidad y utilidad en la interfaz de usuario, pues de ella depende si un usuario será capaz de utilizar eficientemente el sistema y si hará uso de todas las funcionalidades.

Cuadro 22: Comparativa plataformas de desarrollo web. Ventajas y desventajas.

Plataforma	Ventajas	Desventajas
AngularJS	▪ Código de plantilla simple y legible (Web Design, 2012). ▪ Comunidad más grande e innovadora. Además, es promovido por Google. ▪ Permite organizar la aplicación utilizando diferentes partes: controladores, directivas, servicios, filtros y vistas (templates). Y todo esto dentro de módulos que pueden comunicarse entre sí. ▪ Fácil de probar, gracias a la separación de funciones.	▪ Curva de aprendizaje elevada, con nuevos conceptos como directivas, reglas y convenciones creadas exclusivamente para AngularJS. ▪ Ciclo de ejecución complicado de entender. ▪ A veces demasiada libertad en el código de la plantilla, que lo hacen difícil de probar.
Ember.js	▪ Favorece convención sobre configuración, es decir, que no es necesario definir toda la configuración. ▪ Excelente sistema de enrutamiento y una capa opcional de datos. ▪ Características que favorecen el rendimiento, como el ciclo de ejecución y preprocesamiento.	▪ Ha cambiado mucho con el tiempo y existe mucha documentación desactualizada. ▪ Handlebars.js hace que la plantilla no sea fácil de leer y causa confusión.
Backbone.js	▪ Es ligero, rápido y utiliza poca memoria. ▪ Curva de aprendizaje baja. ▪ Utilizado en muchas plataformas de terceros.	▪ No provee estructura ni control de memoria y deja a discresión del desarrollador cómo estructurar el proyecto y manejar la memoria. ▪ Hay muchas opciones para cubrir las funcionalidades que le faltan, lo que implica investigación y gasto de tiempo ▪ No posee transferencia de datos ni hace fácil realizar pruebas, como AngularJS y Ember.js

(Web Design, 2012)

6. Teoría del color Una de los pilares en el diseño de intefaces de usuario es una buena elección de colores, porque esto determina la usabilidad de la plataforma y la experiencia que el usuario tiene al utilizar la aplicación. Por eso es necesaria una introducción a la teoría del color, que muestre los principios básicos y propiedades del color cuando se diseña una interfaz de usuario.

La teoría del color engloba una multitud de definiciones, conceptos y aplicaciones en diseño. Sin embargo, existen tres categorías básicas en la teoría del color que son lógicas y útilies: la paleta de colores, la armonía de colores y el contexto de cómo se utilizan los colores. En general, la teoría

de colores crea una estructura lógica para clasificación de los colores (Morton, 1999).

a. La paleta de colores Es un círculo tradicional de colores en rojo, amarillo y azul. Isaac Newton fué el primero en desarrollar un diagrama circular de colores en 1,666. Desde entonces, se han desarrollado muchas variantes de este concepto (Morton, 1999). En la Figura 4 se pueden apreciar estas variantes.

Figura 4: Teoría del color. Paleta de colores

También existen definiciones o categorías de colores basados en la paleta de colores, empezando por los colores primarios, secundarios y terciarios (Morton, 1999). Esta clasificación se aprecia en la Figura 5.

Figura 5: Teoría del color. Categorías de colores

- **Colores primarios:** Rojo, amarillo y azul. En la teoría tradicional del color, los colores primarios son los tres colores básicos que no pueden mezclarse o formarse con alguna otra combinación de colores. Los demás matices se forman a partir de estos.
- **Colores secundarios:** Verde, naranja y púrpura. Estos se forman a partir de la mezcla de colores primarios.
- **Colores terciarios:** Amarillo-naranja, rojo-naranja, rojo-púrpura, azul-púrpura, azul-verdoso y amarillo-verdoso. Estos colores se forman a partir de la mezcla entre colores primarios y secundarios.

b. Armonía de colores La **armonía** puede ser definida como una combinación placentera de partes, ya sea en la música, poesía o color. En la experiencia visual, la armonía es algo placentero a la vista, atrapa al usuario y le da una sensación de orden. Cuando algo no es armónico, puede ser aburrido o incluso caótico. Por un lado está una experiencia visual tan débil,

que el usuario no se sentirá estimulado. Por el otro, hay una experiencia visual tan fuerte y caótica, que el usuario no puede soportar mirarla. El cerebro humano rechaza lo que no puede organizar o entender. La tarea requiere la presentación de una estructura lógica. Existen algunas fórmulas para alcanzar este equilibrio (Morton, 1999).

- **Esquema basado en colores análogos:** Son cualquiera de tres colores adyacentes en una paleta de 12 colores, donde uno de ellos es el predominante.
- **Esquema basado en colores complementarios:** Son cualquiera de dos colores directamente opuestos en una paleta de colores, además de contener sus adyacentes.
- **Esquema basado en la naturaleza:** La naturaleza provee un punto perfecto de armonía del color.

Figura 6: Teoría del color. Colores análogos

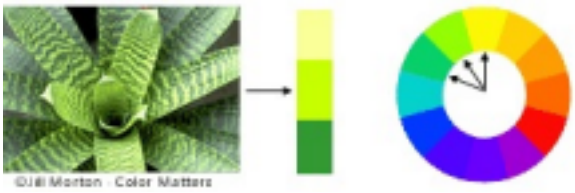


Figura 7: Teoría del color. Colores complementarios

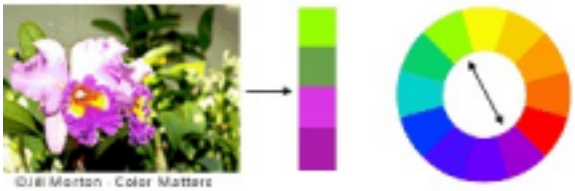
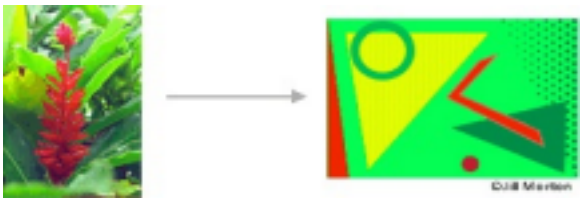


Figura 8: Teoría del color. Colores complementarios



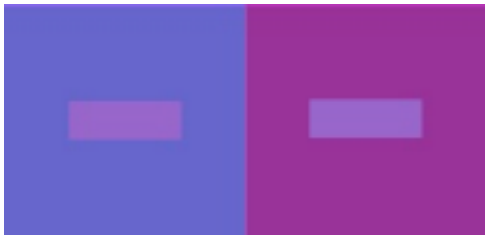
c. Contexto del color Según el contexto, el color se puede comportar de diferentes maneras. Esto depende en el efecto del contraste entre diferentes colores de fondo. La Figura 9 muestra un ejemplo del significado del rojo bajo distintos fondos. El rojo parece más brillante sobre un fondo negro, un poco más apagado contra un fondo blanco. Parece sin vida, bajo un fondo

naranja y lleno de esta bajo un fondo acua (Morton, 1999).

Figura 9: Teoría del color. Contraste de colores

Un mismo color puede parecer distinto según el contexto en que se encuentre y esto ocasiona diferentes lecturas del mismo color. La Figura 10 muestra este fenómeno. El mismo color parece más rojizo en un fondo azul, y más azulado bajo un fondo rojizo. Esto da la percepción de cuatro colores, cuando sólo existen tres (Morton, 1999).

Figura 10: Teoría del color. Diferentes lecturas del mismo color (Morton, 1999)



d. Herramientas de Selección de color Existen herramientas disponibles en la web, para seleccionar esquemas de colores que trabajen armoniosamente entre sí. Una de estas es la herramienta de Adobe CC, la cual permite seleccionar diferentes configuraciones de esquemas y sus identificadores como colores digitales. La Figura 11 muestra esta herramienta en funcionamiento.

e. Colores en *Material Design* Existe una guía para la selección de colores en el tipo de diseño *Material Design*. Su paleta de colores se deriva de una serie de señales de arquitectura contemporánea, señales de tránsito, cinta para marcar pavimento y pistas de atletismo, dando una sugerencia de colores inesperados y vibrantes. Esta paleta de colores se compone de colores primarios y de acento, que utilizan una gama de colores para identificar a una marca. Han sido diseñados para integrarse armoniosamente entre ellos (Google, 2013).

Para utilizarlos se debe empezar por la selección un color primario, y un color secundario o de acento que genere suficiente contraste con el color primario para hacerlos trabajar de forma armoniosa. Cada color, conlleva una serie de matices que pueden ser utilizados a lo largo del diseño de una interfaz para generar suficiente contraste y placer en su utilización.

La paleta de colores está especialmente seleccionada, así como sus matices. La Figura 12 muestra la selección de colores utilizados en *Material Design* y da una idea a la gama de matices que pueden utilizarse en para cada color primario o de acento.

Figura 11: Herramientas de selección de color. Adobe CC

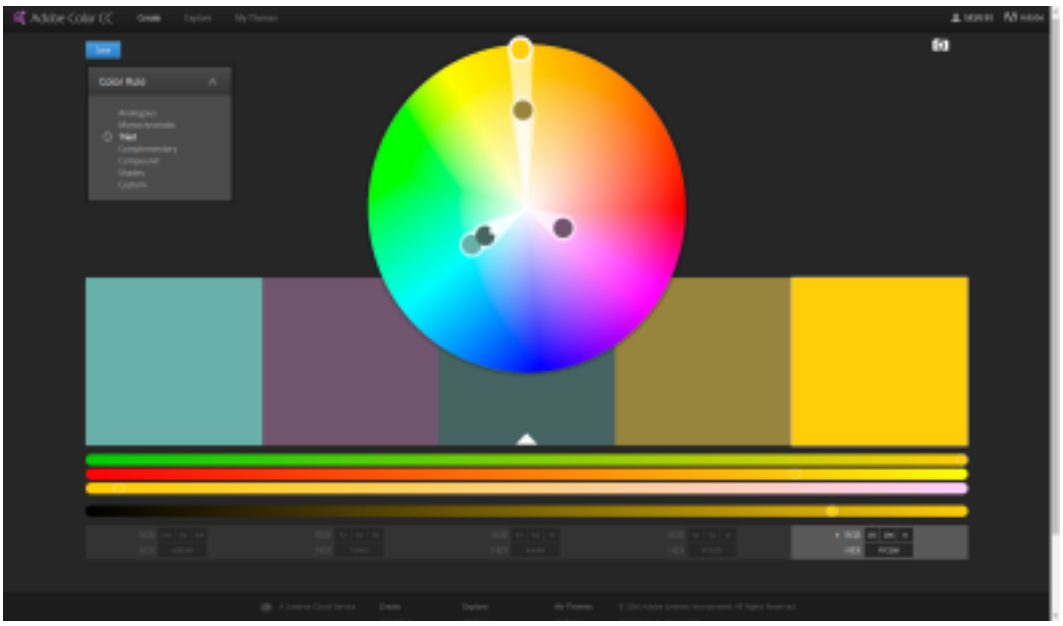
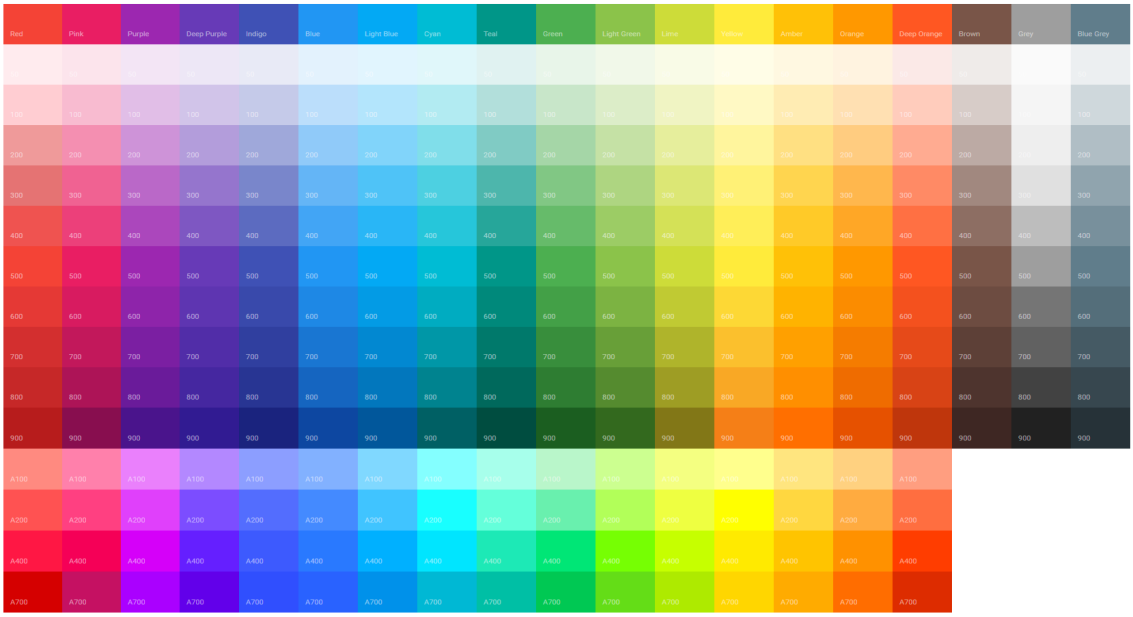


Figura 12: Material design. Selección de colores



V. Marco metodológico

A. General

La fase inicial del proyecto consistió en la recolección de datos para identificar algún problema presente en hogares u empresas guatemaltecas en áreas relacionadas a la energía eléctrica. Para esto, se realizaron entrevistas a personas de diversa procedencia relacionadas al tema en cuestión. Se entrevistó desde padres y madres de familia a personas independientes, con diferentes áreas de trabajo que abarcaron desde empresarios, maestros hasta especialistas del área de electricidad tanto en el área técnica como en el área de ingeniería. Dichas entrevistas fueron basadas en la metodología de Design Thinking, dando como resultado, después de su análisis, la identificación del problema de la falta de control en el consumo de energía.

En el proyecto se realizaron alrededor de cuarenta entrevistas a usuarios de la red eléctrica de Guatemala. De estas se obtuvo información sobre lo que hacen para reducir su consumo eléctrico (según su propia experiencia o consejos de otros usuarios). También fue notable las diferentes preocupaciones de los usuarios, siendo la más importante el desconocer un método que funcione para controlar su consumo, y este es el problema que se desea resolver.

Se realizaron investigaciones sobre cómo en otros países han abordado dicha situación, investigando casos como el de Chicago y Abu Dhabi. Se procedió a identificar qué tipos de herramientas podrían ser utilizadas y qué otras ya existían en el mercado como las ofrecidas por Scheider Electrics.

En la búsqueda de una solución se hicieron nuevas entrevistas para reunir ideas y poder plantear una solución al problema. En esta nueva fase de entrevistas la mayoría de usuarios comentaron que el conocer el consumo a la fecha sería de utilidad. Algunos comentarios dieron a conocer que sabiendo el consumo a la fecha se puede hacer ajustes como: no utilizar el calentador de agua, no utilizar la secadora de ropa, reducir el uso de la plancha de vapor, entre otros.

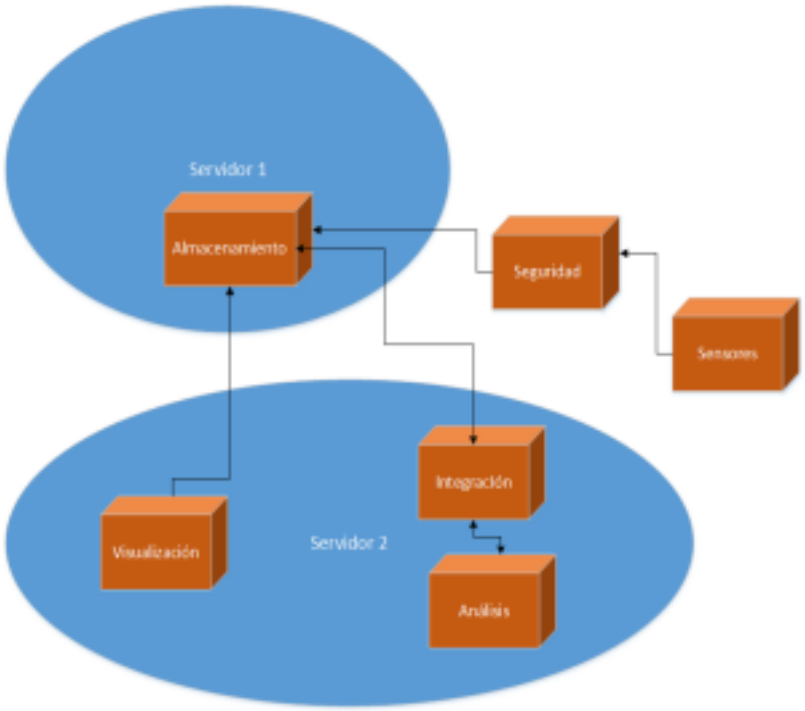
Los resultados de las entrevistas en general, muestran que a los usuarios de la red eléctrica les sería util tener una herramienta que monitoreara el consumo eléctrico en tiempo real. Luego se mencionaron ideas como predecir el consumo total a fin de mes basado en el consumo a la fecha.

Se examinó el tipo de soluciones que se podían implementar dentro de la realidad guatemalteca, concluyendo en el desarrollo de una herramienta que ayude a controlar el consumo de energía de los

usuarios, siendo un sistema de monitoreo eléctrico que utilice un sensor para recolectar los datos de consumo energético y los haga llegar al usuario en tiempo real. Además de brindarle facilidades como la predicción de su consumo de energía eléctrico basado en sus patrones de uso.

Para llevar a cabo el desarrollo del sistema, el proyecto fue dividido en seis módulos. Cada módulo desempeña una función específica dentro del proyecto y se relacionan de esta forma: el módulo de sensores se encarga de hacer las mediciones, recolectar los datos sobre el consumo y enviárla a un sistema de almacenamiento en la nube. El módulo de seguridad, se encarga de proteger la información para que nadie pueda interceptarla y leerla, y que llegue de manera íntegra al servicio de almacenamiento. El módulo de almacenamiento y servicios web, se encarga de almacenar la información enviada por los sensores en un servidor central en la nube, y además, hace la información disponible a través de servicios web para su acceso remoto. En otro servidor, existen los módulos de análisis de datos, integración y visualización. Es el módulo de integración el encargado de comunicar ambos servidores, de forma interna. De esta forma, el módulo de análisis de datos puede acceder a ellos, y hacer un análisis predictivo y de clustering sobre los datos de consumo, que luego son enviados de vuelta al servicio de almacenamiento. Por último, el módulo de visualización se encarga de acceder a los datos de consumo a través de los servicios web y los muestra al usuario final, por medio de una interfaz web y componentes gráficos, que representan las funcionalidades del sistema. En la figura 13 se muestra la relación que existe entre cada módulo.

Figura 13: Diagrama General del Sistema



Aunque fue necesario el mutuo acuerdo entre módulos, cada uno se desarrolló de forma independiente, satisficiendo sus objetivos específicos, para al final, satisfacer el objetivo general. A continuación se presenta la metodología seguida por cada uno para cumplir su función dentro del proyecto.

B. Sensores y protocolos

Para realizar la implementación, el proyecto se dividió en distintas secciones. Puesto que las secciones son dependientes unas de otras se organizaron de la siguiente manera:

- Selección de sensor de voltaje.
- Selección de sensor de corriente.
- Selección del protocolo de comunicación.
- Implementación del módulo.

El proceso de transición de una sección a la siguiente dependió de los resultados obtenidos y de cómo estos afectaron directamente a las secciones futuras. Por esto, las primeras tres secciones fueron las que presentaron menor nivel de dificultad pero su correcto desarrollo y análisis fue vital para asegurar que el desarrollo de la última fase fuera más sencillo. Se realizó un cronograma de actividades con el fin de lograr los objetivos del proyecto en el tiempo estimado. Dicho cronograma se presenta a continuación.

El diagrama de bloques de las secciones que componen el módulo se puede observar a continuación. Cómo las secciones anteriores eran dependientes se realizaron cronológicamente. De igual forma y para facilitar la comprensión, los resultados de cada fase se acompañan de su discusión y análisis.

C. Seguridad de la información

1. Investigación La fase inicial del módulo de seguridad, fué la investigación de los distintos tipos de cifrado de información y los algoritmos existentes para cada una de ellas; que encajen más a las necesidades del proyecto. Se analizaron cuáles podrían ser más funcionales y eficientes; para los tipos y los distintos tamaños de información recopilada. Además, fue necesario saber los ataques existentes, si el algoritmo ha sido atacado en su totalidad o si ha sido atacado parcialmente, y las distintas herramientas para poder realizar ataques.

2. Selección En base a los algoritmos investigados, se seleccionaron algunos de ellos para el análisis con los datos recibidos. Esta selección involucró distintos criterios para comparación; según su arquitectura, seguridad, ataques conocidos, tiempos y cantidad de memoria consumida

en sus operaciones.

3. Implementación Luego de haber completado la selección del algoritmos a utilizar, se realizó la implementación de cada uno de ellos, en el lenguaje de programación Python. Se eligió Python para la implementación, debido a que es un lenguaje conocido, teniendo mucha documentación y herramientas de uso libre. Dentro de esta sección, se comparó cada uno de los tiempos (en milisegundos) de la construcción de llave criptográfica, cifrado y descifrado; utilizando una diferencia de tiempos de inicio y fin en cada operación. También se realizó comparaciones de memoria (en megabytes), utilizando una herramienta gratutita de Python llamada *memory_profiler* para cada respectivo proceso. Cada algoritmo utilizó la misma información y llave. La información utilizada fueron datos ficticios en diferentes cantidades, siendo estas 1, 10, 100, 1000, 10000 y 100000 bytes.

Además, en esta sección se realizó la implementación de los algoritmos asimétricos en base a un tamaño de llave definido. Se analizó la cantidad de tiempo y memoria consumida por cada uno de los procesos de estos algoritmos; generación de llave pública y privada, cifrado y descifrado. El texto de ingreso para estos algoritmos será la llave criptográfica generada para los algoritmos simétricos.

4. Desarrollo Finalmente se empezó la sección de desarrollo; donde en base al algoritmo con mejores características y pruebas conforme a los criterios de seguridad elegidos. Al algoritmo seleccionado; se tomó para utilizarlo dentro del proyecto. Se usará el lenguaje de programación Python para la parte de Raspberry, y en el lenguaje de programación NodeJS para la parte del servidor.

5. Pruebas Al haber implementado el algoritmo seleccionado, se empezó a desarrollar la sección de pruebas. La información se envió y se recibió a través de un canal de comunicación inseguro por medio del internet. Se determinó que el texto cifrado al momento de descrifrarlo sea el original y no haya tenido ningún problema de conversión, utilizando funciones Hash.

6. Pruebas utilizando ataques Esta sección se basó en la utilización de herramientas de ataque; principalmente herramientas de Sniffing, para determinar si el algoritmo de cifrado fué elegido correctamente. Se utilizó el sistema operativo Kali Linux, el cuál es un una extensión de Debian, Linux; específico para pruebas de penetración y ataques de seguridad informática.

D. Almacenamiento de información y servicios web

1. Investigación Para empezar el módulo de API se investigó sobre que es una REST web API y que herramientas y frameworks facilitan su desarrollo. Se listaron los requerimientos de una REST web API y se analizó cada uno para encontrar cuales tenían mayor importancia y por ende mayor prioridad. También fue necesario investigar distintos DBMS y observar cuales cumplian los requerimientos para el manejo de datos.

2. Selección Como siguiente paso se eligió el framework de desarrollo que se adaptaba mejor a los requerimientos de eficiencia. En este paso también se eligierón los DBMS a utilizar. Para seleccionar las herramientas se tomaron en cuenta los objetivos del sistema así como lo investigado en el marco teórico. Para determinar qué información se estaría almacenando y en que DBMS se hizo lo siguiente:

- **Listar todas las partes:** Se identificaron todas las entidades que estarían interactuando del sistema así como la información que describe a cada una.
- **Dibujar un diagrama entidad relación:** Tomando las entidades identificadas se establece una relación entre ellas utilizando un diagrama entidad relación.
- **Asignación de entidades a DBMS:** Dependiendo de la cantidad de información esperada para una entidad esta se asignó al DBMS que mejor cumpliera los requerimientos.

3. Implementación Para iniciar la implementación se realizaron esquemas de los datos con los que se trabajó, y las acciones que interactuaron con los mismos. Los esquemas de datos se armaron en base a los documentos generados en la fase anterior.

4. Pruebas generales Para terminar el desarrollo fue necesario realizar pruebas de carga y eficiencia sobre las acciones implementadas sobre la web API.

E. Integración

La metodología del módulo de integración se puede visualizar en la Figura 14.

1. Investigación Para empezar el módulo de integración se investigó acerca de qué es una integración de sistemas. Luego de entender los requerimientos que tiene una integración de sistemas se estudió técnicas o métodos que se puedan aplicar cuando se esta realizando una integración de sistemas. Esta fase esta enfocada al análisis de los métodos o técnicas encontradas y buscar su uso en la integración.

2. Selección En esta fase se realizó la selección de los métodos y técnicas investigadas que se utilizaron en este módulo. Se tomó en consideración más de una técnica o método, en aras de elegir métodos y técnicas que permitan unificar el sistema. Esta elección se hizo tomando en cuenta los objetivos de este módulo así como lo investigado en el marco teórico.

Para la comunicación con el módulo de Almacenamiento de Información y Servicios Web se utilizó la integración remota. Esta comunicación se realiza utilizando JSON para la transmisión de mensajes, debido a que era más conveniente para ambos módulos y es más eficiente que XML. Por otro lado, para la comunicación con Análisis se utilizó mensajería porque permite enviar información y olvidarse de ella, permitiendo que cada módulo siga ejecutando sus procesos.

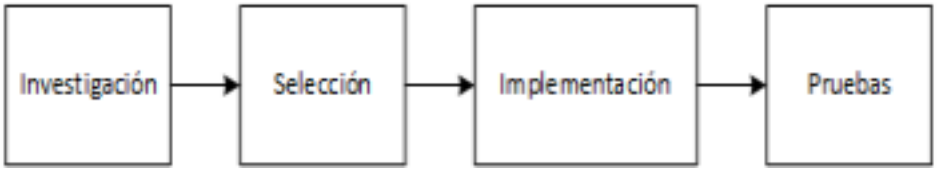
3. Implementación Para iniciar la implementación se eligió un lenguaje de programación o plataforma en el que se estableció el ambiente para unificar los diferentes módulos del proyecto. Esta elección esta basdada en la investigación realizada en el marco teórico, incluyendo aspectos de los diferentes lenguajes de programación y plataformas, así como las diferentes técnicas y métodos que se elegieron en el paso anterior.

Se consideraron tres plataformas: Spring Integration, Mule y Apache Camel. Spring Integration funciona sobre Java Virtual Machine, esto implica que la integración con lenguajes de programación diferentes a Java, es más compleja que en otras plataformas. Mule y Apache Camel son plataformas que implementan patrones de integración. Mule, a diferencia de Apache Camel, no permite utilizar servicios de terceros y tiene una comunidad más pequeña. Apache Camel puede trabajar con otras tecnologías que se usan en el ambiente de integración, entre estas están ActiveMQ, JMS y JBi. Se utilizó Apache Camel porque tiene un mayor potencial de crecimiento, tiene una mayor comunidad, es flexible en el sentido que permite utilizar otras tecnologías de integración.

Una vez se eligió la plataforma para la integración se estableció el ambiente y se empezó a leer la documentación sobre esta plataforma para familiarizarce con la misma. Se establecieron acuerdos con los módulos involucrados en la integración. En estos acuerdos se encuentra la información que se esperá recibir y enviar a estos módulos.

4. Pruebas Esta fase consta de pruebas locales y pruebas con los módulos terminados. Las pruebas locales se hicieron de manera independiente a los otros módulos. En estas pruebas se verificó que la información sea transmitida usando los protocolos que se establecieron en los acuerdos. Las pruebas con los módulos terminados se realizaron cuando los módulos involucrados terminaron la parte que se usará dentro de este módulo. Esto se realizó a través de realizar pruebas donde se transmitió la información a través de los módulos, siguiendo lo establecido en los acuerdos.

Figura 14: Metodología utilizada por el módulo de integración



F. Análisis de datos

1. Investigación Se estudiaron diversos casos de estudio similares a los que se presenta dentro de este proyecto, buscando específicamente en artículos científicos de disposición pública no solamente en búsquedas primarias en Google, sino también los que se encuentran a disposición en la biblioteca virtual de la ACM, y además los disponibles gratuitamente en el portal ResearchGate. Dentro de los casos estudiados se buscó entender qué se había hecho y cómo se había realizado cada uno de los análisis de datos de consumo eléctrico ya sea para realizar análisis de predicción o bien de clasificación de los datos obtenidos. Se observó también qué tipos de algoritmos se habían utilizado para cada uno de los tipos de análisis ya especificados. Se observaron diferentes metodologías basadas en formas estadísticas o regresiones así como también algoritmos del área de aprendizaje de máquina. A partir de esta observación de métodos se procuró buscar cuales eran los métodos más comunes usados, y así mismo observando qué tipo de resultados obtuvieron con los métodos aplicados. De igual manera se puso atención a los datos que se usaban en las diferentes investigaciones leídas y cómo era que se abordaban los problemas de datos que estas tenían, así mismo qué aspectos de los datos resultaban importantes luego de realizar un análisis básico.

2. Selección de métodos En la siguiente fase del área de análisis de datos se pasó a realizar la selección de los algoritmos o métodos que se emplearían tanto para lo referente a predicción como también para la clasificación de los datos de consumo de energía eléctrica. Se tomó como criterio la sugerencia dada por los autores de las diferentes publicaciones científicas tomando en cuenta la justificación que estos daban para cada una de las conclusiones que presentaban. Así mismo se tomaron en cuenta las especificaciones que deseaban desarrollar durante el proyecto considerando complejidad computacional y rapidez de los algoritmos. También se consideró la frecuencia con que estos métodos han sido usados en estudios similares.

3. Selección de lenguaje Para la selección del lenguaje que se utilizaría para poder desempeñar los métodos deseados, se buscó uno que tuviera librerías que ayudaran al desarrollo del proyecto, tomando en cuenta el apoyo y soporte que se le den a las mismas. Se consideró la popularidad del lenguaje dentro de la comunidad de análisis de datos. De igual manera, se tomó

en cuenta el uso que se le da dentro de compañías grandes en el área relacionada al análisis de datos, como la de IBM (IBM Analytics, 2016).

4. Implementación En lo referente a la implementación de los algoritmos se realizó un acercamiento orientado a objetos dentro del lenguaje seleccionado. Dejando como variables de clases aquellas necesarias para buscar una mejor escalabilidad básica de parte de los algoritmos, como el nombre de las columnas a ser leídas y el nombre del archivo donde se encontraran los datos. De igual modo se buscó realizar métodos independientes para cada uno de los pasos importantes dentro de los algoritmos, además de fase de entrenamiento en los casos que son necesarios, también el realizar procedimientos propios del tipo de algoritmos implementados como la clasificación o bien predicción. Se buscó que todos los algoritmos fueran controlados por un solo programa principal el cual se encarga de recibir la solicitud y de devolver un resultado de esta. Esto con el fin de no tener diferentes instancias de un programa principal haciendo la ejecución menos eficiente a que si se hiciera con uno solo. Se buscó generar una instancia guardada de los métodos implementados con una fase de entrenamiento realizada previamente, para evitar gastos de tiempo y procesamiento al entregar los métodos cada que se necesiten.

5. Pruebas Para llevar a cabo pruebas de los algoritmos seleccionados, primero se realizaron haciendo uso de datos de libre acceso de internet, no sólo para entrenar los métodos en el caso donde es necesario sino también para saber cuál de los algoritmos implementados podría funcionar mejor para cada caso. En lo que respecta a predicción se usaron datos de UCI Machine Learning Repository (Lichman, 2013), del cual se obtuvo el grupo de datos llamado Individual household electric power consumption Data Set. Dicho conjunto contiene de 1, 048,576 lecturas con los que se hicieron varias pruebas en las que se varió la cantidades de datos con la que se trabajó. haciendo pruebas con 8,192 datos, 16,384 datos, 32,768 datos, 65,536 datos, 131,072 datos, 262,144 datos, 524,288 datos y finalmente se usaraon todos los datos con los que se contaba.

En el caso de agrupación de datos, dado que no se encontró una base de datos gratuita con suficientes datos como para realizar el trabajo con ellos, se tomó una base de datos de Worlds Bank (WorldBank, 2016) en la sección de World Development Indicators, la cual se tomó como base para generar 184 conjuntos de datos para simular el consumo de 184 usuarios a lo largo de un día. Puesto que no se contaban con suficientes datos para realizar pruebas con pocos datos en esta área, lo que se varío fue el límite máximo de clusters que podían generar ambos algoritmos, variando entre 10, 100 y 184 clusters, haciendo 10 pruebas con cada variación de cantidad máxima de clusters posibles y con ello se observó el tiempo de ejecución de los algoritmos. En lo que respecta a las pruebas de predicción de consumo y regresión de patrones de la misma, estas se llevaron a cabo haciendo uso de un servidor Dell PowerEdge R720 con procesador Intel Xeon de 2GHz y con

32 GB de memoria RAM, con sistema operativo Ubuntu 16.04.01. Por otro lado, las pruebas para cluster se realizaron en una computadora Dell Inspiron 17R, procesador i7-3537U de 2.00GHz, con 2 cores y 4 procesadores lógicos y 8.00GB de RAM, con sistema operativo Windows 10 en versión educativa. Estas pruebas se realizaron con el fin de poder observar el tiempo de ejecución de los métodos y así compararlos para determinar cuál debería ser probado en el servidor. Nótese que previo a realizar pruebas en el área de clustering, se analizaron dos métodos para determinar el número de clusters a en cada uno, estos se probaron a través de realizar tres pruebas diferentes en las cuales se varió el número máximo que podía usar para buscar el número de clusters. Dicho parámetro se varió entre 10, 100 y 183. Estos datos fueron comparados y de estos se escogió el mejor para el proyecto.

6. Análisis de algoritmos El análisis de algoritmos se hizo de forma teórica a través de estudiar cada uno de los pasos que los diferentes algoritmos realizan, y midiendo el tiempo de ejecución total que deberían tomar al ser implementados tomando como principal métrica la cota superior big O. Es de importancia notar que se analizaron ambos algoritmos para tanto los casos de clustering y predicción; además que dichos resultados fueron respaldados con las primeras pruebas de las implementaciones llevadas a cabo.

7. Método para el servidor de prueba Una vez determinados los algoritmos a ser empleados dentro del servidor de prueba basado en los resultados en las secciones anteriores, se procedió a crear un script en Python que manejara la solicitud y los datos dados y pasarlo a la implementación del método que corresponde. De igual manera se realizó una limpieza de código de los scripts correspondientes a ser subidos así como también se creó un archivo de texto con los módulos externos de Python que son necesarios para poder realizar de forma correcta la ejecución de las implementaciones realizadas. Finalmente se procedió a subir todos estos archivos limpios a un repositorio correspondiente. Ya dentro del servidor se llevó a cabo la clonación del repositorio dentro del sistema, se instalaron los módulos con PIP y finalmente se dejó corriendo el servicio de análisis.

G. Interfaz de usuario

La fase inicial del proyecto consistió en la recolección de datos para la identificación de un problema que se presentará en hogares u empresas guatemaltecas en áreas relacionadas a la energía eléctrica. Para esto se procedió a realizar entrevistas a diversos tipo de personas que tiene relación con el grupo mencionado, entrevistando padres y madres de familia y personas independientes, con diferentes áreas de trabajo que abarcaron desde empresarios, maestros hasta especialistas del área de electricidad tanto en el área técnica como en el área de ingeniería. Dichas entrevistas fueron realizadas basadas en la metodología de Design Thinking, de tal modo que el problema identificado

tras haber analizado las entrevistas fue la falta de control en el consumo de energía.

Al haber aislado el problema se procedió a realizar investigaciones sobre como en otros países han abordado dicha situación, investigando casos como el de Chicago y Abu Dhabi. De igual manera se procedió a identificar qué tipos de herramientas podrían ser utilizadas y qué otras ya existían en el mercado como las ofrecidas por Scheider Electrics. Luego de ello se pasó a examinar que tipo de soluciones se podían implementar dentro de la realidad guatemalteca, llegando a concluir en el desarrollo de un sistema de monitoreo de energía eléctrica para hogares, utilizando un sensor propio y diferentes módulos para satisfacer el sistema computacional que se requiere.

A continuación se detalla la metodología utilizada para el módulo de interfaz de usuario.

1. Investigación Esta fué la primera fase realizada, porque se necesita el sustento de una base teórica para cumplir con los objetivos específicos de la interfaz de usuario, que incluyen las características de usabilidad y utilidad. Se realizó una búsqueda para conocer sobre conceptos alternativos relacionados con un buen diseño de interfaz de usuario que satisfagan sus funciones, y determinar si es suficiente cumplir con las características de una interfaz útil o si existen otros conceptos y áreas del diseño que los abarquen. A menudo se confunden ciertos términos en el área de diseño de interfaces que se deben esclarecer.

La investigación incluye todos los conceptos utilizados en este trabajo y sus características, herramientas de desarrollo, comparativas entre plataformas y paradigmas de diseño llevados a cabo a lo largo de la elaboración de este módulo.

2. Diseño de interfaz La fase de diseó de la intefaz de usuario se vió apoyada por todo el equipo del proyecto. Era necesario hacer una definición del tipo de información a recabar, la frecuencia con que serían medidas y la utilidad que tendrían para la interfaz de usuario y sus funciones. Esta definición fue de carácter fundamental en el cumplimiento del objetivo sobre la definición de funcionalidades del sistema. Desde el pricipio, se trató de especificar explícitamente la forma en que trabajaría el sistema y el medio de comunicación entre módulos, lo cual no fué posible por la falta de madurez del proyecto en ese momento. Por esta razón, se concretaron parámetros básicos con los cuales trabajar, los cuales se fueron refinando hasta llegar a una versión final con la que todos los módulos fueran capaces de trabajar.

La descripción general de funcionalidades e interacciones quedó descrita de la siguiente manera:

- **Recolección de datos:** el tipo de datos a recolectar a través del sensor será de potencia y

la carga a la plataforma será realizada cada 30 segundos, los cuales serán almacenados en un servidor web.

- **Análisis de datos:** los datos analizados a través de diferentes algoritmos, estarán disponibles en un servidor web.
- **Disponibilidad de la información:** la información estará disponible a través de un servicio web conectado al almacenamiento central, y que proporcionará los datos a la interfaz de usuario.

A partir de esta delimitación acordada por todos los miembros del equipo, fue posible definir las funcionalidades propias de la interfaz de usuario, y cumplir así con los objetivos de definir las funcionalidades del sistema e implementar una interfaz de usuario útil.

3. Sobre la utilidad y definición de funcionalidades Ambos objetivos están estréchamente relacionados, porque la definición de funcionalidades describe lo que el usuario podrá hacer dentro del sistema, y la utilidad describe si lo que el usuario puede hacer dentro del sistema es útil.

Se realizó un análisis de las funcionalidades que deberían estar disponibles para los usuarios de la plataforma. Todas estas funciones que serían accedidas a través de la interfaz de usuario debieron derivarse solamente de los datos proporcionados por el servicio web, y estos son básicamente los datos de potencia en crudo y su fecha de medición. A partir de estos datos, se planificó una serie de componentes que son los que proporcionan las funcionalidades de la interfaz y el tipo de usuarios que podrían acceder a estos.

a. Definición de componentes Se llevó a cabo la selección de los componentes que debía tener el sistema. Estos componentes se debían presentar al usuario de una forma ordenada y estructurada. En consecuencia, se propuso la clasificación de componentes según su tipo, y una sección que agrupara los componentes pertencientes al mismo tipo. Este acercamiento permitiría una navegación sencilla para el usuario y un mejor entendimiento de la plataforma, desde el primer uso.

Se identificaron secciones básicas que podían agrupar a los componentes según su tipo. Esto llevó a pensar sobre las funcionalidades básicas que se le debían presentar al usuario, a través de una sección que brindara al usuario con información pertinente sobre su consumo, que fuera funcional y brindara utilidad a la interfaz, un área de análisis estadístico, que brindara gráficos más avanzados sobre su comportamiento en cuanto al consumo energético por día de la semana

u hora del día y una sección sobre otras funcionalidades proporcionadas por otros módulos de la plataforma, como un área de análisis predictivo, para usuarios avanzados que desearan una predicción sobre su consumo el día siguiente.

La descripción de los componentes se construyó a partir de iteraciones, pues fue cambiando a lo largo del diseño y fueron necesarias tres versiones distintas para llegar a una final.

La sección de consumo contempla el consumo que ha tenido el usuario a lo largo del tiempo. Se pensó en incluir tres componentes que mostraran esta funcionalidad. Un componente que mostrara el consumo energético hasta el momento, otro que mostrara el gasto en quetzales hasta el momento y uno último de visualización del consumo mensual hasta la fecha en Kw/h (*kilowatts hora*). El primer y segundo componentes tendrían filtros por día, semana y mes. Mientras que el tercer componente tenfría filtro por hora, día y semana.

La descripción de estos componentes fue iterativa, y a continuación se describen sus tres versiones.

Versión 1

1. Consumo en tiempo real:
    - Consumo en lo que va del mes en cifra.
        - Filtro por día en cifras y gráfica de pastel.
        - Filtro por semana en cifras
    - Consumo promedio por día
    - Opción para manterse debajo de la tarifa social, consumo restante para llegar.
    - Gráfica de linea que indica el consumo
        - Por minuto
        - Por hora
        - Por día
2. Configuración
    - Privacidad de datos, por día, por hora y por semana.
    - Perfil, con opción a cambiar contraseña.
    - Visualización: seleccionar qué ver en pantalla principal.

En esta versión aún no se habían definido todas las secciones que tendría la interfaz. Solamente se tenían definidas las secciones de consumo y de configuración. Las funcionalidades eran básicas

y se definió un componente para mostrar el límite en la tarifa social.

Versión 2

1. Consumo en tiempo real:
   - Consumo en lo que va del mes. Cifra en kWh y en quetzales.
   - Componente con tarifa social.
   - Gráfica de linea que indica el consumo
     - Por minuto
     - Por hora
     - Por día
2. Análisis de consumo
   - Consumo promedio por día
   - Filtro por hora del día en cifras y gráfica de pastel.
   - Filtro por día de la semana en cifras y gráfica de pastel.
   - Filtro por semana en cifras.
3. Análisis predictivo
   - Predicción para las siguientes 24 horas.
4. Configuración
   - Privacidad de datos, por día, por hora y por semana.
   - Perfil, con opción a cambiar contraseña.
   - Visualización: seleccionar qué ver en pantalla principal.

La segunda versión agregó las dos secciones sobre análisis de datos, tanto estadístico, como predictivo. Esta sección ya se parece mucho a la versión final con algunas diferencias. Para el análisis de consumo, se tenía planificado tener las cifras y un gráfico de pastel, que mostrara el consumo promedio en la hora del día y día de la semana, y en la configuración se tenía la opción para configurar la privacidad de los datos y cambiar datos en el usuario, como contraseña.

Versión 3

1. Dashboard
2. Consumo en tiempo real:
   - Consumo en lo que va del mes en kWh. Filtro por mes, semana y día.
   - Consumo en lo que va del mes en quetzales. Filtro por mes, semana y día.

- Gráfica de linea que indica el consumo
  - Por hora
  - Por día
  - Por semana

3. Análisis de consumo
   - Consumo promedio en kWh. Filtro mensual, semanal y diario.
   - Gasto promedio en quetzales. Filtro mensual, semanal y diario.
   - Consumo promedio por hora del día en gráfica de barras.
   - Consumo promedio por día de la semana en gráfica de barras.
4. Análisis predictivo
   - Predicción para las siguientes 24 horas.
5. Configuración
   - Perfil con datos generales del usuario.
   - Visualización: seleccionar qué ver en pantalla principal.

Finalmente, en la versión 3 se decidió cambiar los gráficos de pastel por gráficos de barras, porque el usuario los entendía mejor y era más fácil su visualización. Se definió que los usarios no serían editables y que vendrían preconfigurados, por lo tanto no era necesaria una sección de configuración de usuario. Se definió que los componentes para consumo actual y consumo promedio eran lo suficientemente importantes como para compartir su espacio con la funcionalidad de consumo en quetzales. Por esta razón, se hizo una separación de componentes.

La visualización del consumo en tiempo real funciona para el usuario básico que posee el dispositivo instalado en su hogar y desea saber el comportamiento en su consumo hasta la fecha.

b. Identificación de Roles Después de definir las funcionalidades que tendría el sistema, se necesitaba conocer el tipo de usuarios que iban a interactuar con este. Un usuario común podía ser capaz de visualizar su propia información de consumo, y además existiría un usuario capaz de visualizar los datos de todos los usuarios de determinada región, después de llevar a cabo una despersonalización de los datos. Este usuario podría analizar el consumo de todos los usuarios a la vez, utilizando la misma interfaz que un usuario común, porque los tipos de datos siguen siendo los mismos. Además de estos usuarios del sistema, debe existir un usuario administrador, que no cuenta con una interfaz en específico, pero que tiene la potestad de crear, modificar y eliminar usuarios del sistema, o dicho de otra forma, tiene los permisos para hacerlo. Finalmente, se decidieron los siguientes roles, que deberían estar presentes en el sistema:

- **Administrador**: Es el administrador de la plataforma y es quien administra los usuarios. Posee los permisos de visualización del analista.
- **Analista**: Obtiene una visualización general y análisis de datos de todos los usuarios básicos.
- **Básico**: Obtiene una visualización individual, datos sobre su consumo en tiempo real y análisis de datos de su consumo. Este usuario posee el dispositivo de consumo instalado en su hogar.

4. Sobre la usabilidad Existe una variedad de tipos de interfaces de usuario, paradigmas de diseño, teorías del color, estudios de usabilidad y sobre los cuales fué necesario seleccionar uno o varios que satisfacieran los requerimientos del proyecto, furan asequibles y maximizaran su efectividad. De esta selección dependería el posterior diseño visual y plataforma de desarrollo.

a. Metodología de diseño Para cumplir con el objetivo de diseñar una interfaz de usuario **usable**, se seleccionó una metodología de diseño consistente a lo largo del ciclo de vida de la interfaz. Al analizar y comparar las opciones presentes en la investigación previa, la metodología de diseño seleccionada fue la de **Estudio de usabilidad**, propuesta por *Nielsen Norman Group*. Esta metodología fue considerada la más asequible con el tiempo y recursos disponibles de las investigadas, y satisface los objetivos de este módulo del proyecto. Esta metología aprovecha los estudios realizados por entidades reconocidas y evita un gasto adicional en investigación propia. Esta metodología se puede ajustar a las necesidades propias del proyecto (interfaz de usuario para un sistema de información sobre el consumo energético), y e incluye satisfactoriamente una interacción directa con el usuario final, para constatar su efectividad.

La metología seleccionada está sustentada sobre una base teórica y fué aplicada utilizando estos pasos, descritos en el marco teórico:

1. Probar los diseños de los competidores para obtener información barata sobre el rango de interfaces alternativas que tienen funcionalidades similares al sistema propio.
2. Conducir un estudio de campo para ver cómo se comportan los usuarios en su hábitat natural.
3. Hacer prototipos en papel de una o más ideas de diseño y probarlas. Mientras menos tiempo se invierta formulando estas ideas mejor, porque se necesitarán cambiar basadas en los resultados de las pruebas.

4. Refinar las ideas de diseño para probarlas mejor en múltiples iteraciones, empezando por prototipos de baja fidelidad hasta llegar a representaciones altamente fieles ejecutadas desde una computadora. Probar cada iteración.

5. Inspeccionar el diseño a partir de *lineamientos establecidos de usabilidad*, basados en estudios previos o investigaciones.

6. Cuando se ha tomado una decisión y se implementa el diseño final, realizar una última prueba. Problemas sutiles de usabilidad ocurren durante la implementación.

Esta metodología se aplicó a lo largo de todo el diseño y desarrollo de la interfaz de usuario, con el fin de satifacer este objetivo del módulo.

Interfaces alternativas Como primer paso en el diseño de una interfaz usable, se procedió a analizar los tipos de interfaces disponibles en el mercado. Varias plataformas para el monitoreo de enegía en tiempo real se encuentran disponibles para usuarios en otros países. Sin embargo, no es posible su evaluación directa debido a que se necesita un equipo instalado en una residencia, cosa que queda excluida en este trabajo.

A pesar de este inconveniente, fue posible analizar superficialmente el contenido de estas interfaces. En el anexo D se encuentran las capturas de las interfaces analizadas. En total, se analizaron las interfaces de los productos de dos empresas, que se encuentran funcionando en varios países con un sistema muy similar al del proyecto actual.

La primera empresa es Schneider-electric, cuya interfaz se muestra en la Figura 124 y 125 del anexo D, que fabrica dispositivos básicos y avanzados para el monitoreo de energía en tiempo real. La interfaz que se analiza es la de su producto PowerLogic PM8000, que cuenta con su propia pantalla y en consecuencia, una interfaz integrada (embedida). Esto significa que la interfaz es propia del dispositivo y por lo mismo, se encuentra limitada en tamaño, resolución de pantalla y el tipo de información que puede mostrar, tomando en cuenta la capacidad de procesamiento del dispositivo. La Figura 124 muestra una interfaz con información básica sobre el voltaje, amperaje y la potencia en ese momento. Muestra la fecha y la hora y cuatro botones con las funciones de regresar, subir, bajar y otro de función especial. La otra, Figura 125, muestra una gráfica con una numeración en decenas en el eje x y un porcentaje en el eje y. No se logra identificar el tipo de información que quiere mostrar.

La segunda empresa es Efergy, una empresa que trabaja en Estados Unidos y Sudáfrica, con

un sistema de monitoreo de energía eléctrica. El producto y servicio es muy parecido al proyecto que se desea desarrollar y por lo tanto, se pueden extraer buenas conclusiones sobre el tipo de interfaz que debe desarrollarse. Las imágenes que muestran esta interfaz se encuentran en el anexo D, figuras 126 y 127. Al analizar estas interfaces, se observa una plataforma web, con un tablero, y una serie de componentes que muestran la información de consumo energético al usuario. En la barra superior, se observa un panel de navegación hacia las diferentes funcionalidades de la plataforma y el usuario que ingresó actualmente. El diseño y selección de colores es agradable a la vista y no parece difícil de utilizar. Sin embargo, no se detecta un menú de ayuda o preguntas sobre la información que allí se muestra, aunque la interfaz pueda parecer sencilla de entender. La Figura 128 muestra una interfaz embedida y por lo tanto más sencilla, pero que muestra la información más importante que el usuario necesita saber inmediátamente.

Estos ejemplos son de gran ayuda en la definición de los primeros prototipos de la plataforma, pues son interfaces probadas y funcionando en la vida real.

b. Observación de usuarios Siguiendo la metodología, el siguiente paso requiere un trabajo de campo para investigar al usuario en su estado natural. Este estudio ya se llevó a cabo en las fase de descubrimiento del proyecto. Los hallazgos fueron que el usuario espera su factura proporcionada por la EEGSA (Empresa Eléctrica de Guatemala S.A.), y procede a realizar el pago sin objeción alguna. Para el usuario no es posible obtener una descripción detallada sobre su consumo y por lo tanto, no hay forma justificada para realizar un reclamo. Este es el resumen del trabajo ejecutado previamente.

c. Bosquejos en papel Tomando en cuenta la información valiosa proporcionada por las interfaces alternativas y del trabajo de campo ya realizado, lo siguiente fué la preparación de bosquejos que definieran las primeras pantallas con las debería contar el sistema. Las funcionalidades básicas de la interfaz se definieron también de forma iterativa y por esta razón, la visualización de la interfaz fue cambiando tanto visual, como estructuralmente.

Los primeros bosquejos del sistema servirían para pensar en cómo hacer valiosa la información sobre su consumo. ¿Cómo se pueden empaquetar las funcionalidades definidas en la otra sección, de manera que cumplan con las características de una interfaz usable? Se realizaron diseños de poca fidelidad, que fueron cambiando rápidamente a medida que se recibían comentarios sobre los usuarios y expertos en el área. Estas, fueron entrevistas informales que únicamente buscaban indagar sobre generalidades para realizar un diseño eficiente y fácil de usar.

Con el tiempo, se fueron perfeccionando estas entrevistas hasta llegar a un cuestionario definido de la siguiente manera:

1. ¿Qué es lo que más le interesa saber sobre su consumo de energía eléctrica?
2. ¿Entiende todos los componentes de la interfaz? ¿Qué tipo de información cree que muestran?
3. ¿Tuvo problemas para identificar el menú de opciones, el usuario activo, el contenido, etc.?
4. Si se pierde en la interfaz o necesita ayuda, ¿sabe dónde encontrarla?
5. ¿Logró identificar todos los botones y opciones en el menú y barra superior?

Surgieron comentarios interesantes sobre cómo debería de ser la pantalla principal de la plataforma, que tipo de visualización facilitaría el entendimiento del usuario, etc. Un dato interesante es que la mayoría de usuarios tenían un interés mayor en saber su consumo eléctrico actual en el mes y el dinero que llevaban gastado, que en el resto de componentes. Para ellos, contar con estos dos datos serían razón suficiente para que el sistema fuera útil para ellos. Este tipo de comentarios, servía de influencia para el refinamiento de los prototipos.

El anexo H, Figura 146 muestra el documento en cuestión, utilizado para recabar los comentarios de los usuarios. En el proceso y utilizando este cuestionario, se fueron refinando estos bosquejos hasta llegar a pantallas que mostraban una interacción completa. El anexo E muestra estos bosquejos, junto con las funcionalidades que se tenían definidas para entonces. Para fines demostrativos, los bosquejos incluyen distintos tipos de barras de opciones y de navegación, para poder discutirlos también. Ahora se realiza un pequeño análisis sobre su contenido.

La Figura 129 muestra la pantalla del tablero principal. Esta contaría con una barra lateral y dentro un menú de opciones, que son accesos directos hacia las categorías de funciones. Una barra superior con el nombre de la plataforma y el usuario activo. Se muestra una gráfica de líneas y datos sobre el consumo de energía que se llevaba hasta el momento utilizando cuadros informativos, en forma de componentes.

La Figura 130 muestra la pantalla de información de consumo. La barra superior muestra ahora el logo de la plataforma, el usuario activo, una opción para selección de idioma y una opción para ver la ayuda. En el contenido, se muestra el consumo promedio que ha tenido el usuario y la cantidad en promedio de dinero que ha gastado. Después se muestra un gráfico de barras con un detalle sobre el comportamiento de su consumo. Al lado derecho, está el panel de ayuda, con

información sobre los componentes y preguntas comúnes. Se nota el avance en cantidad de opciones a comparación de la pantalla analizada anteriormente.

La siguiente pantalla, en la Figura 131, muestra la misma barra lateral, con la opción de análisis seleccionada. En el contenido están tres gráficas de línea, donde se muestra el consumo promedio por día de la semana y por hora del día. Hay un filtro para el primer gráfico. En la barra superior hay una sección del usuario activo y un símbolo de búsqueda.

La Figura 132 simula una pantalla de configuración, con opciones para configurar el tablero a través de *switches*. Esta es una de las pocas configuraciones disponibles en la plataforma, pues no requiere de una gran personalización por parte del usuario. En esta interfaz, la configuración tiene su propia sección en la navegación.

En la sección de predicción, Figura 133 la interfaz muestra un cuadro con la predicción de consumo para el siguiente día, y el segundo componente es un gráfico de pastel, que muestra los grupos (clusters), con tipo de consumo similar, encontrados con la ayuda de inteligencia artificial. Este último, solo le interesa a una persona realizando evaluaciones sobre el consumo de energía en ciertas regiones del país o ciudad. Por lo tanto, no representa un valor especial para el usuario común.

El siguiente bosquejo, que aparece en la Figura 134, muestra un despliegue de pantalla diferente a los anteriores, pues contiene una barra de navegación horizontal. Esta modificación fué la sugerencia de un experto en el área, cuyo argumento explicaba que la escasa cantidad de opciones para navegar, hacía innecesario el uso de una barra vertical, que ocupaba espacio de más en la interfaz. Este fué el bosquejo realizado para visualizar una interfaz de este tipo. Aquí se puede ver también el aumento en espacio para componentes en el área de contenido.

El último bosquejo que aparece es uno de ingreso a la aplicación, en la Figura 135, que no contiene algo especial. Sólamente el ingreso de credenciales y un botón de ingreso.

d. Selección de colores La selección de colores para la plataforma en general e identificación de marca se realizó con la guía de *Material Design* y de expertos en el área de experiencia de usuario. Entre la gama de colores primarios y de acento disponibles, se seleccionó como color primario el azul-grisaseo (*blue-grey*) y de color secundario el ámbar (*amber*). Esto abrió paso a la creación de un logo que identifique a la plataforma y producto en general. El logo se encuentra representado en la Figura 15. Posteriormente, se utilizaron estos colores en la interfaz de usuario,

con la premisa de ser validados por los usuarios del sistema.

Figura 15: Logo principal. EnerSave

e. Prototipos digitales Después de realizar los diseños de poca fidelidad, llegó la hora de hacer prototipos nativos, visualmente atractivos y que simularan interfaces en la vida real. Los prototipos se realizaron utilizando la interfaz web y estuvieron sujetos a cambios a través de comentarios de usarios mientras no fuera posible evaluarlos con un estudio de usabilidad. Este estudio se postergó hasta tener una versión completa, que pudiera evaluarse.

El diseño incluyó los nuevos colores, surgieron cambios interesantes a lo largo del diseño, y también los componentes cambiaron su forma de presentación. El anexo F muestra los diseños más relevantes recolectados durante esta etapa. En este momento, la interfaz era de carácter púramente demostrativo, pues no presentaba funcionalidad alguna. A continuación, se discuten estas interfaces y sus características.

5. Estudios de usabilidad Cuando ya se tenía una versión de la interfaz completada, se procedió a realizar los estudios de usabilidad. Esto consiste en un estudio de campo que selecciona a ciertos usuarios comunes para realizar tareas específicas dentro de la aplicación. Se realizaron dos estudios en total, y se obtuvo retroalimentación por parte de los usuarios en cada caso. La forma en que se evaluó el rendimiento de los usuarios a través de la interfaz, fue con la ayuda de un cronómetro y una observación sobre comportamientos peculiares por parte de los usuarios mientras realizaban el estudio.

Las condiciones para realizar el estudio eran que el usuario no podía hacer preguntas al entrevistador sobre dónde se encontraban los componentes que estaba buscando, o la información que requería la tarea. Sin embargo, si podían realizar preguntas si no entendían el propósito de la tarea. Para realizar el estudio, el usuario además, debía aceptar un consentimiento informado y llenar una hoja de datos generales, que no solicitaba ningún dato personal. Estos documentos se encuentran en el anexo H, en las figuras 147 y 150, respectivamente.

El procedimiento para realizar el estudio se hizo siguiendo estos pasos:

1. Aislar al usuario en un salón tranquilo, sin distracciones.
2. Solicitar al usuario que leyera el consentimiento informado y si acepta realizar el estudio, que lo firme.
3. Solicitar que rellene la hoja de datos generales y que tome asiento en el lugar donde se realizaría la prueba.
4. Presentarle una hoja con las tareas a realizar y e indicarle que el entrevistador estaría observando mientras el usuario las realice.
5. El usuario realiza las tareas, mientras el entrevistador utiliza el cronómetro para medir el tiempo en que realiza las tareas y también toma notas breves sobre comportamientos del usuario.
6. Se le da al usuario un cuestionario post entrevista, para que indique su experiencia utilizando la plataforma y si quiere incluir alguna sugerencia o recomendación.
7. Agradecer al usuario por su tiempo y terminar el estudio.

Los documentos que aquí se indican están en adjuntos en el anexo H, y a continuación se describen las tareas que realizó el usuario durante la entrevista.

1. Ingresar a la aplicación con las siguientes credenciales:
    a) Usuario: **user123@gmail.com**
    b) Contraseña: **usuario123**
2. Encontrar su consumo en kWh en el mes.
3. Encontrar su consumo en quetzales para el día de hoy.
4. Entrar a la configuración de usuario.
5. Encontrar el consumo promedio de los miércoles.
6. Ver el consumo en tiempo real de esta semana.
7. Encontrar la predicción de consumo para el día de mañana.
8. Encontrar el consumo promedio a las 16:00 horas.

9. Salir de la plataforma (cerrar sesión).

Estas tareas se pueden encontrar en el anexo H, Figura 148. Los estudios de usabilidad se realizaron con los mismos usuarios en ambas ocasiones. Esta fue la manera en que se llevaron a cabo los estudios de usabilidad y sus resultados se encuentran más adelante.

6. Selección de plataforma de desarrollo Se realizó una selección del tipo de interfaz que se desarrollaría, cumpliendo con ambas características de usabilidad y utilidad. Se eligió el tipo de interfaz web, que presenta funcionalidades necesarias para el proyecto, gracias a sus características. A continuación se describen.

- Visualización desde cualquier tipo de dispositivo.
- Único orígen de verdad, por lo tanto, no es necesario actualizar todas las aplicaciones instaladas cada vez que surja una nueva actualización.
- Permite recolectar la información de todos los usuarios para mostrarlos en una misma aplicación, algo que no es posible para aplicaciones integradas en dispositivos o sensores, como se comparó con las interfaces alternativas.

Ahora que ya se ha seleccionado el tipo de interfaz a desarrollar, el siguiente paso consistió en seleccionar una plataforma de desarrollo de interfaces web (en este caso, aplicación web). La variedad de plataformas existentes, sus componentes e interconexiones permiten cierta flexibilidad al momento de elegir una de ellas, porque los componentes son, en su mayoría, de libre acceso. Sin embargo, vale la pena compararlas en base a su efectividad, rendimiento, curva de aprendizaje, comunidad de desarrollo y requerimientos específicos del proyecto. Esta comparación ya se realizó en el marco teórico y se llegó a una decisión basado en los siguientes requerimientos:

- Comunidad sólida, para obtener un excelente soporte en la resolución de problemas.
- Poseer una estructura modular, con el fin de organizar las funcionalidades del proyecto. De esta forma pueden agregarse y eliminarse funcionalidades sin comprometer el resto de la aplicación.
- Estándares de la web, que utilice los estándares definidos por la W3C (*World Wide Web Consortium*).
- Flexibilidad, que permita modificar fácilmente la lógica de la interfaz, y la lógica de la aplicación agregando funciones innovadoras.

- Eficiencia y efectividad, que utilice procedimientos para aumentar la rapidez de carga de la aplicación y formas de evitar recargas innecesarias.
- Facilidad de uso, que pueda ser fácilmente comprensible por cualquier desarrollador de interfaces de usuario.

La plataforma seleccionada para el desarrollo de la interfaz de usuario es **AngularJS** en su versión **1.5.0**, que cumple satisfactoriamente con las características antes mencionadas.

Adems, para la mejora en el proceso de desarrollo se adquirió una licencia para la plantilla **Fuse** disponible para esta plataforma, de la cual se obtuvieron ciertos estilos utilizados en la plataforma, con características de Material Design.

Para la sección de gráficos se está utilizando Angular-nvD3, una biblioteca gratuita para gráficos, con versión en AngularJS.

7. Desarrollo e implementación Por tratarse de un diseño iterativo, el desarrollo de la interfaz se dio de la mano con el diseño y la interacción con los usuarios. Como ya se mencionó anteriormente, el tipo de interfaz desarollado es web en la plataforma AngularJS y utilizando componentes de la plantilla Fuse. El desarrollo se llevó a cabo en un IDE (*Integrated Development Environment*), que es una herramienta para un desarrollo de software, más sencillo que un editor de texto.

La aplicación guarda un diseño modular, lo que permite agregar y remover componentes para las pruebas de usabilidad. Además, la plataforma de desarrollo, permite una mayor eficiencia si se trabaja de forma estructurada, favoreciendo a la parte de usabilidad. En el diseño estructural de la plataforma, se definió que cada componente tendría su propia plantilla, su propio archivo de estilos, archivo de controlador y servicio. Todo esto facilitaría su modificación y lo aislaría del resto de la aplicación.

VI. Resultados

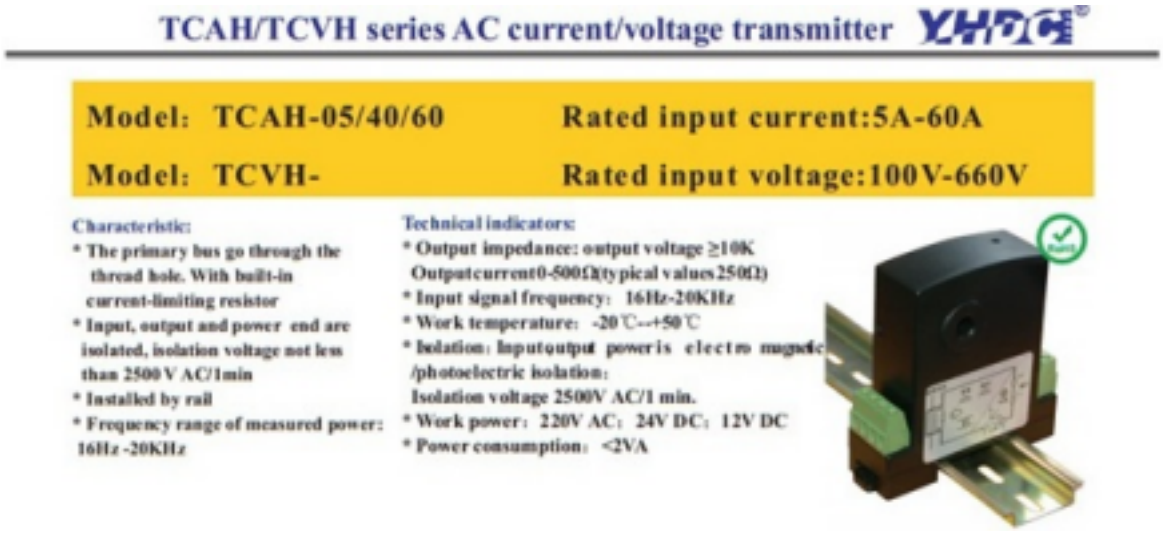
Tras haber finalizado el proceso de investigación de cada uno de los módulos, estos llegaron a resultados específicos de su área, tanto la parte de Sensores y Protolocos de Comunicación, Seguridad de la Información, Almacenamiento y Servicios Web, Integración, Análisis de Datos e Interfaz de Usuario. Dichos resultados se presentan a continuación.

A. Sensores y protocolos

1. Selección de Sensor de Voltaje. Para encontrar el sensor de voltaje más apropiado se establecieron las siguientes características con mayor prioridad: facilidad de conexión, valor de salida entre 0 y 5 V, valor mínimo de medición de 100 Vrms y frecuencia de medición de 60 Hz.

Tomando en cuenta estas propiedades se escogió el sensor de voltaje YHDC serie TCVH:

Figura 16: Sensor de voltaje utilizado para la implementación del módulo.

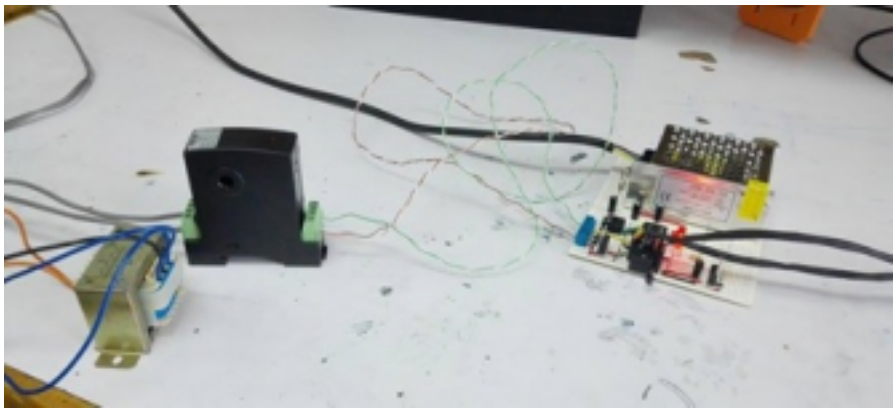


(Zhang, J.)

Cuadro 23: Mediciones utilizadas para caracterizar y calcular el error del sensor de voltaje.

Valor Real (Vrms)	Salida Sensor (V)	Valor Medido (V)	Error (%)
120	0.99	0.99020	0.0198
208	1.72	1.71875	0.0727
240	2	1.99510	0.2451
480	3.99	3.99020	0.0049

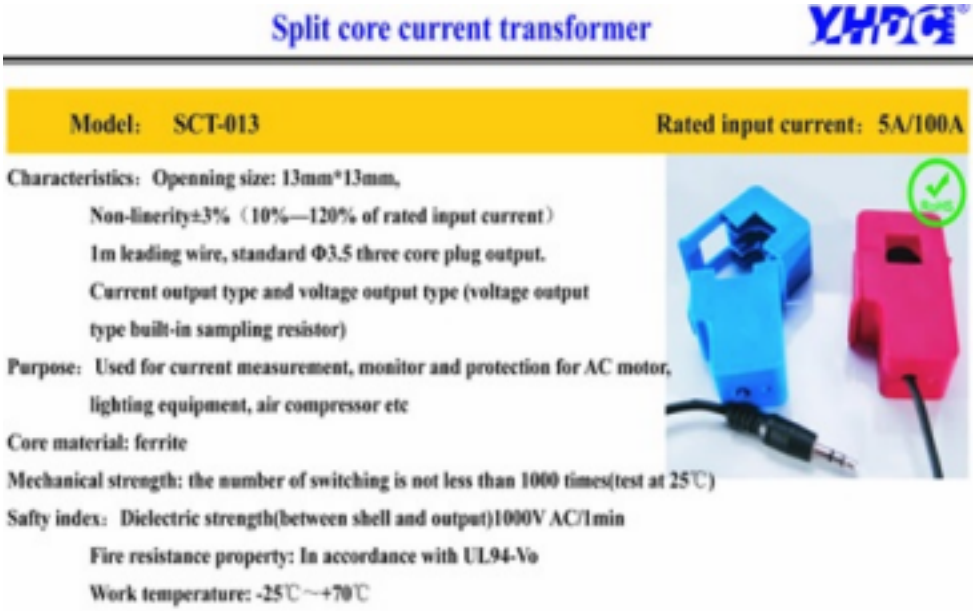
Figura 17: Mediciones tomadas para los distintos valores de voltaje mostrados en la tabla anterior.



[Elaboración propia]

2. Selección de Sensor de Corriente. Para seleccionar el sensor de corriente se establecieron los siguientes criterios principales: que fuese un sensor no invasivo, rango de salida de 0 a 5V, rango de medición apropiado para la corriente de consumo en un hogar y de fácil instalación. Siguiendo estos criterios se seleccionó el sensor YHDC SCT 013-000.

Figura 18: Sensor de corriente utilizado para la implementación del módulo.



(YHDC, 2013)

Figura 19: Especificaciones de entrada y salida del sensor de corriente.

Table of technical parameter:

Model	SCT-013-000
Input current	0-100A
Output type	0-50mA

(YHDC, 2013)

Debido a que el sensor no detecta cambios con corrientes menores a 5A no fue posible realizar mediciones de referencia con distintos valores, sin embargo, se logró medir la corriente de un calentador de agua, cuyo consumo fue de 6.5A. Con esto fue posible obtener una referencia y determinar cuánto error se obtenía al aplicar un modelo lineal al sensor.

Cuadro 24: Mediciones realizadas para calcular el error del sensor de corriente.

Valor real (A)	**Valor medido (A)**	**Error (%)**
10.2	10.1560	0.4312
13.4	13.2197	1.3455
20.46	20.4960	0.1758
23.85	23.9426	0.3883
30.28	30.0700	0.6936

3. Protocolo de Comunicacón. Para la realización del proyecto fue necesario utilizar dos protocolos de comunicación estándar. Para la comunicación inalámbrica se utilizó el módulo de radiofrecuencia nRF24L01+, cuya configuración y manejo requirió utilizar el protocolo serial SPI. No fue necesario realizar ninguna configuración con este protocolo puesto que la librería lib_rf2gh4_10.h utilizada para implementar el módulo RF se encarga de toda la configuración y envío de datos. Este protocolo se utilizó tanto para el manejo del módulo emisor como para el módulo receptor.

Otro protocolo de comunicación fue necesario para transferir la información desde el microcontrolador hacia la Raspberry Pi. Para esta comunicación se utilizó el protocolo UART, con la misma configuración en ambos dispositivos, dicha configuración se muestra a continuación.

Cuadro 25: Parámetros de configuración para el protocolo UART.

Velocidad de transmisin (baudios)	**Bits a transmitir**	**Paridad**
9600	8	Ninguna

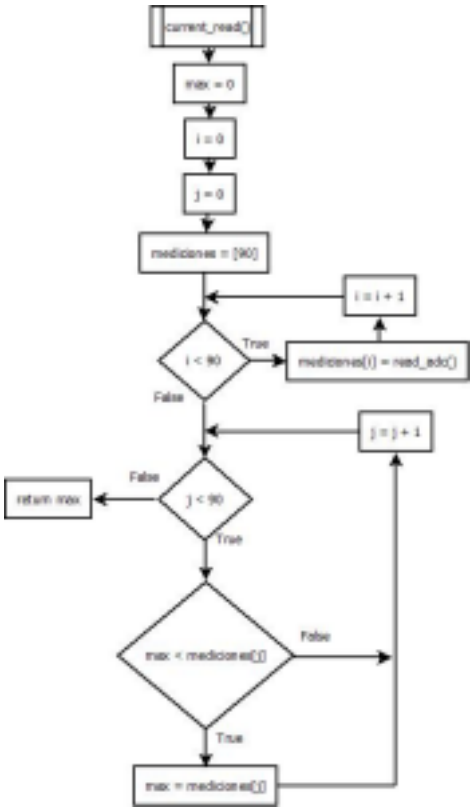
4. Implementación del Módulo.

Figura 20: Diagrama de implementación del módulo.

[Elaboración propia]

a. Lectura de Sensores. Debido a que el sensor de corriente devuelve un voltaje AC fue necesario implementar una función llamada current_read() para obtener la amplitud de la señal. El funcionamiento de la función se muestra en el diagrama de flujo siguiente.

Figura 21: Diagrama de flujo del funcionamiento de la función current_read.

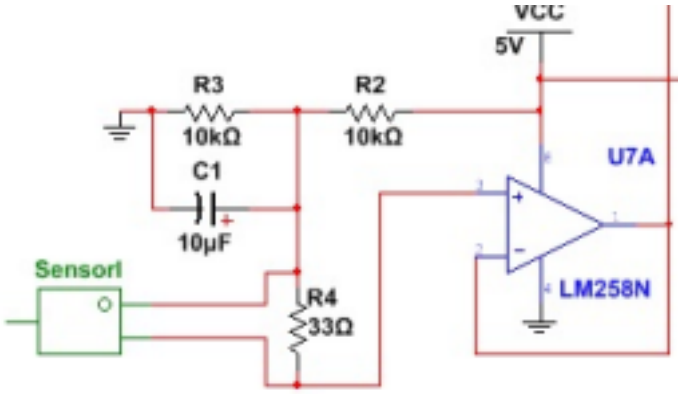


[Elaboración propia]

Debido a que el microcontrolador no puede recibir señales negativas fue necesario acondicionar

la señal. Esto se realizó con el circuito siguiente. (Daniel, 2015)

Figura 22: Circuito acondicionador de la señal del sensor de corriente.



[Elaboración propia]

A continuación se muestran mediciones realizadas con la función current_read().

Cuadro 26: Mediciones de voltajes AC utilizando la función current_read.

Voltaje Digital	143	173	190	218	157	166	191
Voltaje Medido (V)	2.793	3.379	3.711	4.258	3.066	3.242	3.730
Voltaje Real (V)	2.9	3.4	3.9	4.4	3.14	3.3	3.86
Diferencia	0.107	0.021	0.189	0.142	0.074	0.058	0.130

b. Comunicación inalámbrica . Para implementar el módulo RF emisor con el microcontrolador se utilizó la librería lib_rh2gh4_10.h creada por la empresa Bizintek Innova, S.L.

Figura 23: Cambios realizados a la librería lib_rf2gh4_10.h, versión original a la izquierda y versión modificada a la derecha.

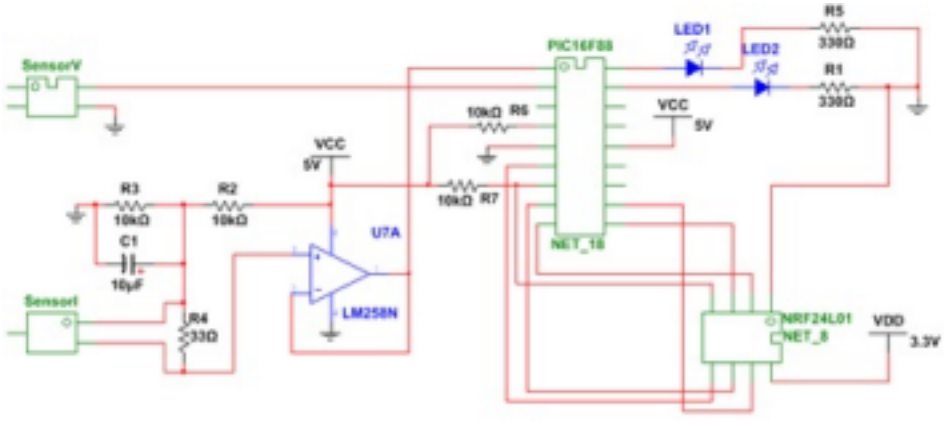
```
// PORTB
#define RF_IRQ      PIN_B0
#define RF_CS       PIN_B5

// PORTC
#define   RF_CE     PIN_C2
#define   SCK       PIN_C3
#define   SDI       PIN_C4
#define   SDO       PIN_C5

// PORTB
#define   RF_IRQ_TRIS   TRISB,0
#define   RF_CS_TRIS    TRISB,7

// PORTC
#define   RF_CE_TRIS    TRISC,2
#define   SCK_TRIS      TRISC,3
#define   SDI_TRIS      TRISC,4
#define   SDO_TRIS      TRISC,5
```

```
28  // PORTB
29  #define RF_IRQ      PIN_B0
30  #define RF_CS       PIN_B5
31  #define RF_CE       PIN_B3
32  #define SCK         PIN_B4
33  #define SDI         PIN_B1
34  #define SDO         PIN_B2
35
36  // PORTB
37  #define RF_IRQ_TRIS   TRISB,0
38  #define RF_CS_TRIS    TRISB,5
39  #define RF_CE_TRIS    TRISB,3
40  #define SCK_TRIS      TRISB,4
41  #define SDI_TRIS      TRISB,1
42  #define SDO_TRIS      TRISB,2
43
44
45
46
```

[Elaboración propia]

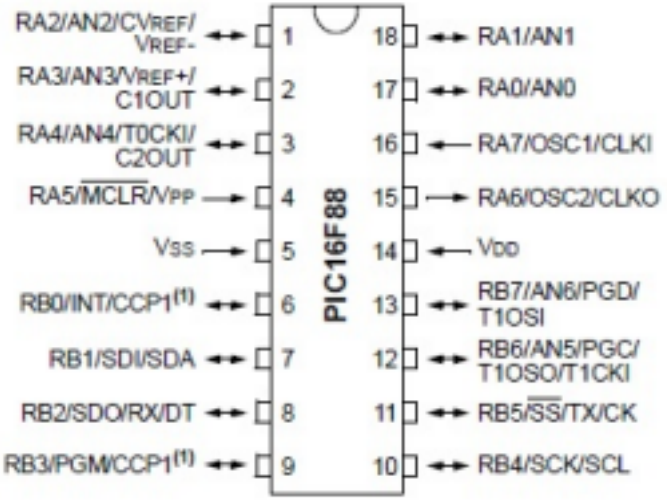
Se incorporó el código creado en la sección de digitalización con las funciones proveídas por la librería para implementar el módulo RF transmisor. El circuito para implementar la digitalización y transmisión por radiofrecuencia se muestra en la figura siguiente.

Figura 24: Circuito utilizado para implementar la digitalización y transmisión RF.



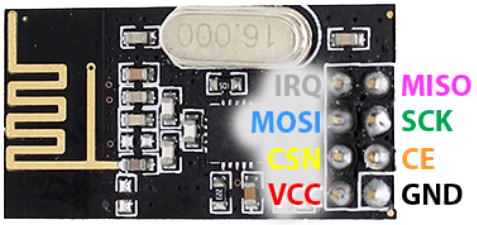
[Elaboración propia]

Figura 25: Distribución de pines del microcontrolador.



(Microchip, 2013)

Figura 26: Distribución de entradas del módulo de radiofrecuencia.



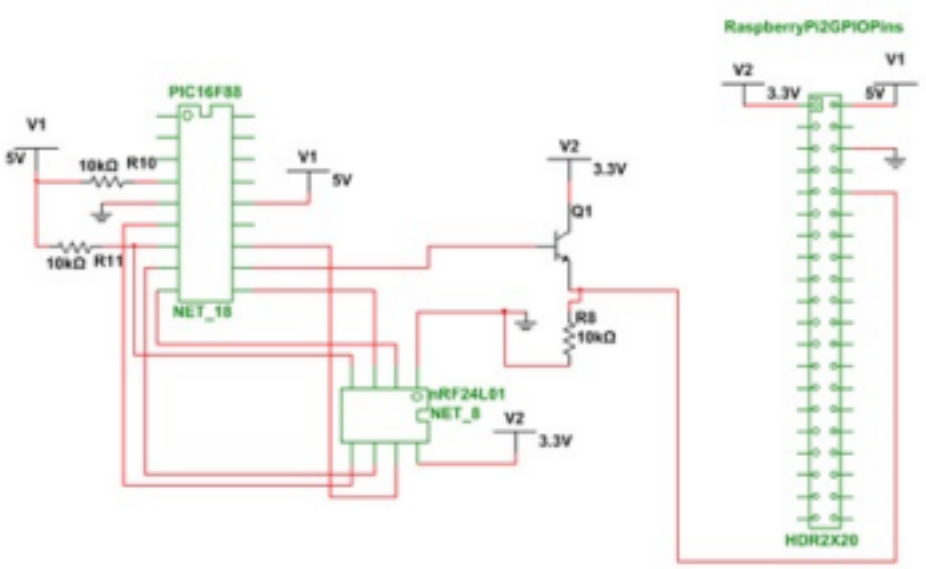
(Galaz, 2015)

El ciclo de funcionamiento del circuito es que se toma una muestra de la señal de cada sensor, se realiza la digitalización, se envía por medio de radiofrecuencia y se espera 1 segundo antes de empezar de nuevo. Los LEDs en el circuito indican si hay un error con la comunrcación inalámbrica. Si se enciende el LED1 es porque hay un problema con el hardware y si se enciende el LED2 es porque el módulo receptor no confirmó la recepción del mensaje.

Para configurar el módulo receptor se utilizó otro microcontnolador PIC16F88 con la librería lib_rf2gh4_10.h.

El diagrama de conexión se muestra en el esquemático siguiente.

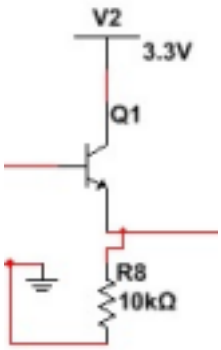
Figura 27: Diagrama de conexión entre el módulo RF, el microcontrolador y la Raspberry Pi 2.



[Elaboración propia]

c. Procesamiento de la información y almacenamiento en el servidor . Para recibir información desde el microcontrolador a la Raspberry se le configuró el puerto UART a 9600 baudios y se conectó el pin TX del microcontrolador al pin RX de la Raspberry, pero el microcontrolador funciona con 5V y la Raspberry con 3.3V entonces fue necesario hacer una conversión lógica de nivel. Para eso se utilizó un transistor y una resistencia como se muestra a continuación.

Figura 28: Circuito utilizado para convertir 5V a 3.3V en el canal de UART.



[Elaboración propia]

El microcontrolador envía 2 datos cada segundo hacia la Raspberry, entonces se revisa que cada vez que el buffer de datos de UART sea igual a 2 se guarde la información en variables. La recepción consiste en los dos valores obtenidos del ADC de los sensores entonces es necesario

realizar la conversión a valores de voltaje y corriente rms. Para realizar la conversión se utilizan las siguientes fórmulas.

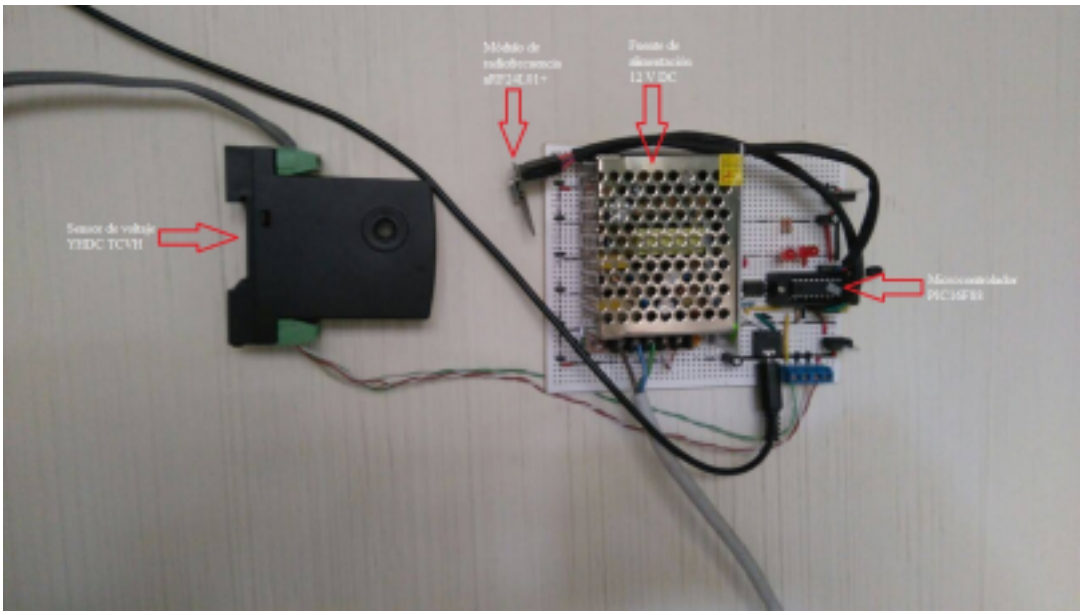
$$Vrms = \frac{10000}{83} * (voltajedigitalizado * \frac{5}{255} + 0{,}1) + \frac{82}{83} \quad [V] \qquad \text{(VI..1)}$$

$$Irms = \frac{625}{16} * (corrientedigitalizada * \frac{5}{255} + 0{,}1) + \frac{149}{256} \quad [A] \qquad \text{(VI..2)}$$

Figura 29: Conexión de los sensores a la red eléctrica para realizar la toma de mediciones.

[Elaboración propia]

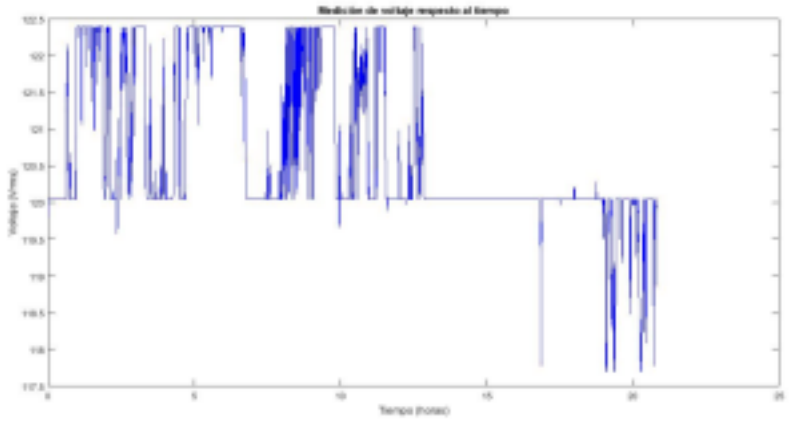
Figura 30: Circuito digitalizador y transmisor.



[Elaboración propia]

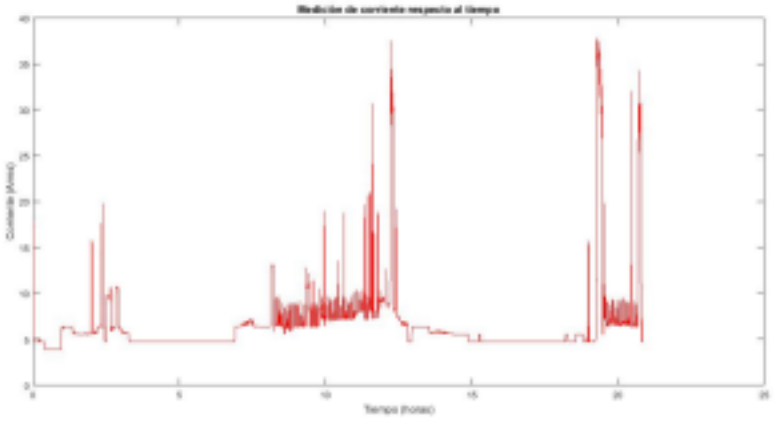
A continuación se muestran las gráficas de las primeras mediciones de voltaje y corriente y cálculos de potencia en un hogar. Se realizaron mediciones a lo largo de aproximadamente 21 horas y la primer medición fue alrededor de las 10 a.m.

Figura 31: Medición de voltaje a lo largo de 21 horas.



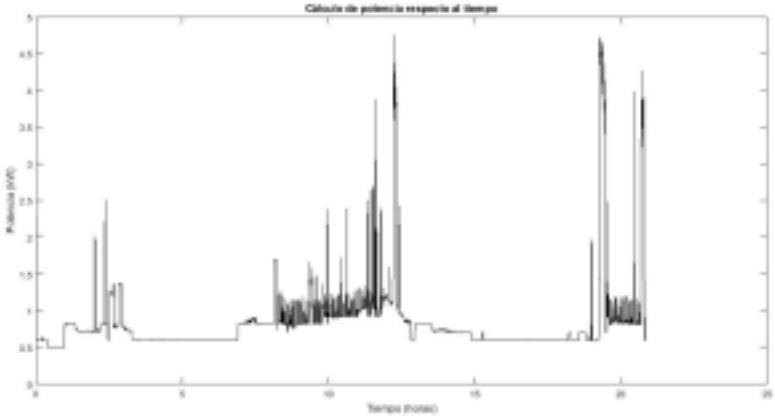
[Elaboración propia]

Figura 32: Medición de corriente a lo largo de 21 horas.



[Elaboración propia]

Figura 33: Cálculo de potencia utilizando los valores de voltaje y corriente rms.



[Elaboración propia]

B. Seguridad de la información

1. Implementación de algoritmos simétricos: Tiempo Se analizó la cantidad de tiempo que cada uno de los algoritmos criptográficos simétricos ejecuta para las tres funciones; cifrado, descifrado y generación de llave criptográfica. Este análisis de tiempo se realizó como una diferencia de tiempo de inicio contra tiempo final para cada función. Los datos de entrada son bytes desde 1 hasta 100000 en múltiplos de 10.

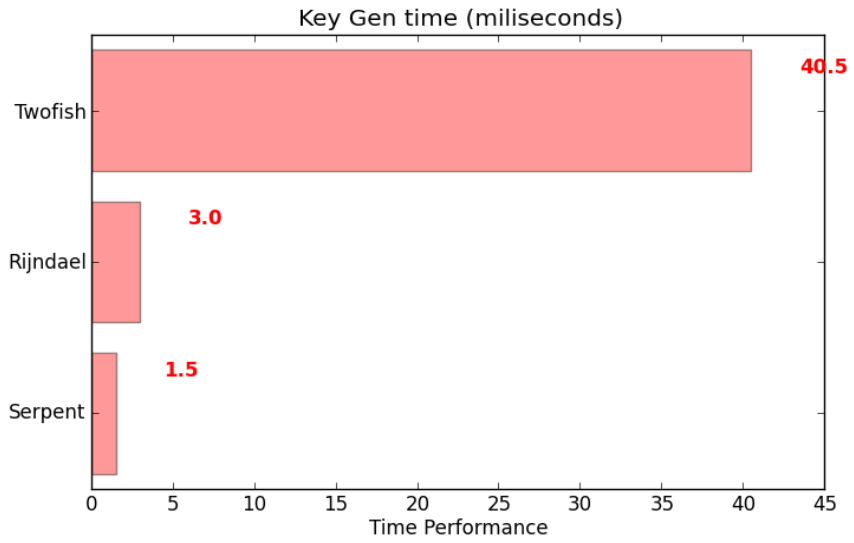
a. Generación de llave :

Cuadro 27: Comparativa de tiempo de generación de llave en milisegundos de algoritmos simétricos

Cantidad de bytes	**Twofish**	**Rijndael**	**Serpent**
1 byte	39.0	3.0	4.0
10 bytes	39.0	3.0	1.0
100 bytes	39.0	3.0	1.0
1000 bytes	40.0	3.0	1.0
10000 bytes	38.0	3.0	1.0
100000 bytes	48.0	3.0	1.0

Con base en los datos de tiempo de generación de llave criptográfica del Cuadro 27, para los algoritmos simétricos; se pudo determinar un promedio, el cuál se obtuvo como resultado final: Twofish con 40.5ms, Rijndael con 3.0ms y Serpent con 1.5ms, cómo se muestra en la Figura 34.

Figura 34: Comparativa de tiempo de generación de llave en milisegundos de algoritmos simétricos.



b. Cifrado :

Cuadro 28: Comparativa de tiempo de cifrado en milisegundos de algoritmos simétricos.

Cantidad de bytes	**Twofish**	**Rijndael**	**Serpent**
1 byte	0.0	0.0	0.0
10 bytes	0.0	0.0	0.0
100 bytes	6.0	7.0	2183.0
1000 bytes	64.0	67.0	22619.0
10000 bytes	638.0	650.0	224815.0
100000 bytes	6420.0	6514.0	2245036.0

Con base en los datos de tiempo de cifrado del Cuadro 28, para los algoritmos simétricos; se pudo determinar un promedio, el cuál se obtuvo como resultado final: Twofish con 1188.0ms, Rijndael con 1206.3ms y Serpent con 415775.5ms, como se muestra en la Figura 35.

Figura 35: Comparativa de tiempo de cifrado en milisegundos de los tres algoritmos simétricos.

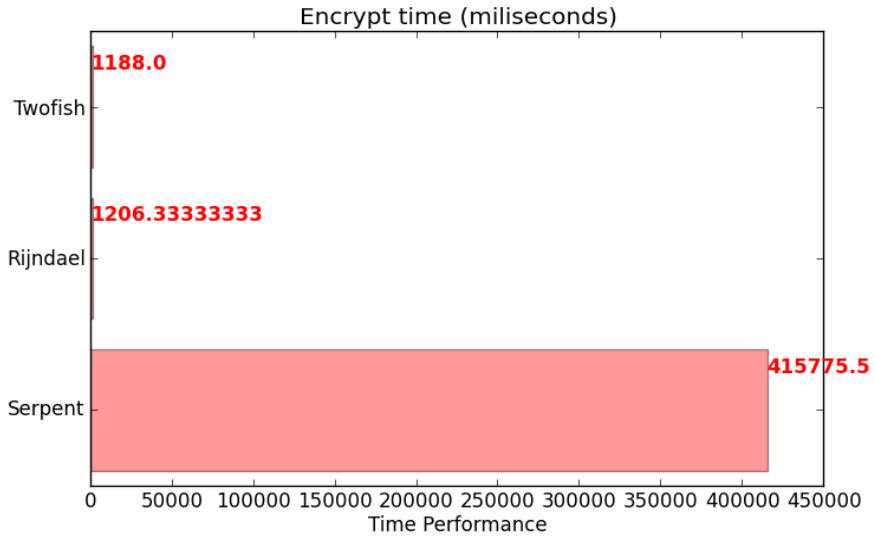
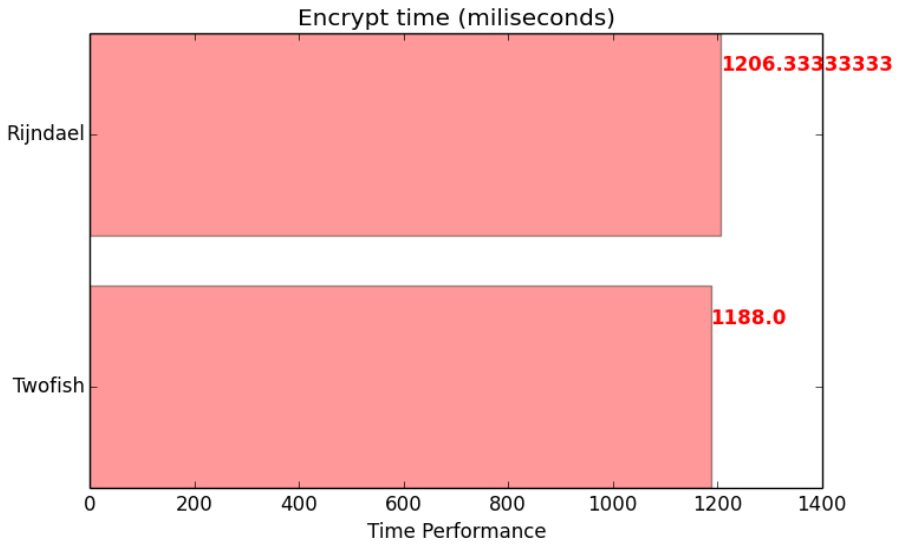


Figura 36: Comparativa de tiempo de cifrado en milisegundos de Twofish y Rijndael.



c. Descifrado :

Cuadro 29: Comparativa de tiempo de descifrado en milisegundos de algoritmos simétricos

Cantidad de bytes	**Twofish**	**Rijndael**	**Serpent**
1 byte	0.0	1.0	0.0
10 bytes	0.0	0.0	0.0
100 bytes	6.0	7.0	2044.0
1000 bytes	65.0	67.0	21026.0
10000 bytes	637.0	650.0	209082.0
100000 bytes	6388.0	6510.0	2085111.0

Con base en los datos de tiempo de descifrado del Cuadro 29, para los algoritmos simétricos; se pudo determinar un promedio, el cuál se obtuvo como resultado final: Twofish con 1182.66ms, Rijndael con 1205.83ms y Serpent con 386210.50ms, cómo se muestra en la Figura 37.

Figura 37: Comparativa de tiempo de descifrado en milisegundos de los tres algoritmos simétricos.

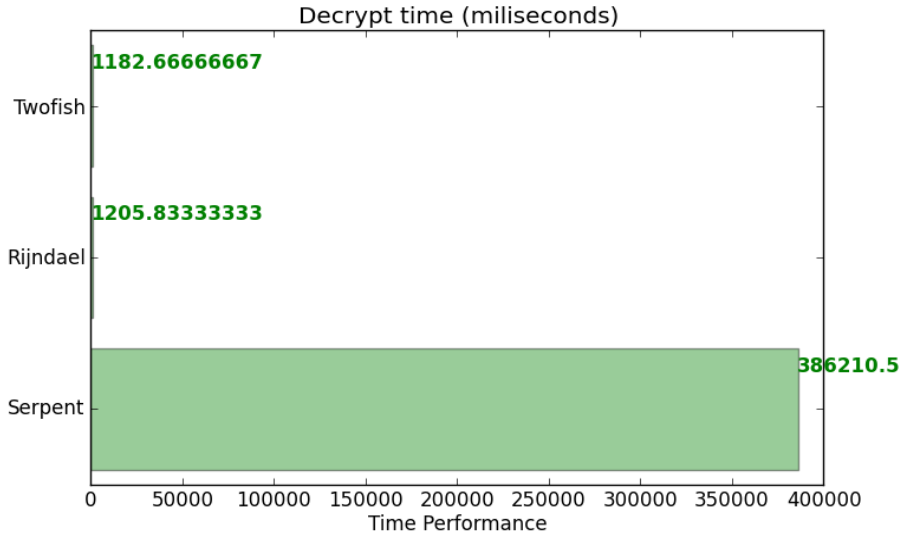
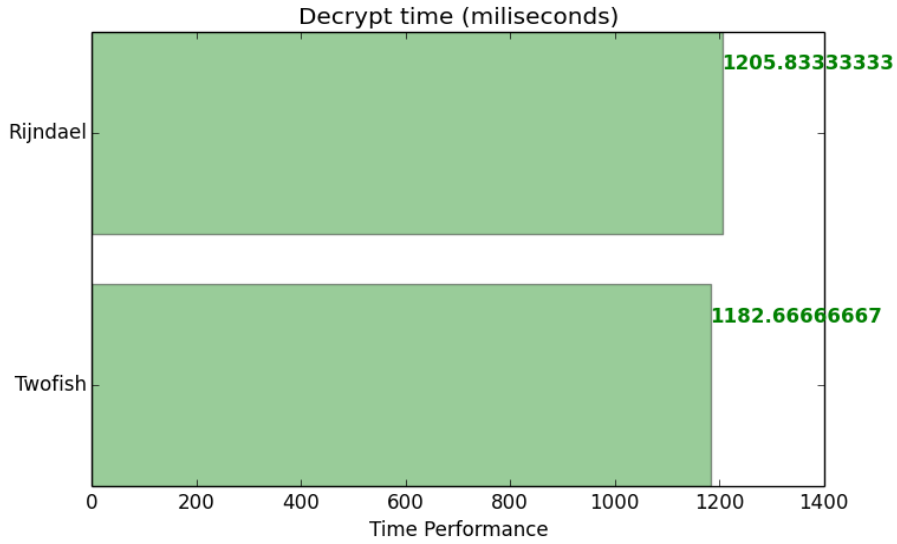


Figura 38: Comparativa de tiempo de descifrado en milisegundos de Twofish y Rijndael



2. Implementación de algoritmos simétricos: Memoria Se analizó la cantidad de memoria que consume cada uno de los algoritmos criptográficos simétricos para cada una de sus funciones; cifrado, descifrado y generación de llave. El análisis se realizó utilizando una herramienta llamada *memory_profiler* para cada función. Los datos de entrada son bytes desde 1 hasta 100000 en múltiplos de 10.

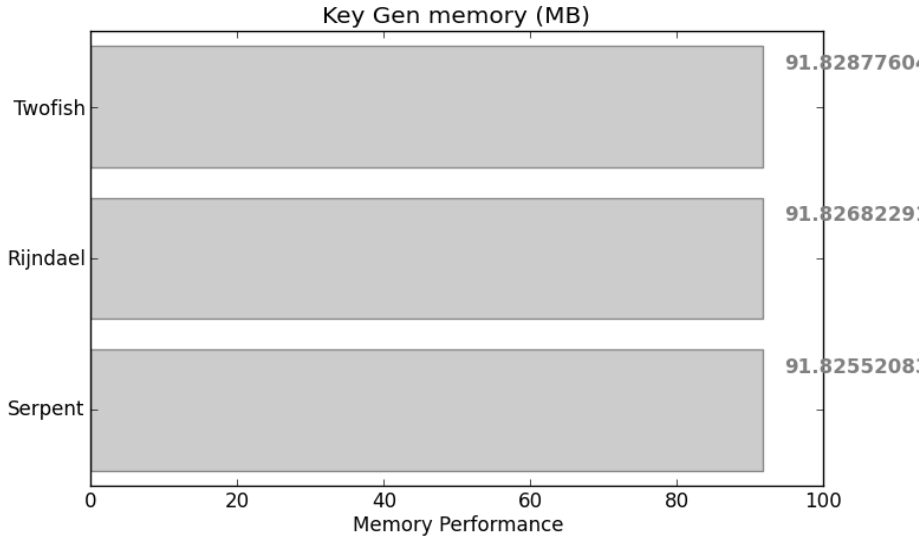
a. Generación de llave :

Cuadro 30: Comparativa de memoria descifrado en megabytes de algoritmos asimétricos

Cantidad de bytes	**Twofish**	**Rijndael**	**Serpent**
1 byte	56.53906	56.54687	56.55859
10 bytes	76.66406	76.66406	76.66406
100 bytes	89.51171	89.51171	89.51171
1000 bytes	100.6015	100.6015	100.6015
10000 bytes	108.4804	108.4804	108.4804
100000 bytes	119.1562	119.1562	119.1562

Con base en los datos de memoria de descifrado del Cuadro 30 para los algoritmos simétricos; se pudo determinar un promedio, el cuál se obtuvo como resultado final: Twofish con 91.8287mb, Rijndael con 91.8268mb y Serpent con 91.8255mb, como se muestra en la Figura 39.

Figura 39: Comparativa de memoria de generación de llave en megabytes de algoritmos simétricos.



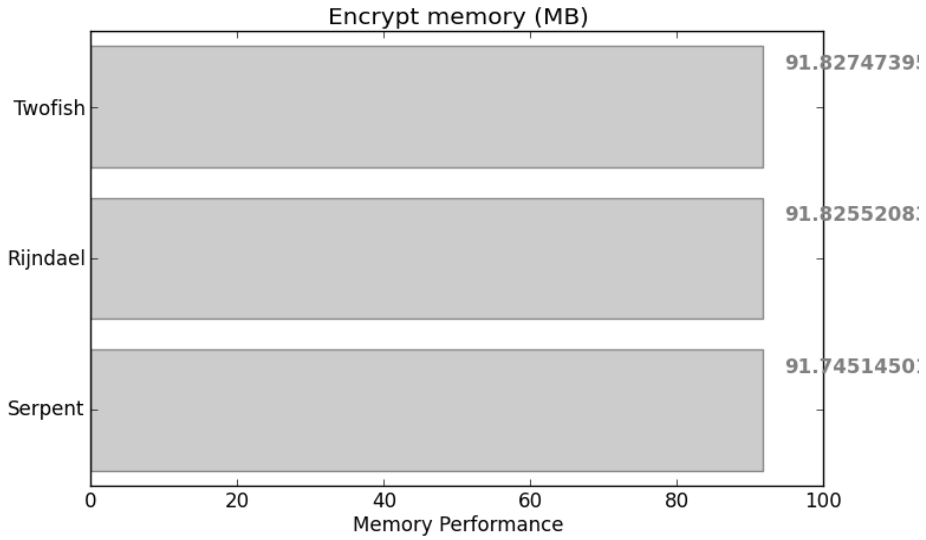
b. Cifrado :

Cuadro 31: Comparativa de memoria cifrado en megabytes de algoritmos asimétricos

Cantidad de bytes	**Twofish**	**Rijndael**	**Serpent**
1 byte	56.53906	56.53906	56.55078
10 bytes	76.66080	76.66406	76.66406
100 bytes	89.50927	89.51171	89.51171
1000 bytes	100.6015	100.6015	100.6015
10000 bytes	108.4804	108.4804	108.4804
100000 bytes	118.6796	119.1562	119.1562

Con base en los datos de memoria de cifrado del Cuadro 31, para los algoritmos simétricos; se pudo determinar un promedio, el cuál se obtuvo como resultado final: Twofish con 91.8274mb, Rijndael con 91.8255mb y Serpent con 91.7451mb, como se muestra en la Figura 40.

Figura 40: Comparativa de memoria cifrado en megabytes de algoritmos simétricos.



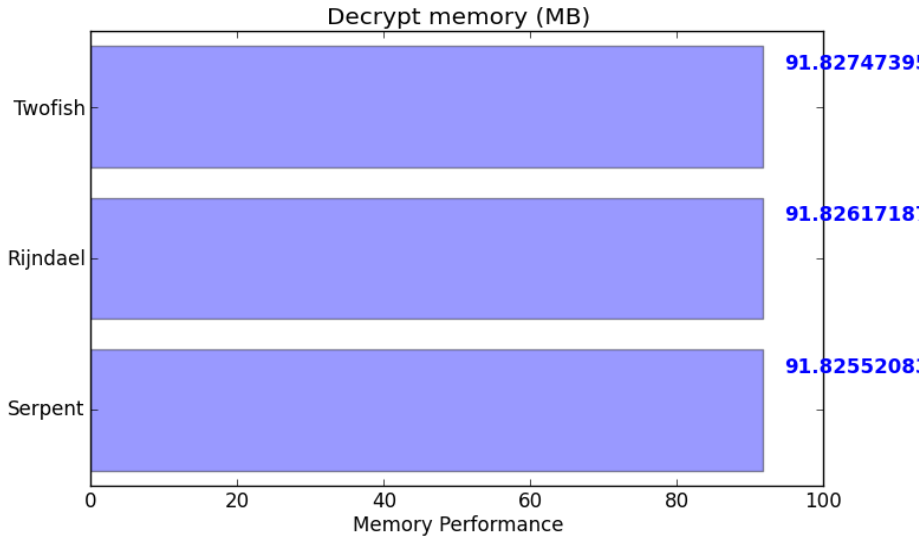
c. Descifrado :

Cuadro 32: Comparativa de memoria descifrado en megabytes de algoritmos asimétricos

Cantidad de bytes	**Twofish**	**Rijndael**	**Serpent**
1 byte	56.53906	56.54296	56.55078
10 bytes	76.66406	76.66406	76.66406
100 bytes	89.51171	89.51171	89.51171
1000 bytes	100.6015	100.6015	100.6015
10000 bytes	108.4804	108.4804	108.4804
100000 bytes	119.1562	119.1562	119.1562

Con base en los datos de memoria de descifrado del Cuadro 32 para los algoritmos simétricos; se pudo determinar un promedio, el cuál se obtuvo como resultado final: Twofish con 91.8274mb, Rijndael con 91.8261mb y Serpent con 91.8255mb, como se muestra en la Figura 41.

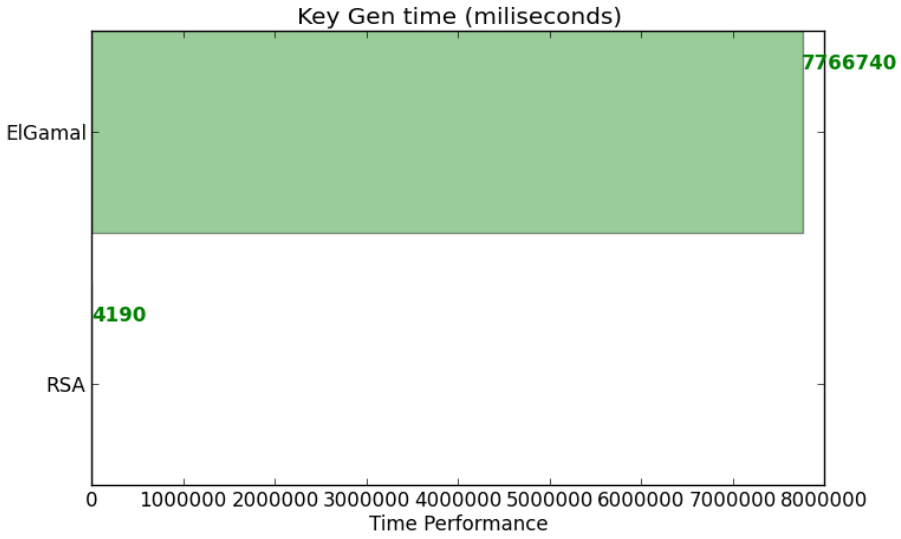
Figura 41: Comparativa de memoria de descifrado en megabytes de algoritmos simétricos.



3. Implementación de algoritmos asimétricos: Tiempo Se analizó la cantidad de tiempo que cada uno de los algoritmos criptográficos asimétricos ejecuta para las tres funciones; cifrado, descifrado y generación de llave criptográfica. Este análisis de tiempo se realizó como una diferencia de tiempo de inicio contra tiempo final para cada función. El dato de entrada, es la llave criptográfica simétrica.

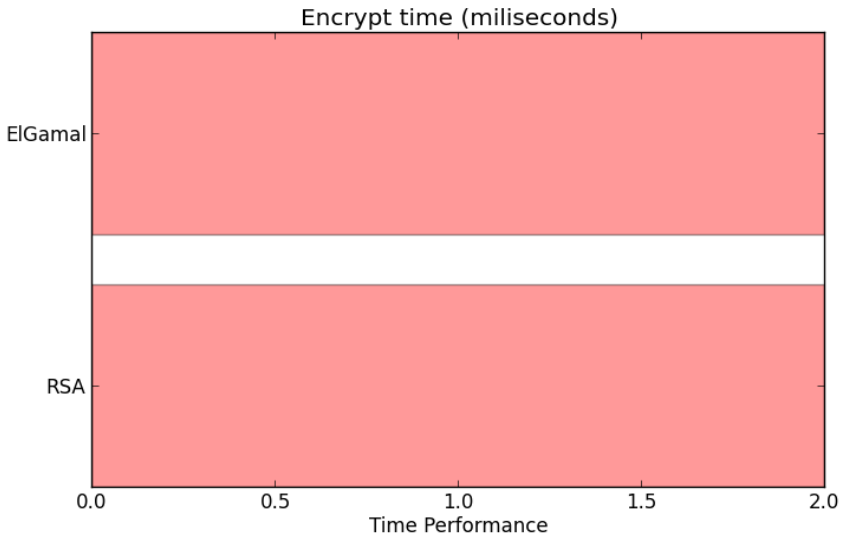
a. Generación de llave criptográfica :

Figura 42: Comparativa de tiempo para generar llaves criptográficas de algoritmos asimétricos.



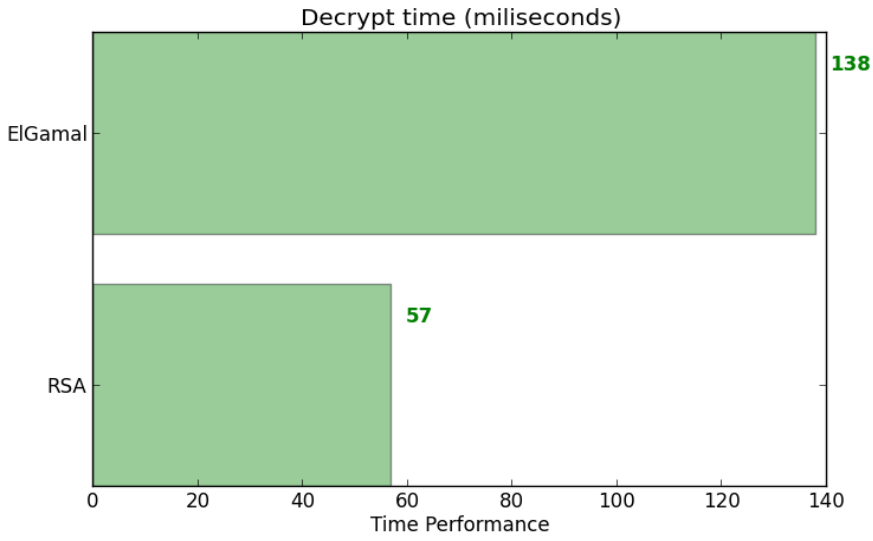
b. Cifrado :

Figura 43: Comparativa de tiempo para cifrar en milisegundos de algoritmos asimétricos.



c. Descifrado :

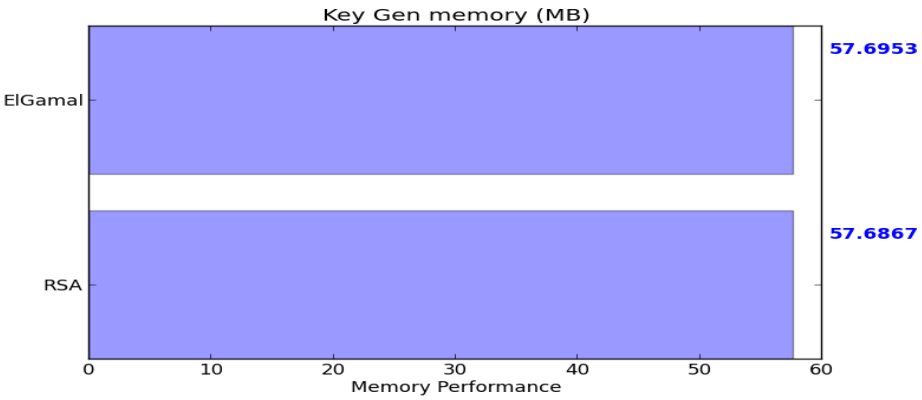
Figura 44: Comparativa de tiempo para descifrar en milisegundos de algoritmos asimétricos.



4. Implementación de algoritmos asimétricos: Memoria Se analizó la cantidad de memoria que cada uno de los algoritmos criptográficos asimétricos ejecuta para las tres funciones; cifrado, descifrado y generación de llave criptográfica. Este análisis de memoria se realizó utilizando una librería llamada *memory_profiler* para cada función. El dato de entrada, es la llave criptográfica simétrica.

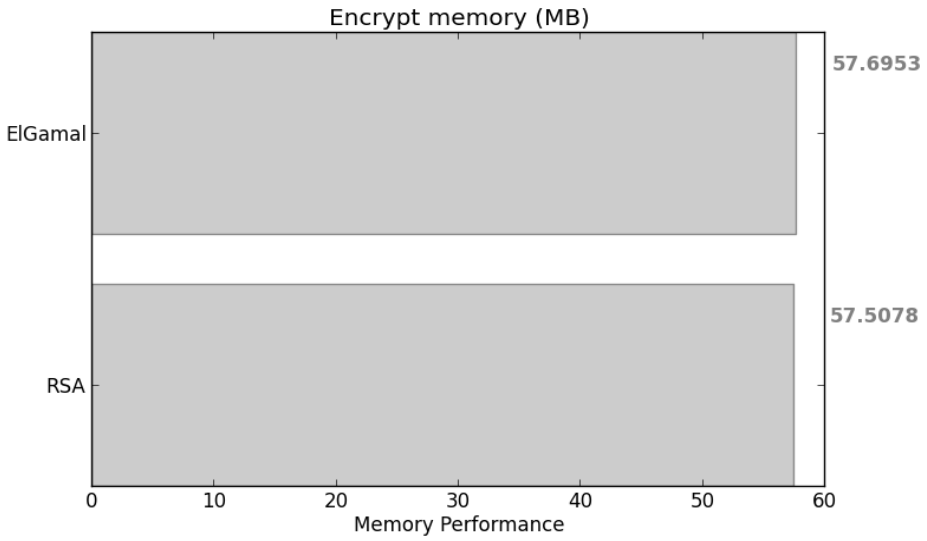
a. Generación de llave criptográfica :

Figura 45: Comparativa de memoria para generar llaves criptográficas en megabytes de algoritmos asimétricos.



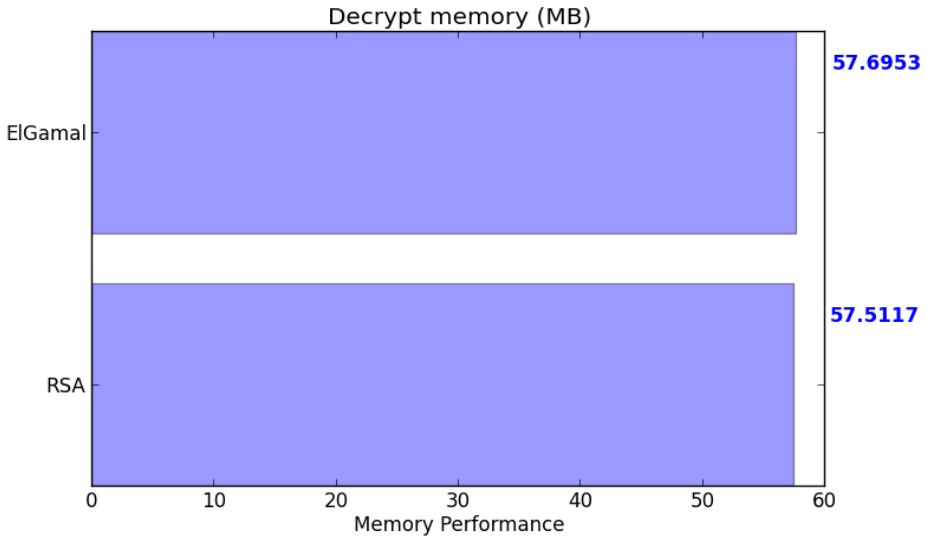
b. Cifrado :

Figura 46: Comparativa de memoria para cifrar en megabytes de algoritmos asimétricos.



c. Descifrado :

Figura 47: Comparativa de memoria para descifrar en megabytes de algoritmos asimétricos.



5. Selección de algoritmo

a. Algoritmos simétricos Se seleccionó como algoritmo simétrico a **AES Rijndael** con 256 bits de tamaño de llave, 128 bits de tamaño de bloque y modo de operación CTR

(Counter); para el cifrado de los datos pertenecientes al proyecto.

b. Algoritmos asimétricos Se seleccionó como algoritmo asimétrico a **RSA** con 4096 bits de tamaño de llave; para el intercambio de la llave criptográfica simétrica.

6. Pruebas de confidencialidad Luego de haber realizado la implementación dentro de la Raspberry utilizando como algoritmo de cifrado simétrico AES Rijndael y como algoritmo de cifrado asimétrico RSA. Se pudo realizar pruebas de confidencialidad utilizando las herramientas descritas en la sección de herramientas de ataque dentro del marco teórico.

Se realizó un análisis de paquetes de red utilizando la herramienta Wireshark, para determinar si la información se envía y se recibe de manera cifrada e ilegible.

Se utilizó la herramienta FindMyHash para determinar si algun valor hash utilizado al momento de la conexión pueda ser un riesgo para el proyecto.

a. Wireshark: Intercambio de llave para el intercambio de llave, se obtuvo.

Figura 48: Wireshark: Intercambio de llaves: Solicitud

```
GET /api/sensors/register/1 HTTP/1.1
Host: ec2-54-70-229-71.us-west-2.compute.amazonaws.com:3000
Content-Length: 953
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Accept: application/json
User-Agent: python-requests/2.3.0 CPython/2.7.6

publicKey=-----BEGIN+PUBLIC+KEY-----
%0AMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAqFN%2BxvfWwES6zDq7nW42%
0ANAb01D9GZCkhUeFbltrBB7%2FHa%2FVLVlUvUnWFOF1Hydoew%2B9wJk6oZh%2BXhvup
llJZ%0AN5ISyMXjU4nK0UOhFX3%2BHboSh%2Bs1jZ6JHkJ3v4QaCS3Mu9htNQ4RLIhamMA
lFYLu%0AqpUPVDsS2xOAinAVv1DRY%2Fb5GVIxRWg1mWd5YRIOnFFFxwRCyNJ020uLvkjr
ME83%0APXDyF%2Bfn9FbBOUlrOcLO9WlGsu%2BFXObHRtfkIDESvRu0fbhvO%2FTuRyJmX
PAM85bu%0ASjILLcRY0T4T16JLV3t2%2FjnLm5bFr6%2F3pguKZXvmguRLOzURqAjA6e3g
4OcIUhnv%0AKXdDigkQ8rUGJRXq8Omx47XQfNUqwnd6oK%2BXNfBRgkja3zYLKvz%2BNFj
m4sPYN8x1%0AreseGQRvFp0d%2B63XzoC9F4yjexZjIF5lF1tuUSpszKYRcnaNLyziBDib
lTGF7INS%0AT7XQJyl8op%2BA1yblbp6yopRuWCWHhLyXk4L1dLyQ2hZYcOiD0YpdOaVAz
PA8vRwM%0ArbIRHYgZyIuFbTXkcIoi0W2RmwSaL2XAkjp%2Fhru75i0F8J9Cdw9NwhH1Ti
B0FrYH%0A22XTloersN7LNXSJhccJ5WAZ4K18a61U9izMVwPJyR2aVUY%2Bfph7d8aYIqi
MltDb%0ARtshhkE%2F%2BC5GKOVZo9%2B3YDcCAwEAAQ%3D%3D%0A-----
END+PUBLIC+KEY-----
&hash=1df7fe3c4260c843fce1322a6e5241e5c30998d6f78af640888f7183f09365bf
f09365bf
```

Figura 49: Wireshark: Intercambio de llaves: Respuesta


```
HTTP/1.1 200 OK
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Download-Options: noopen
X-Content-Type-Options: nosniff
Content-Type: application/json; charset=utf-8
Content-Length: 791
ETag: W/"317-HfOf0+LPYiT7MLQPraqngQ"
Date: Sat, 22 Oct 2016 23:26:31 GMT
Connection: keep-alive

{"data":{"key":"H0GP9Y7zqd9enQcpEzomz1d37SZA/O5D1drjac6ovCZ7+DfNal+LrW
brODqEpZPf4vcXBMk+0JAdUakMMLoMLuXfxINbMO72gCsDMe2UmeFf+UnUbZlP1dgdvNtV
6I3pMTSDLQOL9pZc+gTSlzBLzGkeTG3SPvavMfnm8UmamG7bNInuRK9TjZ07YJt5pmwb/x
iqDPwJoVssh/jm7kGwgZ92bBvy8fwCMNA2YqVTX7QjmHhyrj5WevDDetvZ1Ptil1oa+b8p
3E0qvHPPkPhFk7vzqxl3rwrGde0iNqIJx8d5M1Dtwn/RkIvMqF3WuwFKnDofmWm+eXVzAh
gq+LndwGzCALDP/8fk6SIfznxydYvB4JEfLEXRRFzd7ty5kMnz9DI6nl7w9tfJbUe6l+Ph
JXM80ksFm1hhK/CFRAh1gKxgNQm+CJDUE+IRydDaquztLtWngR7YRcwYS3oYtqXOz8BE8t
yyT4DeVDnIo26DuT/P7gfRLootxXwpfYuSIm/nv1pDoAhourCVMlXTS+gwkz1QVVr5k2Vg
nfvosie7rnZZmnuKSiiOZ6Fl3a8kDkBfiDu5ndY6ZjPaYPZRQN0oCU3O1f5dYgztYqdIKM
Aaw6oPxsvdBDxYu2LkVcDchaSTfUsBWbE+NXU40fIEvqFAguchBrZB30JcQIv1XVKGiB4=
","hash":"4176b52f283a3a0ae387a43d4b99a9320095152b494f91217300a71934f6
4d41","error":false}}HTTP/1.1 200 OK
```


b. Wireshark: Envío de información cifrada : para el envío de información, se muestra dos de las solicitudes y respuestas.

Figura 50: Wireshark: Envío de información #1: Solicitud


```
GET /api/sensors/pushMeasurement/1 HTTP/1.1
Host: ec2-54-70-229-71.us-west-2.compute.amazonaws.com:3000
Content-Length: 398
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Accept: application/json
User-Agent: python-requests/2.3.0 CPython/2.7.6

cypheredContent=AmLZiE4%2FRZ7QgiEJopOCepNUs%2FntLCAA1lpWTPERP5kPR2U%2B
DMhE0DBdjvPYUoRoIGy3%2FkB%2B%2BJVRfOY%2BmiUXKVECzasOiORfCMzh3k8KwmnzaQ
azGa%2F8YHn%2BzViy9leaNXDDvU1we%2B1ceUGBpZ4YYq%2Bjwv%2FPl%2B4xSLrEASoO
0IGbyt9V9QfrkxFVXmn9rfvo6M7w%2FIsceFkf02Bkj2kkdmiwQ0Xbeo7PfRP5U0fL39QO
6TUOPNYk4vYQRMFWWEwIGp3jwTiCIkeFikNQO5vyYQ%3D%3D&hash=6662331b7676e03f
d42213c46bc2227719e16e4843412be4c879fe08433a265fGET
/api/sensors/pushMeasurement/1 HTTP/1.1
```

Figura 51: Wireshark: Envío de información #1: Respuesta

```
HTTP/1.1 200 OK
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Download-Options: noopen
X-Content-Type-Options: nosniff
Content-Type: application/json; charset=utf-8
Content-Length: 36
ETag: W/"24-htQGVXW3GkCm9OaF//Ew+w"
Date: Sat, 22 Oct 2016 23:26:41 GMT
Connection: keep-alive

{"data":"All 4 measurements pushed"}
```

Figura 52: Wireshark: Envío de información #2: Solicitud

```
GET /api/sensors/pushMeasurement/1 HTTP/1.1
Host: ec2-54-70-229-71.us-west-2.compute.amazonaws.com:3000
Content-Length: 384
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Accept: application/json
User-Agent: python-requests/2.3.0 CPython/2.7.6

cypheredContent=rPqtrn7OWl0VNUFzxGJZsJNUs%2FntLCAA1lpWTPERP5mpXFhKcgkM
K1Cl3X0fW0N7LoFieYWdKt9VRpQUSC34LFECzasOiORfCMzh3k8KwmmdGfJTjZMXM5JjgK
uqzrcXF0OFhd12UmOywpsM39HyR6%2Bjwv%2FPl%2B4xSLrEASoO0IF7HIBwD1p5n3RlGQ
bxcE8YRzm%2B%2BBA6qHIBxPcnrMyS52iwQ0Xbeo7PfRP5U0fL39Qtg0xqnE0T%2BPAD4v
Zo9PuBw77R1dBEVHKORvx5bQHsSA%3D%3D&hash=0f4b16fa642bd2c376236efd9b5ce1
6ae3ec74efb10245183437f24720fe03ffGET /api/sensors/pushMeasurement/1
HTTP/1.1
```

Figura 53: Wireshark: Envío de información #2: Respuesta

```
HTTP/1.1 200 OK
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Download-Options: noopen
X-Content-Type-Options: nosniff
Content-Type: application/json; charset=utf-8
Content-Length: 36
ETag: W/"24-htQGVXW3GkCm9OaF//Ew+w"
Date: Sat, 22 Oct 2016 23:27:20 GMT
Connection: keep-alive

{"data":"All 4 measurements pushed"}
```

c. FindMyHash: Intercambio de llaves Se analizó el valor Hash del intercambio de llaves.

Figura 54: FindMyHash: Intercambio de llaves: Solicitud

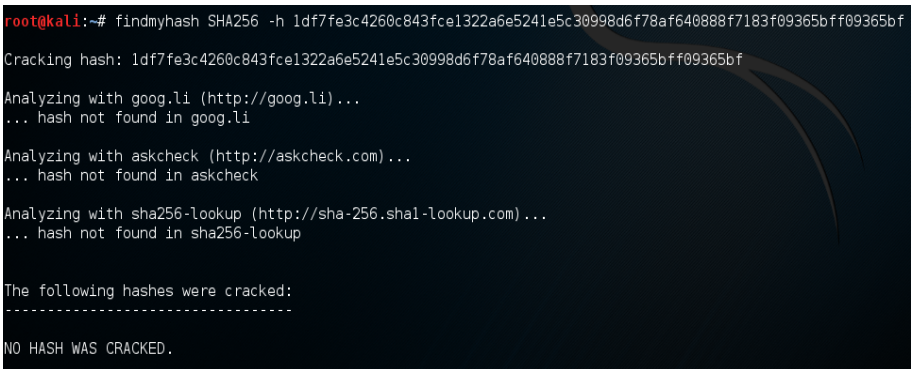
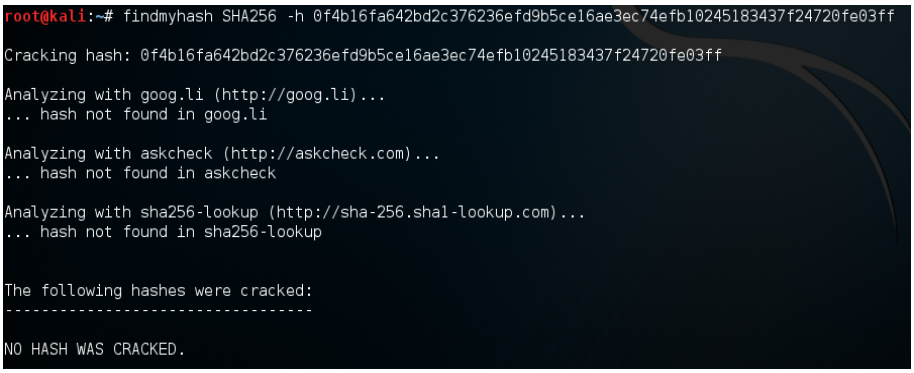


Figura 55: FindMyHash: Intercambio de llaves: Respuesta

d. FindMyHash: Cifrado de datos Se analizó el valor Hash del envío de información.

Figura 56: FindMyHash: Envío de datos #1: Respuesta

Figura 57: FindMyHash: Envío de datos #2: Respuesta



C. Almacenamiento de información y servicios web

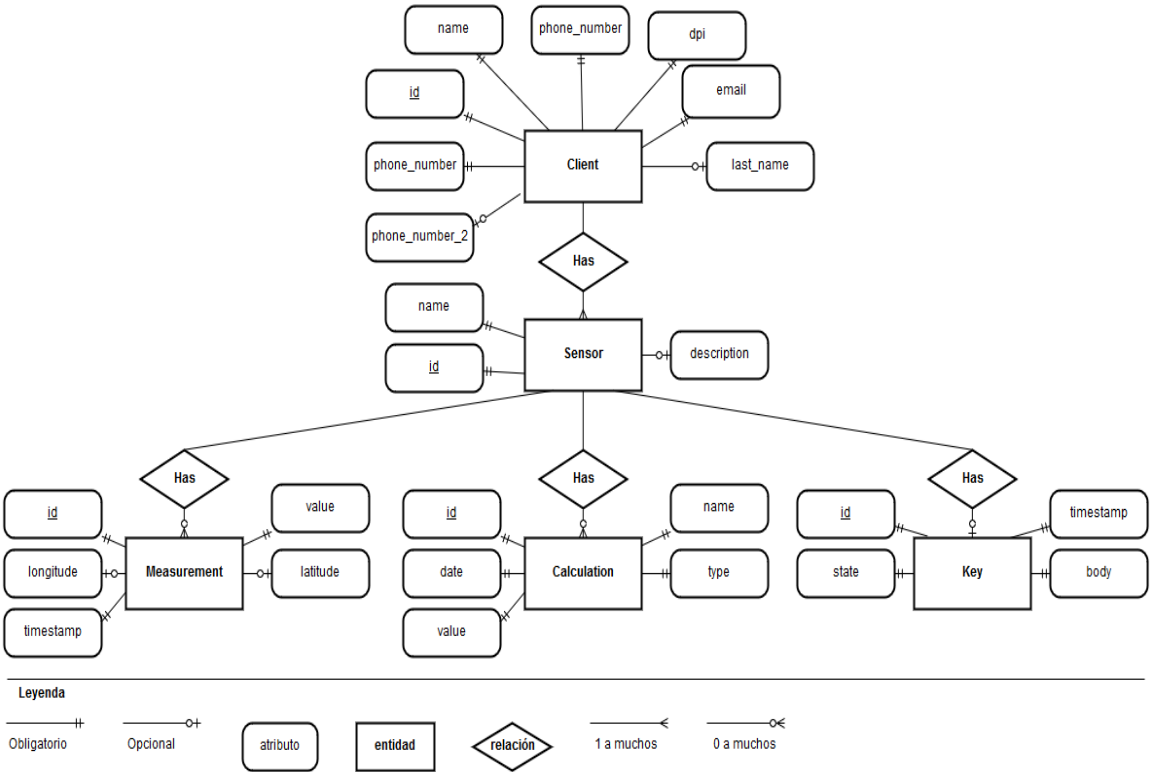
En el módulo de Almacenamiento de información y servicios web se obtuvieron los siguientes resultados.

Para este proyecto se realizaron dos implementaciones de una REST API. La primera versión del proyecto únicamente hace uso del DBMS PostgresSQL. La segunda versión utiliza dos DBMS distintos, PostgreSQL y HBase. En el resto del documento la primera versión se conocerá como la versión simple y la segunda versión se conocerá como la versión híbrida.

1. Bases de datos Para la versión simple del proyecto se implementó una base de datos para almacenar la información de consumo y la información proveniente del módulo de análisis. La base de datos fue implementada utilizando PostgreSQL 9.x. La base de datos está alojada en un servidor de Amazon Web Services con Sistema Operativo Ubuntu 14.04. El acceso a la misma es por medio de usuario y contraseña.

En la siguiente figura se encuentra el modelo entidad relación utilizado para implementar la base de datos.

Figura 58: Diagrama entidad relación para la versión simple

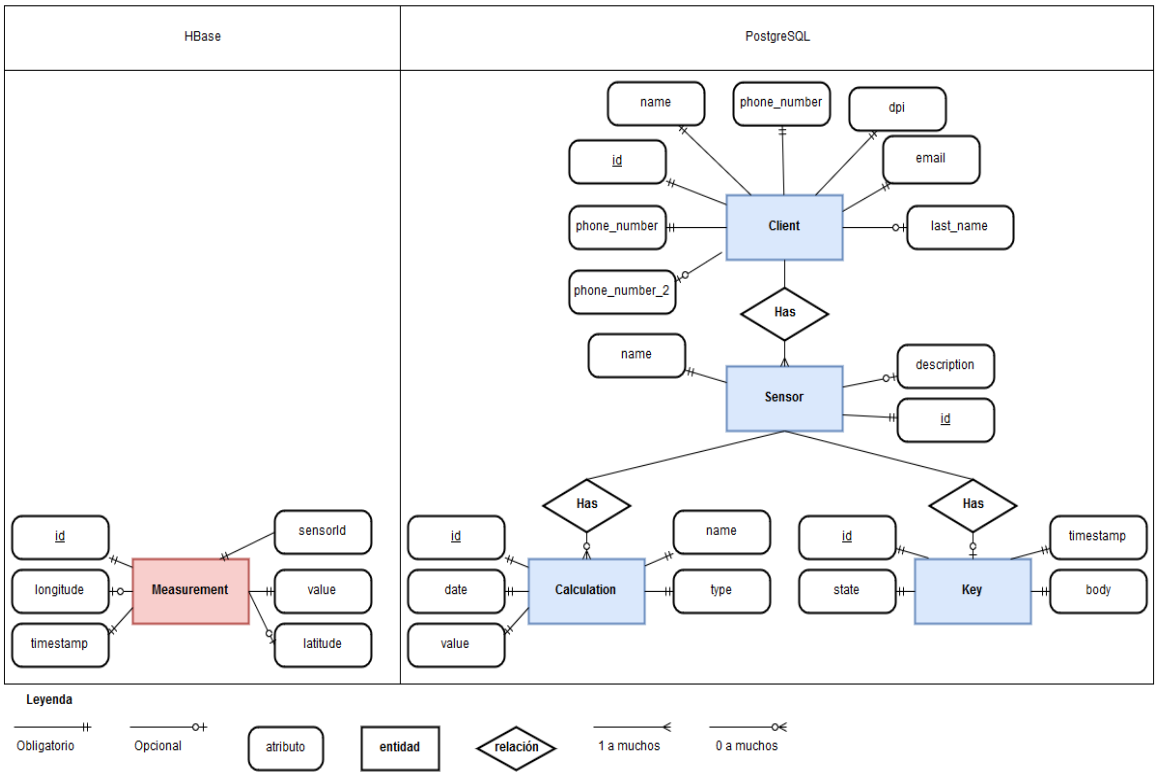


Para la versión híbrida del Proyecto se implementaron dos bases de datos distintas, una utilizando PostgreSQL 9.x y otra utilizando HBase 1.2.x. El acceso a las base de datos se establece de

la misma manera que en la versión simple. Para comunicarse con HBase se utilizó la REST API provista por el DBMS. Esta implementación se alojó en el mismo servidor que la versión simple del proyecto.

En la siguiente figura se encuentra el modelo entidad relación utilizado para implementar las bases de datos.

Figura 59: Diagrama entidad relación para la versión híbrida



En ambas versiones la entidad Client es utilizada para almacenar la información personal del usuario (nombre, apellido, número de teléfono, número de télefono 2, correo electronico). La entidad Client tiene una relación de uno a muchos con la Entidad Sensor, lo que permite que un usuario pueda monitorear varios hogares. La entidad Sensor representa al sensor dentro del sistema y sus atributos son utilizados únicamente para almacenar números de serie del sensor o el estado físico del mismo. La entidad Sensor tiene una relación de uno a muchos con la entidad Measurement y la entidad Calculation. Sensor también tiene una relación con Key, en esta relación Sensor puede tener o no una instancia de Key, que a su vez indica la existencia o no del cifrado entre el servidor y el sensor. Calculation es utilizado para almacenar la información de los análisis realizados por el módulo de análisis. En ambas versiones Measurement almacena la información de consumo energético, pero en la versión híbrida esta información se almacena en HBase. Para establecer esta

relación Measurement tiene un campo que hace referencia al Sensor que pertenece.

2. REST API En ambas versiones implementó una REST API para proveer acceso al módulo de Integración, módulo de Seguridad y módulo de Sensores. La REST API fue desarrollada utilizando el framework loopback versión 2.x sobre NodeJS versión 6.x. La REST API se encuentra alojada en el mismo servidor que la base de datos.

La REST API unicamente permite el acceso a la información y operaciones a cuatro de las cinco entidades. Dentro de loopback las entidades se conocen como modelos. Tanto para Cliente como Sensor se exponen todas las operaciones creación, lectura, actualización y eliminación, también conocidas como CRUD, para elementos individuales y no las colecciones. Adicionalmente para el modelo Sensor se exponen dos rutas adicionales provistas para la comunicación con el módulo de Seguridad y el módulo de Sensores. Para los modelos de Measurement y calculation únicamente se permite el acceso a las operaciones de creación y lectura.

Como se mencionó anteriormente en el proyecto se trabajó con dos DBMS. La conexión con el DBMS PostgreSQL se hizo por medio del conector soportado por el framework loopback. En el caso de HBase se configuró un conector personalizado a manera que el proyecto se comunicara con HBase por medio del API REST provisto por el DBMS.

3. Pruebas Se realizaron pruebas de carga sobre el punto de entrada para la inserción de nuevas medidas. Las pruebas de carga se realizaron sobre las dos versiones del proyecto. Las pruebas de carga se realizaron utilizando el software JMeter 3.x. Las pruebas se configuraron a manera de simular 800 usuarios ingresando una medida cada uno sobre un período de 10 segundos. El resultado de la prueba se exportó a un archivo con formato csv con información sobre cada una de las 800 transacciones realizadas. Para presentar los resultados más importantes de las pruebas por medio de gráficas se utilizó la aplicación web BlazeMeter.

A continuación se presentan las gráficas de resultados para las dos versiones del proyecto:

Figura 60: Tiempo de respuesta promedio en relación al tiempo para ambas versiones

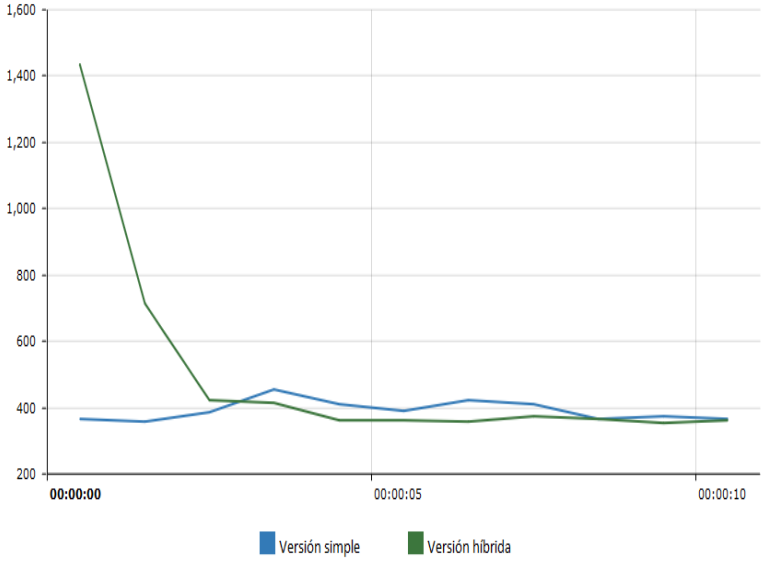


Figura 61: Gráfica de tiempo de respuesta en función de la cantidad de transacciones por segundo para la versión híbrida

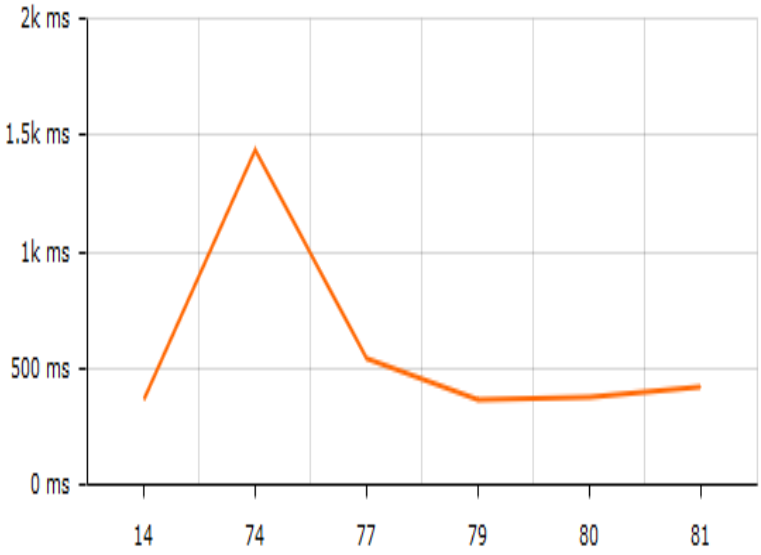
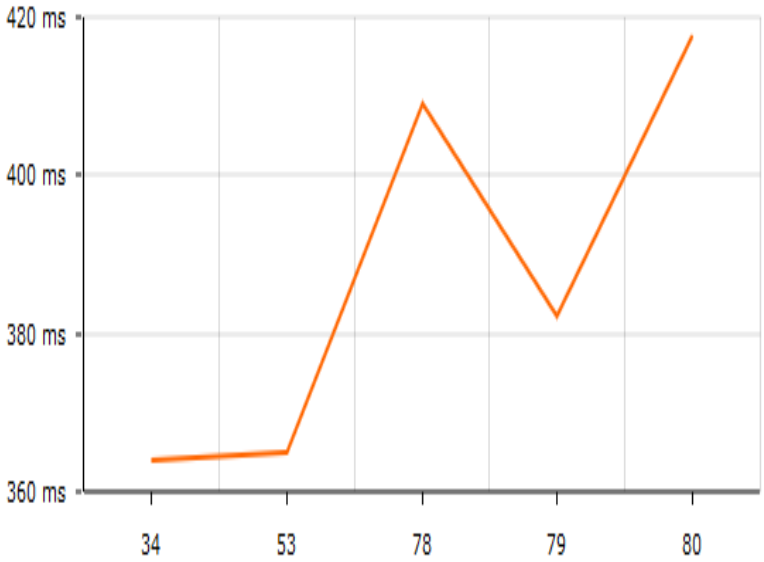
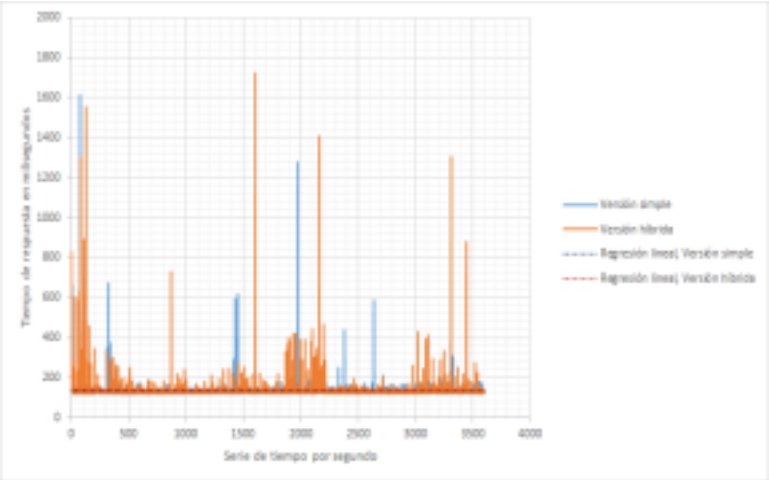


Figura 62: Gráfica de tiempo de respuesta en función de la cantidad de transacciones por segundo para la versión simple



Adicionalmente se probaron ambas versiones simulando un usuario durante una hora. Esto equivale a 3600 peticiones en una hora a cada versión de la REST API. Estas pruebas se hicieron desde una ambiente similar a la prueba anterior. Para la versión híbrida el tiempo promedio de respuesta es de 137.74 milisegundos, mientras que el tiempo de respuesta para la versión simple es de 132.45 segundos. Para los tiempos de respuesta de la versión híbrida se obtiene una desviación estándar de 67 milisegundos y para la versión simple se obtiene una desviación estándar de 39 milisegundos. A continuación se muestran los resultados.

Figura 63: Gráfica de tiempo de respuesta para la versión híbrida y para la versión simple



D. Análisis de datos

Para llevar a cabo las implementaciones de los métodos elegidos se usó Python como lenguaje de programación, haciendo uso de la biblioteca de clases Scikit Learn.

1. Análisis de algoritmos La complejidad de la mayoría de algoritmos usados en esta investigación ya han sido determinados en la literatura (Pedregosa, 2012). Por ello solamente se analizará la complejidad del método de Pham y su equipo. Más adelante se presenta un cuadro con las complejidades de tiempo de todos los métodos relevantes para este proyecto en el área de análisis de datos.

a. Método de Pham *et al.* Este método, como ya se mencionó anteriormente, se basa en realizar el llamado de una función hasta k veces, donde k es la cantidad máxima de clústeres posibles, dentro de la función mencionada. Lo que se procede a hacer es primero llamar al método sobre el cúal se está basando la agrupación, para obtener los centros, luego se pasa a convertir los datos obtenidos a un diccionario para que se pueda utilizar dentro del algoritmo; nótese que esta parte se realiza k_i veces, donde $k_i \leq k$, luego de ello se pasa realizar una sumatoria sobre la norma de la resta del $centro_i$ y un clúster sobre el que se está iterando. Este grupo tendría una complejidad de k_i, se hace una evaluación sbre la cantidad actual de grupos que está trabajando, cuando esta es diferente de 1 y la sumatoria sobre la norma es distinta de 0, se divide esta última por el valor que arroja el siguiente código:

Figura 64: Pseudocódigo para el cálculo de factor dentro del método de Pham *et al.*

```
alfa(k, n_d):
    if k == 2:
        return 1 - (3/(4*n_d))
    else:
        return alfa(k-1, n_d) + (1-alfa(k-1,n_d))/6
```

Donde k, y n_d son el número de dimensiones con el que se está trabajando, n_d se mantuvo en 2 y k era la cantidad k_i para la iteración en la que se estuviera trabajando.

Dado que el proceso anterior se repite hasta para k veces, y el código mostrado en la figura 64, se realiza de forma recursiva dentro de la declaración del *else*, puesto que por cada llamada esta se invoca 2 veces más, es decir que por cada llamada de la función se generan 2^n invocaciones más a esta, lo cual da como resultado una complejidad de.

$$O(k * 2^n) \tag{VI..3}$$

Cabe decirse que se ha ignorado la complejidad que agrega cada suma, resta, división o multiplicación puesto que estas no se consideran al ser de $O(1)$, que es menor a las funciones presentadas.

2. Predicción Para esta sección se usaron los algoritmos de Árboles de Decisión o DT (por sus siglas en inglés Decision Tree) y Máquina de Vectores de Soporte en su implementación para regresión conocida como SVR (por sus siglas en inglés Support Vector Regression).

a. Comparación Al realizar la prueba con 8,192 datos se obtuvieron los siguientes resultados para el DT y el SVR de forma respectiva

Figura 65: Árbol de decisión con 8,192 datos.

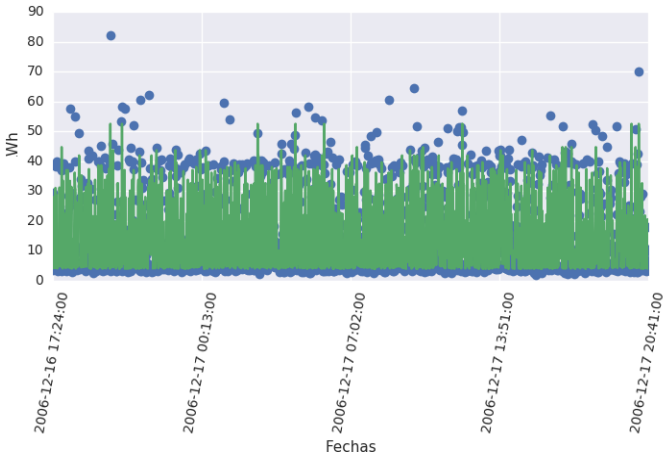
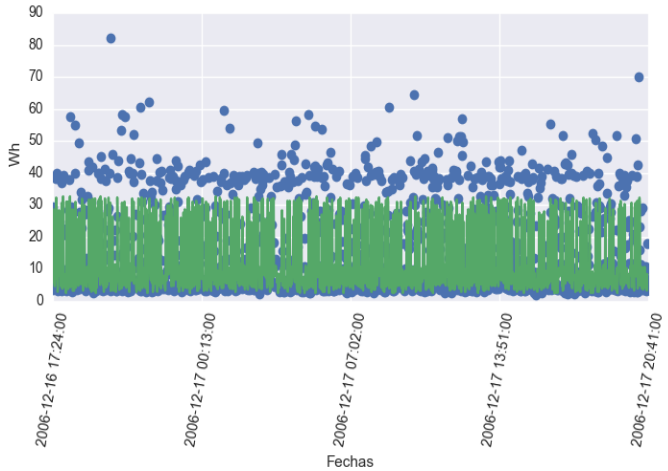


Figura 66: Máquina de vectores de soporte 8,192 datos.



En cuanto a los resultados de las pruebas con 16,384 datos se pueden apreciar los resultados

del DT y SVR en las siguientes gráficas de forma respectiva.

Figura 67: Árbol de decisión con 16,384 datos.

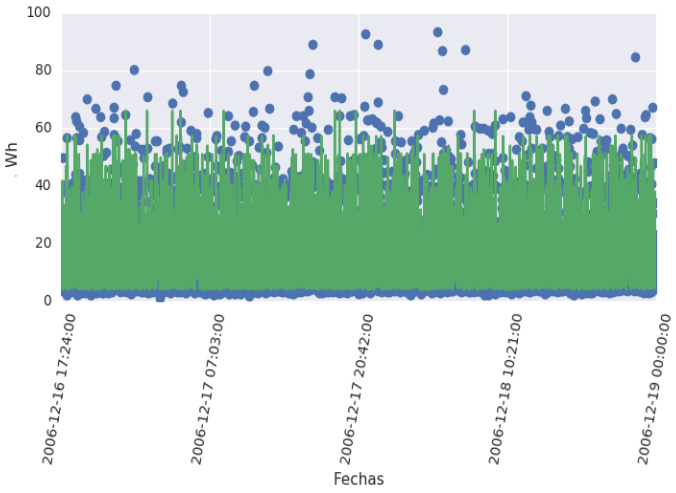
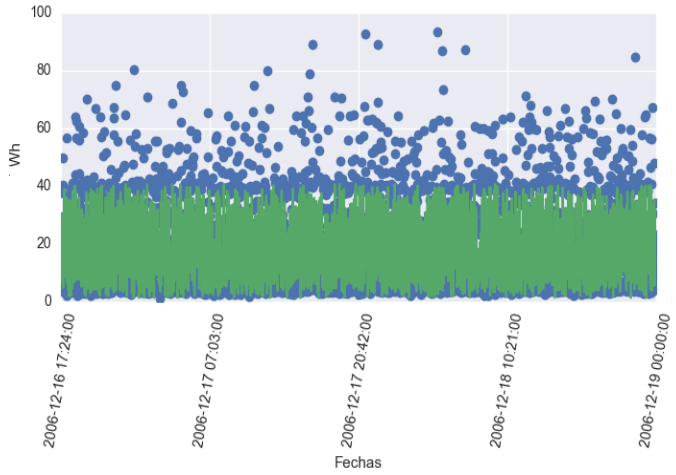


Figura 68: Máquina de vectores de soporte 16,384 datos.



Para el caso de 32,768 datos, con el DT y el SVR se obtuvieron las siguientes gráficas como resultado, respectivamente.

Figura 69: Árbol de decisión con 32,768 datos.

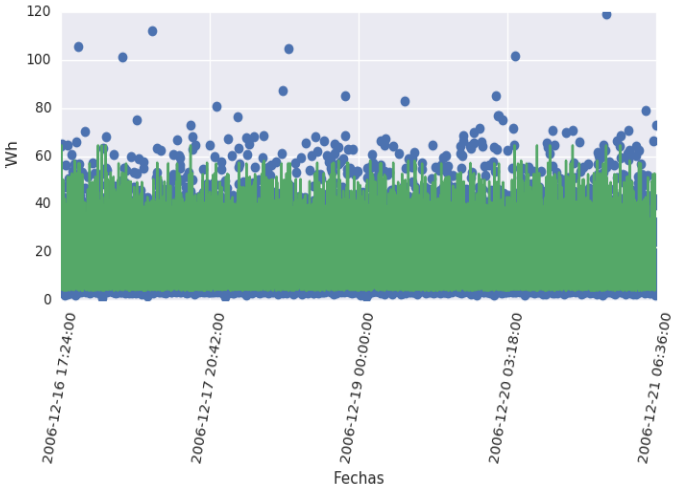
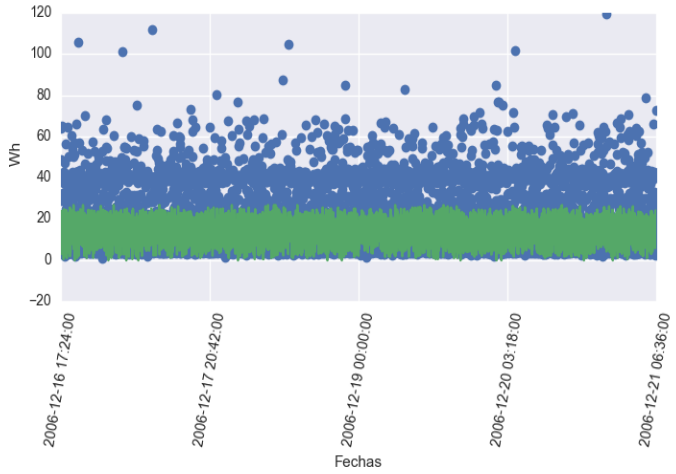


Figura 70: Máquina de vectores de soporte 32,768 datos.



En lo que respecta al caso de 65,536 datos se obtuvo el siguiente resultado con el DT y SVR correspondientemente.

Figura 71: Árbol de decisión con 65,536 datos.

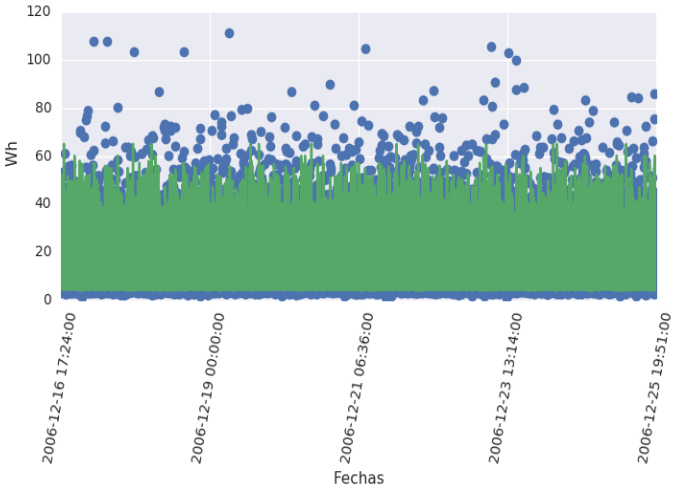
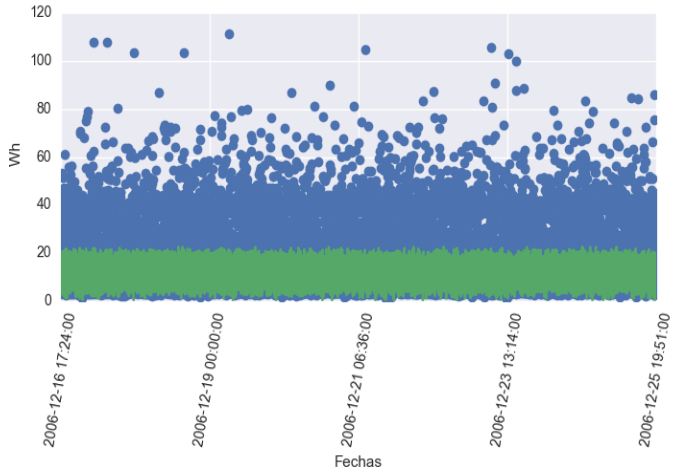


Figura 72: Máquina de vectores de soporte 65,536 datos.



Cuando se usaron 131,072 datos el DT y SVR produjeron los siguientes resultados gráficamente.

Figura 73: Árbol de decisión con 131,072 datos.

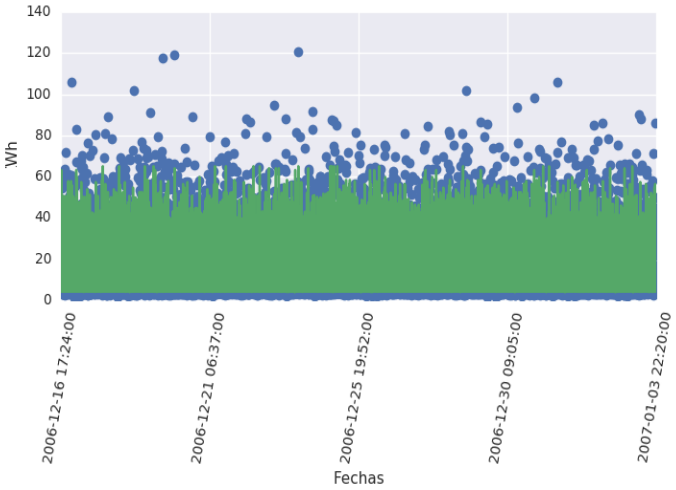
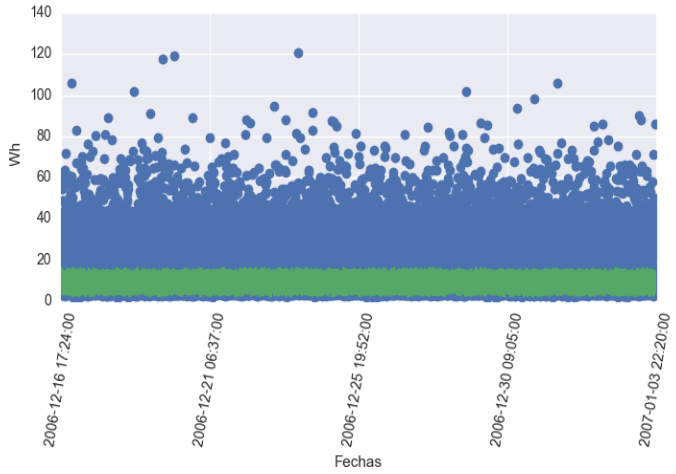


Figura 74: Máquina de vectores de soporte 131,072 datos.



Al usar 262,144 los siguientes resultados fueron dados por el DT y el SVR.

Figura 75: Árbol de decisión con 262,144 datos.

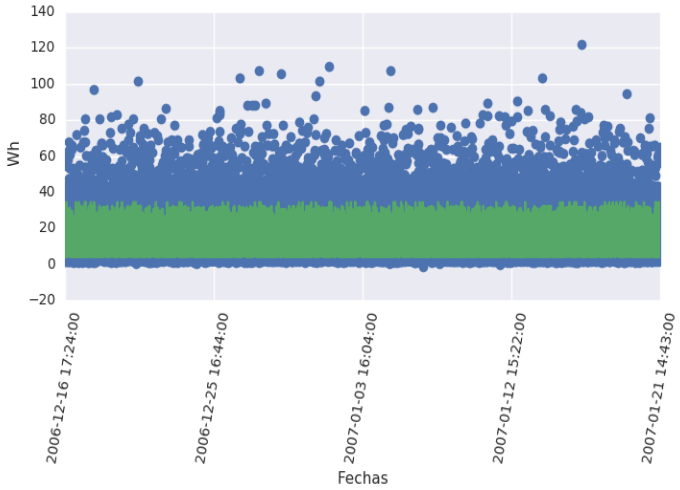
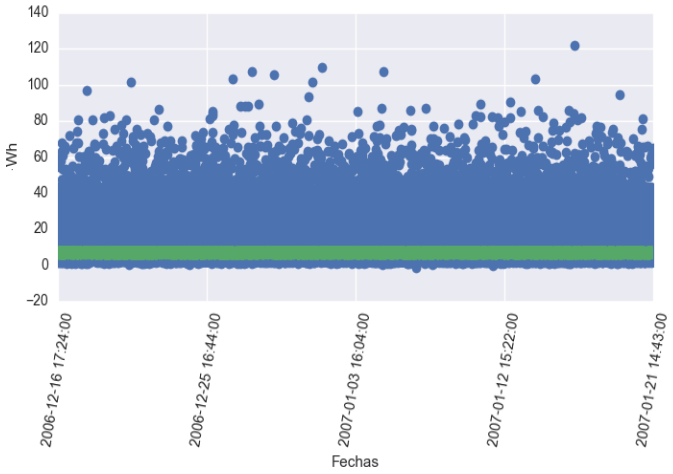


Figura 76: Máquina de vectores de soporte 262,144 datos.



Para el uso de 524,288 datos se dio lugar a las gráficas siguientes como resultado del DT y del SVR.

Figura 77: Árbol de decisión con 524,288 datos.

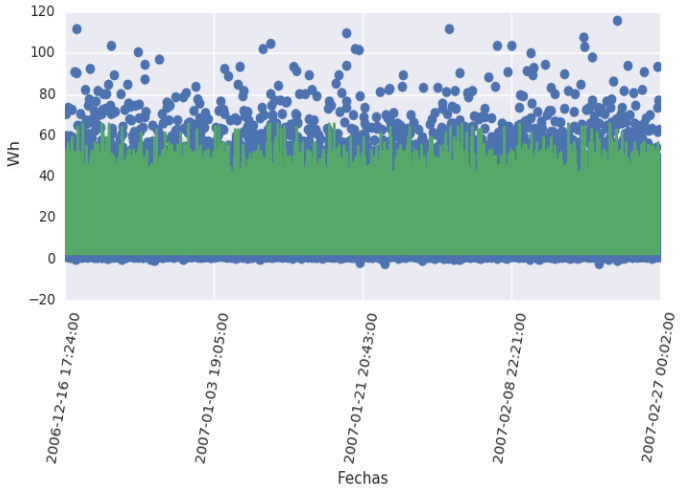
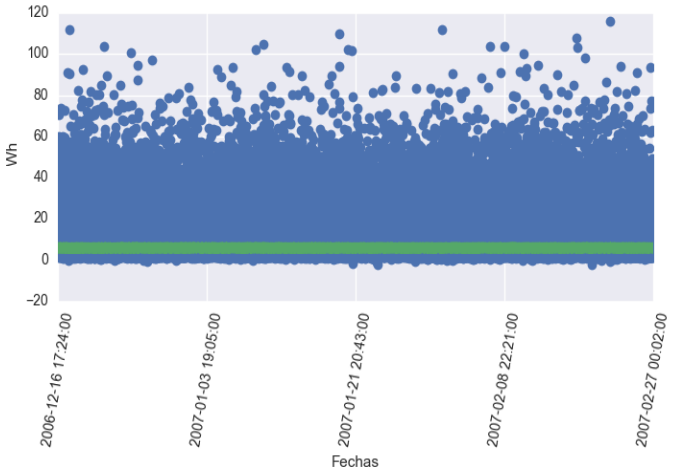


Figura 78: Máquina de vectores de soporte 524,288 datos.



Finalmente, para el caso donde se hizo uso de 1,048,575 datos se dieron lugar a la siguiente gráfica como resultado de parte del DT.

Figura 79: Árbol de decisión con 1,048,575 datos.

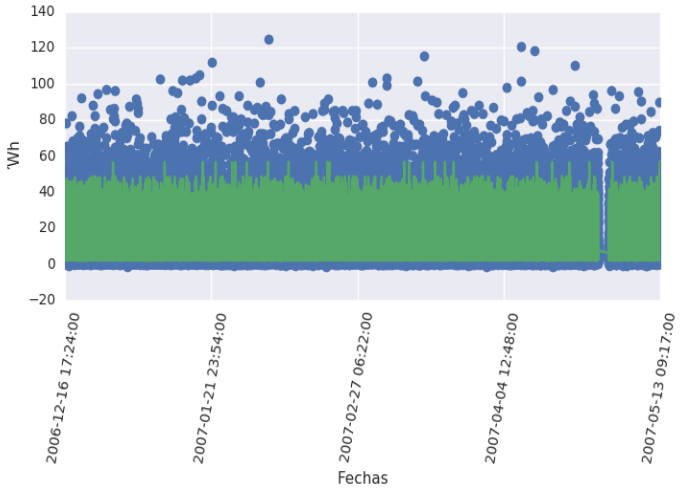
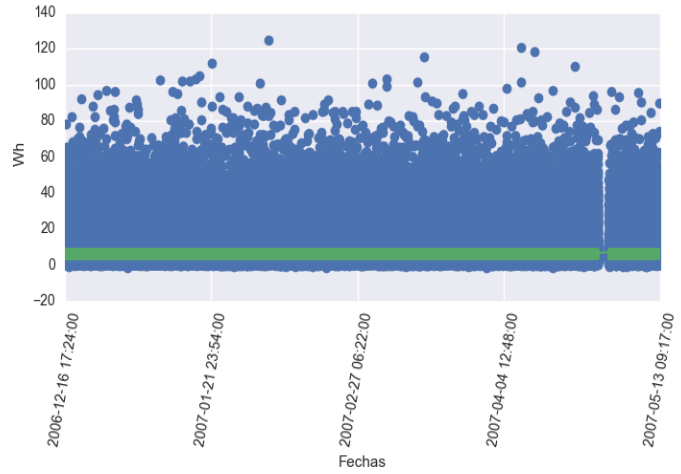
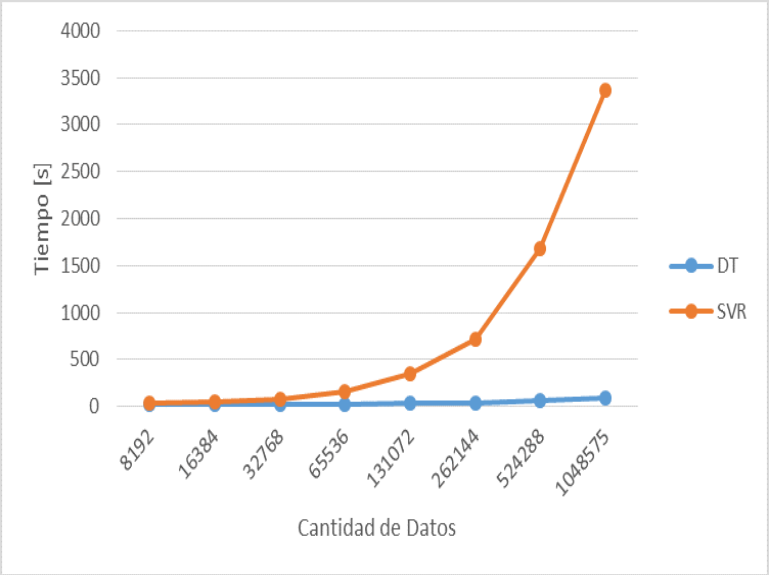


Figura 80: Máquina de vectores de soporte 1,048,575 datos.



Además, se muestran tanto los tiempos como la exactitud que presentó cada uno de los términos en cada caso con diferente cantidad de datos.Nótese que la exactitud que se presenta está dada en base al coeficiente r^2.

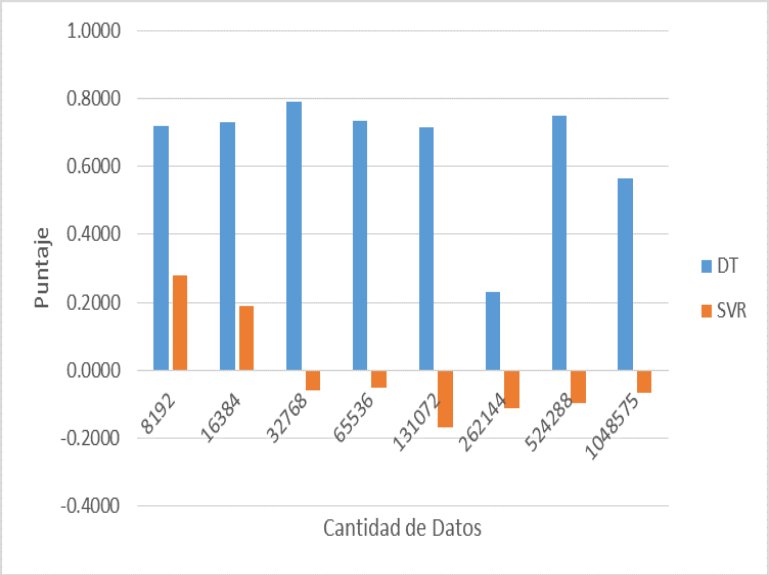
Figura 81: Tiempos de DT y SVR con diferente cantidad de datos.



Cuadro 33: Tiempo medido en segundos de las ejecuciones de DT y SVR

Cantidad De Datos	**Árboles de Decisión** [s]	**Máquina de Vectores de Soporte** [s]
8192	22.9974	35.7533
16384	23.3831	55.0245
32768	24.4123	79.5259
65536	27.0550	156.36262
131072	31.1146	342.9769
262144	39.9521	721.8081
524288	58.1358	1678.5656
1048575	93.5641	1972.7190

Figura 82: Puntaje de DT y SVR con diferente cantidad de datos.



Cuadro 34: Puntaje de las ejecuciones de árboles de decisión y máquina de vectores de soporte

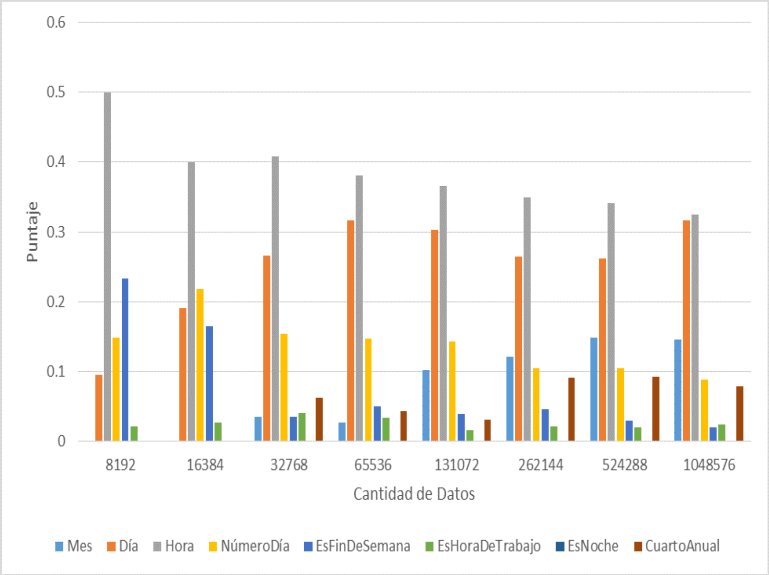
Cantidad De Datos	**Árboles de decisión**	**Máquina de vectores de soporte**
8192	0.7195	0.2805
16384	0.7297	0.1887
32768	0.7909	-0.0591
65536	0.7345	-0.0502
131072	0.7146	-0.1667
262144	0.2292	-0.1131
524288	0.7489	-0.0957
1048575	0.5630	-0.0661

Para que ambos algoritmos presentandos se propusieron siete características extraidas del dato de la fecha, siendo estos mes, dia, hora, numeroDia, esFindeSemana, esHoraDeTrabajo, esNoche, cuartoAnual. La siguiente tabla presenta las características que fueron seleccionadas para cada una de las iteraciones, a través de usar un método basado en el algoritmo de Bosques Aleatorios. La figura consecuente muestra de forma gráfica el puntaje que obtuvo cada característica para las distintas cantidades de datos.

Cuadro 35: Características seleccionadas para las ejecuciones de árboles de decisión y máquina de vectores de soporte

Cantidad de datos	**Característica seleccionada**
8192	hora, numeroDia, esFindeSemana
16384	dia, hora, numeroDia, esFindeSemana
32768	dia, hora, numeroDia
65536	dia, hora, numeroDia
131072	dia, hora, numeroDia
262144	dia, hora
524288	mes, dia, hora
1048575	mes, dia, hora

Figura 83: Puntaje de las características para las diferentes iteraciones.



En la tabla que a continuación se presenta, se dan a conocer los parámetros que fueron elegidos de forma automática para la aplicación del método de máquina de vectores de soporte. Dicha selección se hizo a través del método de búsqueda de rejilla (Grid Search en ingés) usando validación cruzada (Cross Validation en inglés), con cinco pliegues o subconjuntos (conocidos como folds en ingés).

Cuadro 36: Parámetros para la máquina de vectores de soporte en las diferentes iteraciones

Cantidad De Datos	**Épsilon**	**C**
8192	0.1	256
16384	0.1	0.125
32768	0.1	1024
65536	0.5	512
131072	0.5	8
262144	0.5	16
524288	0.5	32
1048575	0.5	1

b. Uso con datos reales del sensor Al obtener datos, durante 23 horas consecutivas con el sensor desarrollado como parte del proyecto, se obtuvieron los siguientes resultados.

Se presentan a continuación las gráficas producidas con la aplicación en la predicción tanto del método de DT como del SVR.

Figura 84: Árbol de decisión con datos de 23 horas de lectura del sensor.

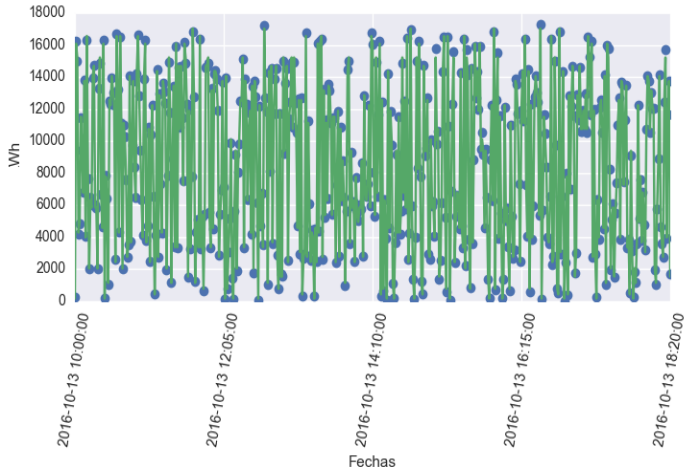
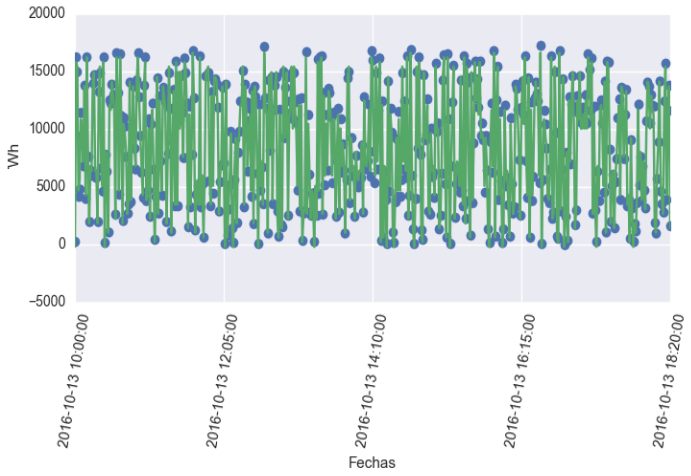


Figura 85: Máquina de vectores de soporte con datos de 23 horas de lectura del sensor.



En cuanto a los tiempos que se tomaron los métodos se obtuvieron los siguientes resultados.

Cuadro 37: Tiempo medido en segundos de las ejecuciones de DT y SVR con 23 horas de datos del sensor.

Árboles de Decisión [s]	**Máquina de Vectores de Soporte** [s]
0.6639	10.5389

En lo referente a la exactitud dada por el coeficiente r^2, para ambos métodos se obtuvo lo siguiente.

Cuadro 38: Puntaje de las ejecuciones de árboles de decisión y máquina de vectores de soporte con 23 horas de datos del sensor

Árboles de Decisión	**Máquina de Vectores de Soporte**
0.9992	0.9936

Nótese que al usar estos datos totalmente mapeados desde el sensor, se obtuvieron los siguientes pesos de las características.

Cuadro 39: Pesos de características en datos desde el sensor

Característica	**Peso**
mes	0
día	0.476875947625
hora	0.200814144389
númeroDía	0.231688684792
esFinDeSemana	0.0516056780705
esHoraDeTrabajo	0.0390155451232
esNoche	0
cuartoAnual	0

Entonces como se puede apreciar en el cuadro 39, las variables día, hora y númeroDía fueron las usadas dado que fueron las que más puntaje de peso presentaron.

De igual manera, se presentan los valores para ϵ y C con los que se trabajaron para estos datos en el método de máquina de vectores de soporte

Cuadro 40: Parámetros para la máquina de vectores de soporte con datos del sensor

Cantidad De Datos	**Épsilon**	**C**
2501	0.1	2048

3. Clasificación Para esta parte se usaron los algoritmos de KMeans y Hierarchical Clustering. Tras la implementacin de estos se obtuvieron los resultados siguientes.

a. Elección de método para cantidad de clusters Se compararon los métodos de Silueta y el sugerido en el trabajo de Phem *et al.* Se ejecutaron ambos métodos para determinar la cantidad de agrupaciones que se usaría. Para seleccionar el más adecuado se tomó en cuenta el tiempo de ejecución de ambos.

Figura 86: Ejecución de la implementación del método de Phem *et al.*

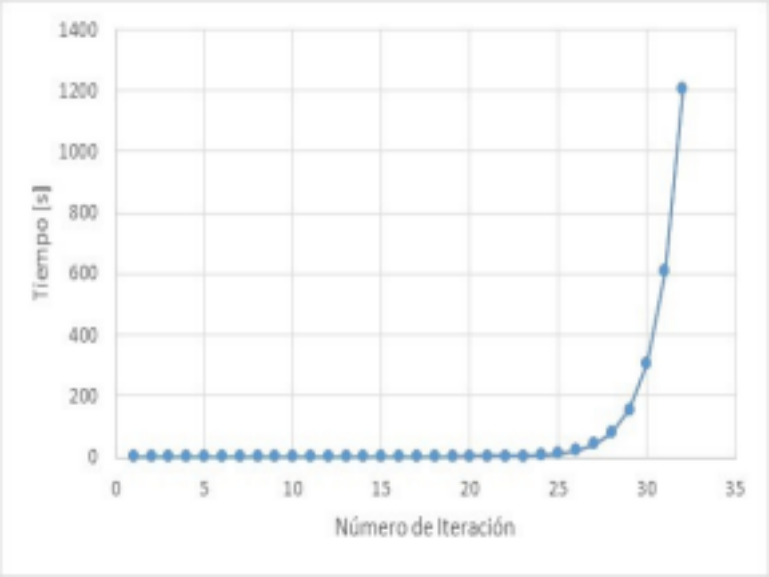
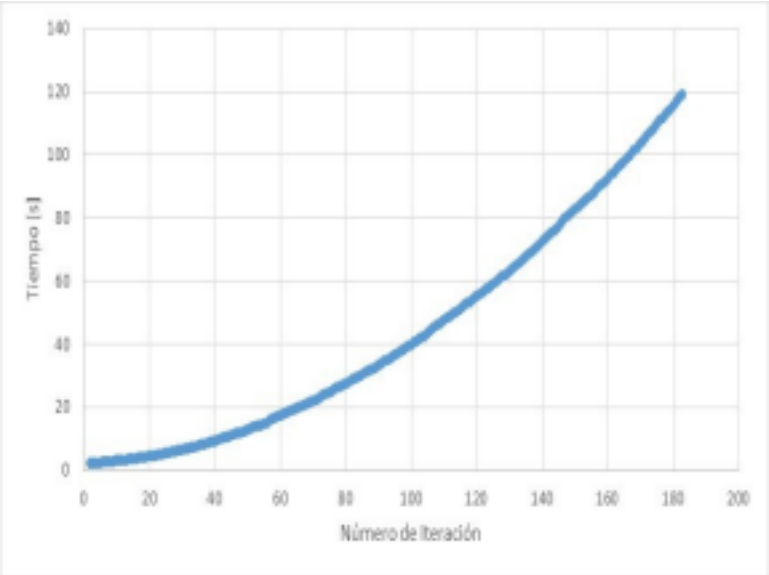


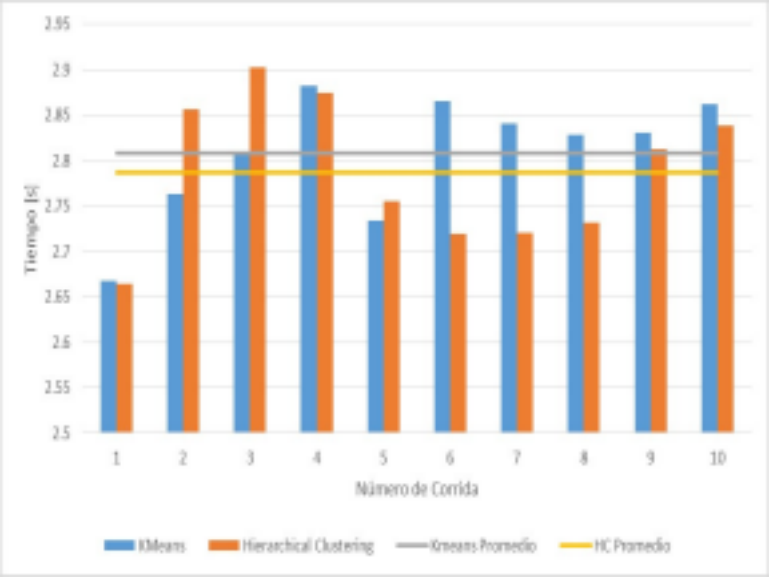
Figura 87: Ejecución de la del método de la silueta.



Dado el tiempo y complejidad de los algoritmos presentados, se optó por hacer uso del método de la silueta.

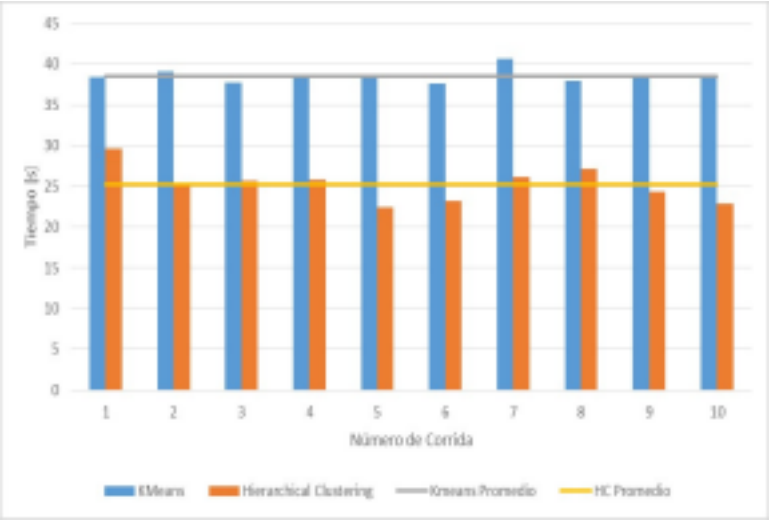
b. Comparación Se realizó la comparación de los métodos de agrupación seleccionados a través de 10 iteraciones diferentes variando la cantidad máxima posible de clusters. A continuación se presentan los resultados de los tiempos de ejecución de los algoritmos.

Figura 88: Tiempos de ejecución de KMeans y Hierarchical Clustering con 10 clústeres máximo.



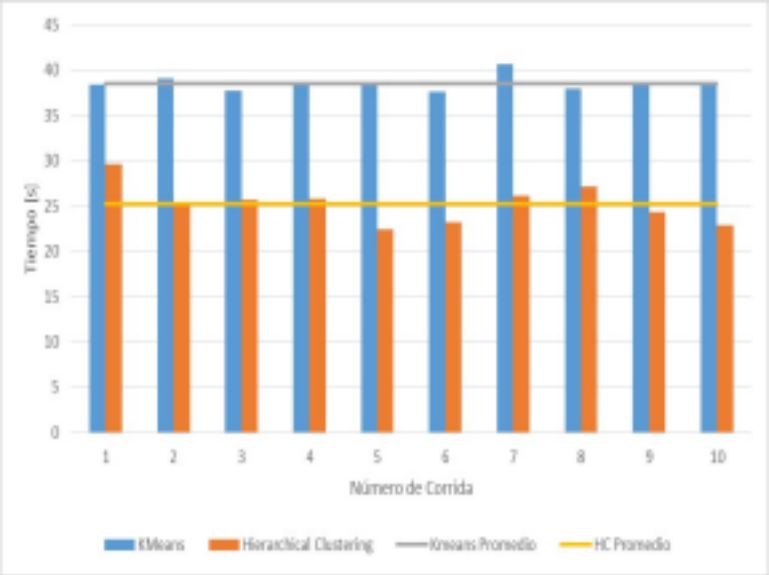
Con 10 como la variable de cantidad máxima de clusters posible el método de Hierarchical Clustering resultó más rápido que el de KMeans por 0.0207 segundos en promedio. En la siguiente gráfica se presentan los tiempos de ejecución y la diferencia en los promedios de estos para 100 grupos.

Figura 89: Tiempos de ejecución de KMeans y Hierarchical Clustering con 100 clústeres máximo..



Al ser el máximo de clusters en 100 se apreció que el método de Hierarchical Clustering fue más rápido que el de KMeans con una diferencia de 13.3143 segundos en promedio. A continuación se puede apreciar de mejor manera la diferencia en los tiempos de ejecución y así como en los tiempos promedio de ambos métodos para 183 agrupaciones.

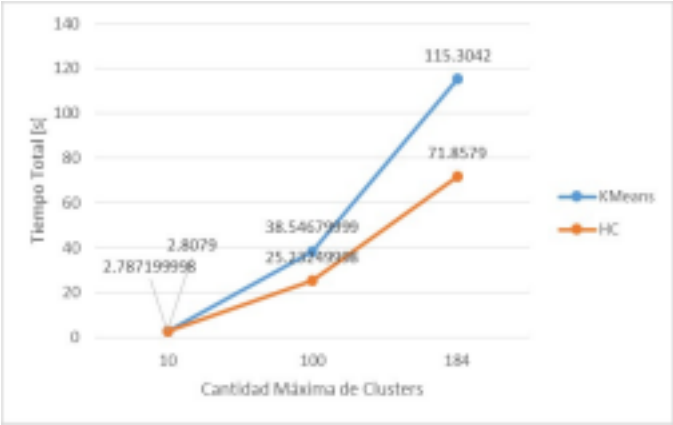
Figura 90: Tiempos de ejecución de KMeans y Hierarchical Clustering con 183 clústeres máximo..



Para el caso de 183 clusters como máximo se notó que el método de Hierarchical Clustering fue de ejecución más rápida que el de KMeans por 43.4463 segundos en promedio. Se presenta ahora la gráfica donde se ejemplifica la diferencia mencionada tanto en promedio como en cada una de las diferentes iteraciones.

Las diferencias en los tiempos de ejecución con las diferentes cantidades máximas de clusters se aprecian en la siguiente gráfica.

Figura 91: Tiempos promedio de ejecución de KMeans y Hierarchical.



Así mismo se presentan los resultados de las cantidades de clusters a utilizar sugerido por el método de la silueta tanto al ser aplicado con el método de KMeans como con el de Hierarchical Clustering, los cuales resultan ser la misma cantidad para ambos casos.

Cuadro 41: Cantidad sugerida de clústeres para cada método.

Cantidad máxima de clusters	10	100	184
Kmeans	2	99	183
Hierarchical Clustering	2	99	183

E. Integración

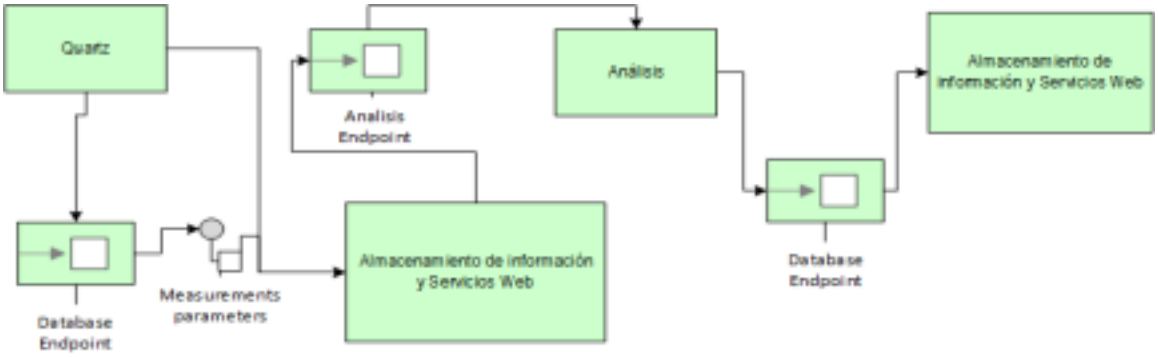
Los resultados del módulo de Integración generaron decisiones para lograr un buen resultado.

1. Plataforma de comunicación Para establecer comunicación entre el módulo de Almacenamiento de Información y Servicios Web y el módulo de Análisis de Datos se decidió utilizar Apache Camel para establecer el ambiente. Esto se alojó en un servidor de Amazon Web Services (AWS). Para establecer comunicación con el módulo de análisis de datos se implementó ActiveMq ya que se puede comunicar con Python fácilmente, que es el lenguaje que utiliza el módulo de Ańalisis.

2. Protocolos de comunicación Para la comunicación con los dos módulos se hicieron acuerdos en los cuales se detalla los protocolos o el conjunto de reglas que se deben seguir en la comunicación. Se proponen dos acuerdos, uno para el módulo de Análisis de Datos y otro para Almacenamiento de Información y Servicios Web. El acuerdo de Análisis de Datos detalla el formato que se espera del CSV (Coma Separeted Values) que contiene la información para cada análisis, así como la manera en que se utiliza este módulo y el formato de la respuesta de este módulo. En el acuerdo para Almacenamiento de Información y Servicios Web se establece la URL que se va a consumir asi como el tipo de solicitud, el formato de los parámetros que se esperan en esta solicitud y el formato de la respuesta que se espera.

3. Comunicación entre módulos Para realizar la comunicación de información entre el módulo de Almacenamiento de la Información y Servicios Web y el módulo de Análisis de Datos se propone el siguiente flujo de información representado con un diagrama de patrones de integración que ya fueron definidos en el marco teórico.

Figura 92: Diagrama general de integración



En este diagrama se muestran tres componentes, Quartz, Almacenamiento de información y Servicios Web y Análisis. El componente de Quartz es el responsable de iniciar el flujo de información para lograr realizar los tres procesos de análisis. Estos tres procesos de análisis son: predicción, entrenamiento de predicción y agrupación (clustering). El entrenamiento de predicción y clustering requieren crear parámetros para consumir un servicio web. Por esta razón el flujo pasa "Database Endpoint", este punto extremo crea "Measurements parameters" que representa los parmámetros del servicio web a consumir. El proceso de predicción no requiere de parámetros antes de consumir un servicio web, por lo que se conecta directamente con el componente que representa los servicios web "Almacenamiento de Información y Servicios Web". La respuesta del servicio web se envía a "Analisis Endpoint", este punto extremo es el responsable de traducir la información recibida y enviarla al componente que representa el módulo de análisis "Análisis". Por último la respuesta de análisis se recibe en "Database Endpoint", quien formatea estos resultados en el formato para almacenarse en la base de datos.

Para confirmar que se cumple este flujo de información y que en realidad se establece la comunicación entre los módulos mencionados, se realizaron pruebas por cada proceso de análisis en diferentes condiciones. En los cuadros 42, 45 y 48 se muestran los resultados bajo condiciones ideales. Las condiciones ideales implican una conexión a internet estable y el almacenamiento de más de un sensor, consumos de energía para cada sensor y la misma cantidad de registros de consumo por sensor. Los cuadros 43, 46 y 49 muestran los resultados en condiciones desfavorables. A diferencia de las condiciones ideales, estas condiciones contaban con una conexión a internet con menor desempeño y no se contaba con la misma cantidad de registros de consumo para cada sensor. Los cuadros 44, 47 y 50 muestran una variable que se controló. A diferencia de las condiciones anteriores, en estas se controló el acceso a internet, provocando una falla en la comunicación. Cada cuadro muestra la parte del proceso que se estaba observando, la cantidad de mensajes enviados y recibidos, la cantidad de mensajes esperados y el porcentaje de cuantos mensajes se esperaban con cuantos se recibieron o enviaron. Todos las pruebas se realizaron con información de cuatro

sensores ficticios, es decir la información solo se colocó en la base de datos para realizar estas pruebas.

Para el análisis de clustering se utilizó quartz para iniciar el proceso una vez cada minuto. De estas pruebas se obtuvieron los siguientes resultados.

Cuadro 42: Resultados pruebas para proceso de clustering

Parte de proceso	**Mensajes obtenidos**	**Mensajes esperados**	**Porcentaje**
Solicitud de consumo	132	132	100 %
Respuestas con consumos	132	132	100 %
Envíos a análisis	132	132	100 %
Respuestas de análisis	132	132	100 %
Solicitudes para guardar	528	528	100 %
Total	1056	1056	100 %

Cuadro 43: Resultados pruebas para proceso de clustering con internet de menor desempeño

Parte de proceso	**Mensajes obtenidos**	**Mensajes esperados**	**Porcentaje**
Solicitud de consumo	168	168	100 %
Respuestas con consumos	165	168	98 %
Envíos a análisis	165	168	98 %
Respuestas de análisis	0	168	0 %
Solicitudes para guardar	0	672	0 %
Total	498	1344	37 %

Cuadro 44: Resultados pruebas para proceso de clustering con señal de internet interrumpida

Parte de proceso	**Mensajes obtenidos**	**Mensajes esperados**	**Porcentaje**
Solicitud de consumo	108	108	100 %
Respuestas con consumos	75	108	69 %
Envíos a análisis	75	108	69 %
Respuestas de análisis	0	108	0 %
Solicitudes para guardar	0	432	0 %
Total	258	864	30 %

Para confirmar que este flujo de información funciona correctamente se hicieron pruebas para contar los mensajes enviados y recibidos con los módulos de Almacenamiento de Información y Servicios Web y el módulo de Análisis. El proceso comienza con establecer comunicación con el módulo de Almacenamiento de Información y solicitar información de consumo (fila uno). La fila dos muestra la cantidad de respuestas que se obtuvieron al consumir el servicio web del cual se obtiene la información de consumo. Las siguientes dos filas muestran la comunicación con el módulo de análisis. Se contó cuantos mensajes se enviarion a este módulo y cuantas respuestas se obtuvieron. Por último se muestra la cantidad de solicitudes que se hicieron al servicio web que se encarga de guardar los resultados de los análisis. Este número es mayor ya que el análisis provee una respuesta por cada sensor, y se debe guardar la respuesta de cada sensor.

Cuadro 45: Resultados pruebas para proceso de entrenamiento de predicción

Parte de proceso	**Mensajes obtenidos**	**Mensajes esperados**	**Porcentaje**
Solicitud consumo	132	132	100 %
Respuestas con consumos	132	132	100 %
Envíos a análisis	528	528	100 %
Recibir de análisis	528	528	100 %
Total	1320	1320	100 %

Cuadro 46: Resultados pruebas para proceso de entrenamiento de predicción con internet de menor desempeño

Parte de proceso	**Mensajes obtenidos**	**Mensajes esperados**	**Porcentaje**
Solicitud consumo	168	168	100 %
Respuestas con consumos	165	168	98 %
Envíos a análisis	660	672	98 %
Recibir de análisis	660	672	98 %
Total	1653	1680	98 %

Cuadro 47: Resultados pruebas para proceso de entrenamiento de predicción con señal de internet interrumpida

Parte de proceso	**Mensajes obtenidos**	**Mensajes esperados**	**Porcentaje**
Solicitud consumo	108	108	100 %
Respuestas con consumos	75	108	69 %
Envíos a análisis	300	432	69 %
Recibir de análisis	300	432	69 %
Total	783	1080	72 %

Para el proceso de entrenamiento de predicción se realizaron la misma cantidad de pruebas. Para este proceso el flujo de información inicia de la misma manera, se hace una solicitud para recibir el consumo de los sensores. Una diferencia que vale la pena mencionar es la comunicación con el módulo de análisis para este proceso. Para cada sensor que se encuentre en los consumos se va a realizar el entrenamiento. Por esta razón se envían más mensajes al módulo de análisis. El módulo de análisis responde una vez por cada envío que se le hace y para este proceso no es necesario almacenar información.

Cuadro 48: Resultados pruebas para proceso de predicción en condiciones ideales

Parte de proceso	**Mensajes obtenidos**	**Mensajes esperados**	**Porcentaje**
Solicitud sensores	132	132	100 %
Respuestas con sensores	132	132	100 %
Envíos a análisis	528	528	100 %
Recibir de análisis	528	528	100 %
Solicitudes para guardar	528	528	100 %
Total	1848	1848	100 %

Cuadro 49: Resultados pruebas para proceso de predicción con internet de menor desempeño

Parte de proceso	**Mensajes obtenidos**	**Mensajes esperados**	**Porcentaje**
Solicitud sensores	168	168	100 %
Respuestas con sensores	165	168	98 %
Envíos a análisis	660	672	98 %
Recibir de análisis	660	672	98 %
Solicitudes para guardar	660	672	98 %
Total	2313	2352	98 %

Cuadro 50: Resultados pruebas para proceso de predicción con señal de internet interrumpida

Parte de proceso	**Mensajes obtenidos**	**Mensajes esperados**	**Porcentaje**
Solicitud sensores	108	108	100 %
Respuestas con sensores	75	108	69 %
Envíos a análisis	300	432	69 %
Recibir de análisis	300	432	69 %
Solicitudes para guardar	300	432	69 %
Total	1083	1512	72 %

Este proceso inicia de distinta manera en comparación a los otros dos. Este flujo de información inicia solicitando información sobre los sensores. En los cuadros se muestra la cantidad de solicitudes hechas para la información de los sensores y la cantidad de respuestas recibidas. Al igual que el proceso anterior, este proceso debe enviar un mensaje por cada sensor que se encuentre en los resultados, esto es porque la respuesta debe ser por cada sensor. Por último cada respuesta que se obtuvo del módulo de análisis se almacena en la base de datos.

F. Interfaz de usuario

1. Estudio de usabilidad I

Cuadro 51: Tarea 1. Ingreso a la aplicación

Usuario	**Tiempo(seg)**	**Observaciones**
1	30	
2	35	Tuvo dificultades para posicionar el puntero.
3	15	
4	54	Tuvo problemas para ingresar correctamente las credenciales
5	18	
6	12	
7	25	
8	38	Tuvo problemas para ingresar las credenciales
9	26	
10	45	Tuvo problemas con el ingreso de datos.

Cuadro 52: Tarea 2. Encontrar consumo en kWh en el mes

Usuario	**Tiempo(seg)**	**Observaciones**
1	11	
2	10	Indicó que el tablero no le parece la página principal
3	6	
4	15	No supo identificar los valores en los componentes.
5	14	Necesita que los componentes digan que son.
6	12	
7	8	
8	21	Se perdió un poco en el tablero
9	15	Los colores le parecen muy fuertes, no entiende algunos componentes.
10	7	

Cuadro 53: Tarea 3. Consumo en quetzales para el día de hoy

Usuario	**Tiempo(seg)**	**Observaciones**
1	6	No identificó la pestaña de hoy.
2	3	
3	5	
4	10	No identificó la pestaña de hoy.
5	7	
6	5	
7	8	
8	9	No identificó la pestaña de hoy.
9	4	
10	8	No identificó la pestaña de hoy.

Cuadro 54: Tarea 4. Entrar a configuración

Usuario	**Tiempo(seg)**	**Observaciones**
1	5	
2	6.5	
3	5	
4	18	No había visto que el usuario estaba en la barra superior.
5	7.5	
6	8	
7	14	No se dirigió rápidamente a la barra superior para buscar la opción
8	9	
9	15	
10	11	

Cuadro 55: Tarea 5. Consumo promedio de los miércoles

Usuario	**Tiempo(seg)**	**Observaciones**
1	12	
2	13	
3	10	
4	25	No supo a qué se refería la pregunta.
5	15	
6	8	
7	18	Le costó identificar cuál gráfica contenía la información.
8	19	Pensó que la gráfica en tiempo real contenía esta información.
9	9	
10	11	

Cuadro 56: Tarea 6. Consumo en tiempo real de esta semana

Usuario	**Tiempo(seg)**	**Observaciones**
1	7	
2	6.5	
3	5	
4	6	
5	8	
6	8	
7	10	
8	12	Trató de seguir buscando el dato, a pesar que ya lo tenía.
9	5	
10	15	No supo a qué se refería la pregunta

Cuadro 57: Tarea 7. Predicción de consumo mañana

Usuario	**Tiempo(seg)**	**Observaciones**
1	6	
2	5	
3	16	No sabía a qué se refería el término predicción
4	14	Se quedó analizando la razón de este dato.
5	18	Le causó duda porque debía de buscar este dato.
6	20	No sabía a qué se refería el término predicción
7	10	
8	4	
9	6	
10	15	No sabía a qué se refería el término predicción

Cuadro 58: Tarea 8. Consumo promedio a las 16:00 horas

Usuario	**Tiempo(seg)**	**Observaciones**
1	25	Problemas para saber a qué se referia las 16:00 horas
2	18	No identificó rápidamente el dato, a pesar de tenerlo enfrente.
3	9	
4	26	Problemas para saber a qué se referia las 16:00 horas
5	12	
6	12	
7	11	
8	25	Problemas para saber encontrar la gráfica que contenía este dato
9	12	
10	6	

Cuadro 59: Tarea 9. Salir de la aplicación

Usuario	**Tiempo(seg)**	**Observaciones**
1	5	
2	4	
3	4	
4	7	
5	4	
6	6	
7	8	
8	5	
9	4	
10	8	

2. Estudio de Usabilidad II

Cuadro 60: Tarea 1. Ingreso a la aplicación

Usuario	**Tiempo(seg)**	**Observaciones**
1	25	
2	26	Tuvo dificultades para posicionar el puntero, nuevamente.
3	14	
4	15	
5	19	
6	13	
7	18	
8	20	
9	27	
10	26	

Cuadro 61: Tarea 2. Encontrar consumo en kWh en el mes

Usuario	**Tiempo(seg)**	**Observaciones**
1	8	
2	10	
3	4	
4	6	
5	3	Le gustó que se incluyera un nombre a los componentes.
6	2.5	
7	7.5	
8	7	
9	4	
10	4	

Cuadro 62: Tarea 3. Consumo en quetzales para el día de hoy

Usuario	**Tiempo(seg)**	**Observaciones**
1	4	
2	3	
3	2.5	
4	5	No identificó la pestaña de hoy.
5	6	
6	8	
7	7	
8	3	No identificó la pestaña de hoy.
9	5.5	
10	6	

Cuadro 63: Tarea 4. Entrar a configuración

Usuario	**Tiempo(seg)**	**Observaciones**
1	6	
2	5.5	
3	3	
4	8	
5	7.5	
6	8	
7	11	
8	9	
9	18	
10	10	

Cuadro 64: Tarea 5. Consumo promedio de los miércoles

Usuario	**Tiempo(seg)**	**Observaciones**
1	9	
2	7	
3	9	
4	12	Dijo que no le encuentra sentido a este dato.
5	12	
6	7	
7	16	Tuvo que releer la pregunta para entenderla
8	18	Se distrajo un poco con la interfaz
9	7	
10	10	

Cuadro 65: Tarea 6. Consumo en tiempo real de esta semana

Usuario	**Tiempo(seg)**	**Observaciones**
1	8	No entendió la información de la gráfica
2	6.5	
3	5	
4	5	
5	7	
6	9	
7	11	
8	11	Trató de seguir buscando el dato, a pesar que ya lo tenía.
9	7	
10	7	

Cuadro 66: Tarea 7. Predicción de consumo mañana

Usuario	**Tiempo(seg)**	**Observaciones**
1	5	
2	4	
3	21	Hizo algunas preguntas sobre este dato.
4	9	
5	9.5	
6	14	
7	6	
8	4	
9	5	
10	11	

Cuadro 67: Tarea 8. Consumo promedio a las 16:00 horas

Usuario	**Tiempo(seg)**	**Observaciones**
1	10	
2	15	
3	18	Pregunt sobre la utilidad de este dato.
4	15	
5	13.5	
6	9	
7	8	
8	16	Tuvo que identificar la gráfica que contenía este dato
9	11	
10	7	

Cuadro 68: Tarea 9. Salir de la aplicación

Usuario	**Tiempo(seg)**	**Observaciones**
1	4	
2	5	
3	7.5	
4	6	
5	3.5	
6	4	
7	4	
8	5	
9	4	
10	9	

Figura 93: Datos generales. *Edad*

Figura 94: Datos generales. *Sexo*

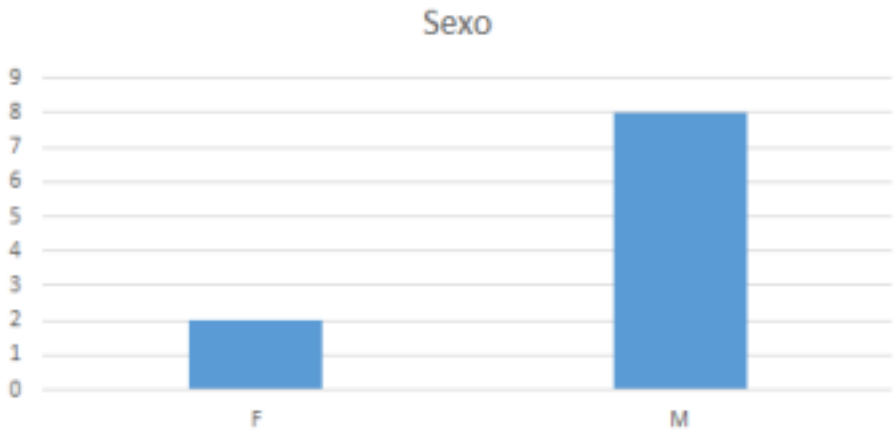


Figura 95: Datos generales. *Educación*

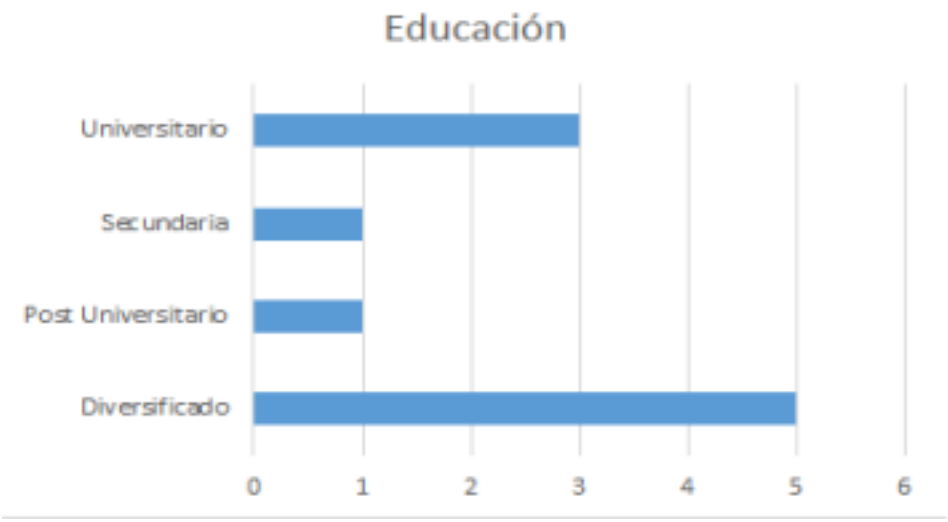


Figura 96: Datos generales. *Pregunta 1*

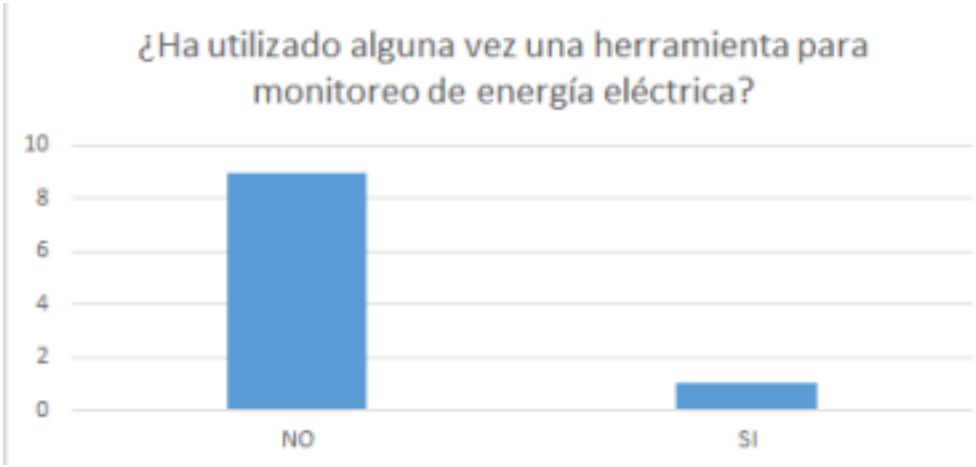


Figura 97: Datos generales. *Pregunta 2*

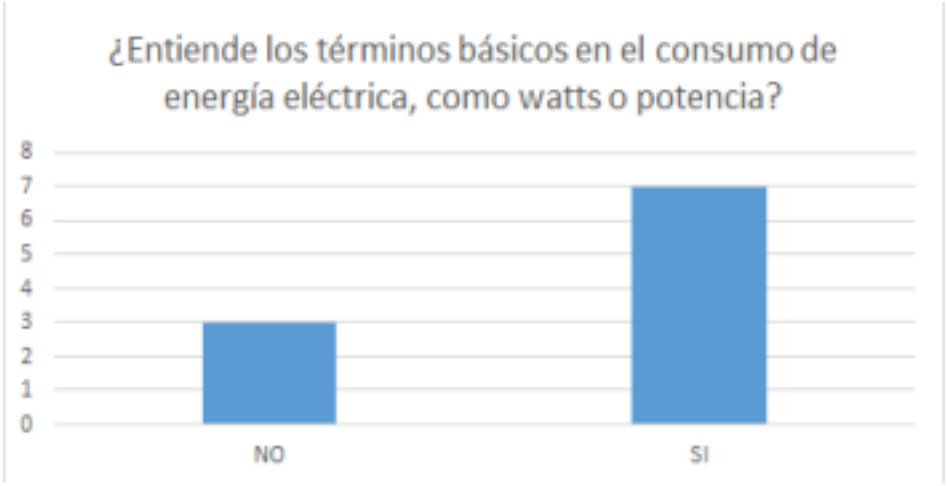


Figura 98: Datos generales. *Pregunta 3*

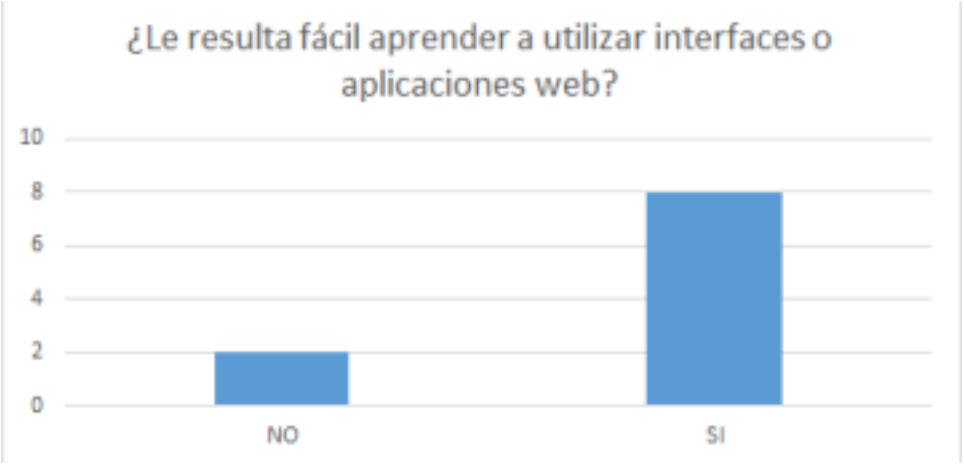


Figura 99: Datos generales. *Pregunta 4*

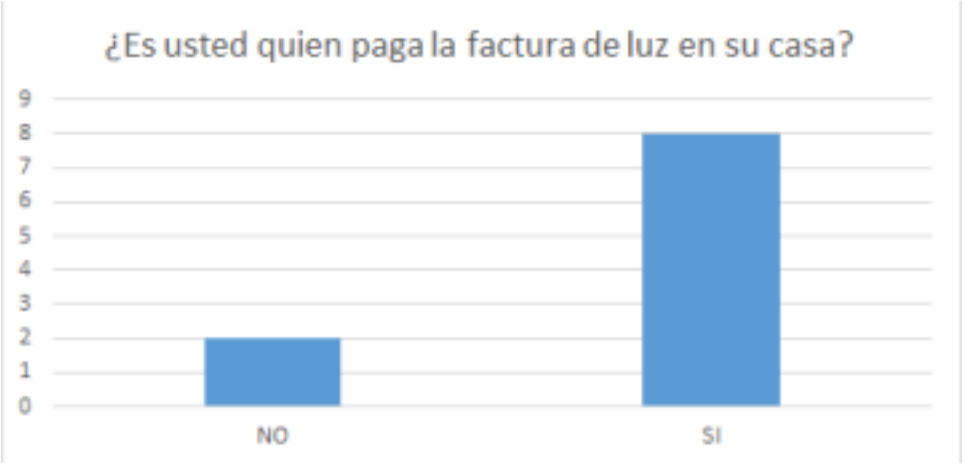
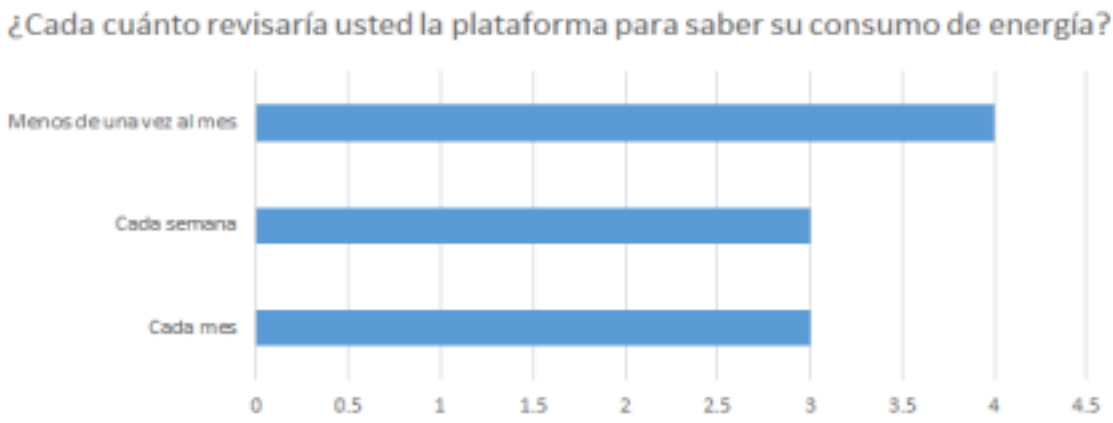


Figura 100: Datos generales. *Pregunta 5*



Cuadro 69: Tiempo promedio por tareas. Estudios de usabilidad I y II

Tarea	**Tiempo promedio (seg) Estudio I**	**Tiempo promedio (seg) Estudio II**
Ingreso a la aplicación	29.8	15.2
Encontrar consumo en kWh en el mes	11.9	5.6
Consumo en quetzales para el día de hoy	5.7	5
Entrar a configuración	9.9	8.6
Consumo promedio de los miércoles	13.8	10.7
Consumo en tiempo real de esta semana	8.25	7.65
Predicción de consumo mañana	11.4	8.85
Consumo promedio a las 16:00 horas	15.6	12.25
Salir de la aplicación	5.5	5.2

3. Funcionalidades de la interfaz de usuario La versión final de funcionalidades disponibles a través de la interfaz de usuario, quedó definida en la tabla 70.

Cuadro 70: Descripción de Componentes de la plataforma. (Web Design, 2012)

Sección	**Componente**	**Descripción**
Tablero	Dashboard o Tablero principal	Esta es la pantalla principal de la plataforma. Por defecto, muestra todos los componentes disponibles en las demás secciones. Sin embargo, se puede configurar para mostrar solo los que el usuario desee.
Consumo	Consumo Actual	Este es un componente central, pues muestra la cantidad de energía en kW/h consumida hasta el momento, en el mes, semana y día. Contiene una barra que indica cómo va el consumo según el límite que el usuario haya establecido en su configuración.
	Consumo Actual en Quetzales	Similar al componente anterior, este es un componente central, pues muestra la cantidad de dinero en quetzales que se debe pagar hasta el momento, en el mes, semana y día.
	Consumo en Tiempo Real	Esta es una gráfica de línea, que muestra el comportamiento del consumo por hora, día y semana. El eje y indica la cantidad consumida en kWh durante cada hora, día o semana.
Análisis	Consumo Promedio	Este componente tiene su parecido con el de Consumo Actual. Sin embargo, muestra la cantidad de energía en kW/h consumida en promedio, por mes, semana y día.
	Consumo Promedio en Quetzales	Este componente se parece al de Consumo Actual en Quetzales. Sin embargo, este muestra la cantidad de dinero en quetzales que se ha acumulado en promedio, por mes, semana y día.
	Consumo promedio por día de la semana	Este es un componente en gráfico de barras, que indica la cantidad de energía en kW/h consumida en promedio, según el día de la semana. Cada barra representa un día de la semana (lunes, martes,...) e indica el consumo promedio para ese día.
	Consumo promedio por hora del día	Este es un componente en gráfico de barras, que indica la cantidad de energía en kW/h consumida en promedio, según la hora del día. Cada barra representa una hora del día (11, 12,...) e indica el consumo promedio para esa hora.
Usuario	Ingreso a la aplicación	Esta es una pantalla de ingreso a la plataforma. Muestra una pantalla con campos para correo y contraseña, para que el usuario utilice sus credenciales para autenticarse y acceda únicamente a sus datos de consumo.
	Configuración de Usuario	Esta pantalla muestra los datos generales del usuario activo, y la configuración de los componentes que desea que aparezcan en el tablero.

4. Interfaz de Usuario La interfaz de usuario construida es de tipo web y está desarrollada en AngularJS en su versión 1.5. Las Figuras desde la 101 hasta la 111 muestran la interfaz de usuario finalizada después de los estudios de usabilidad.

Figura 101: Estudio de Usabilidad II. *Splash screen*

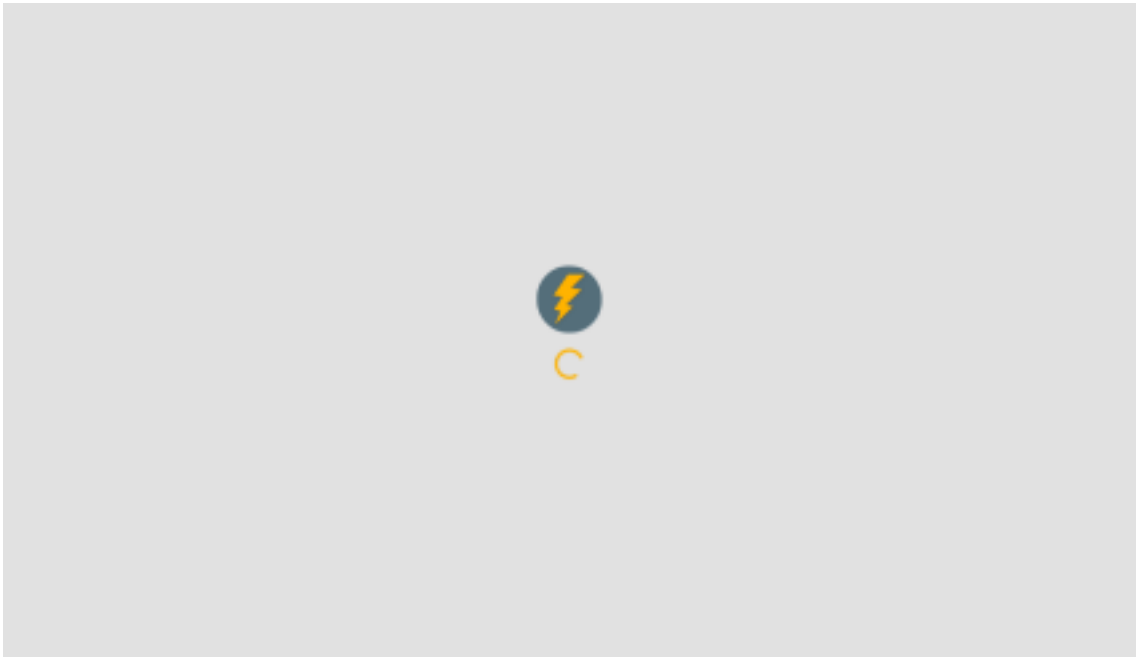


Figura 102: Estudio de Usabilidad II. Ingreso

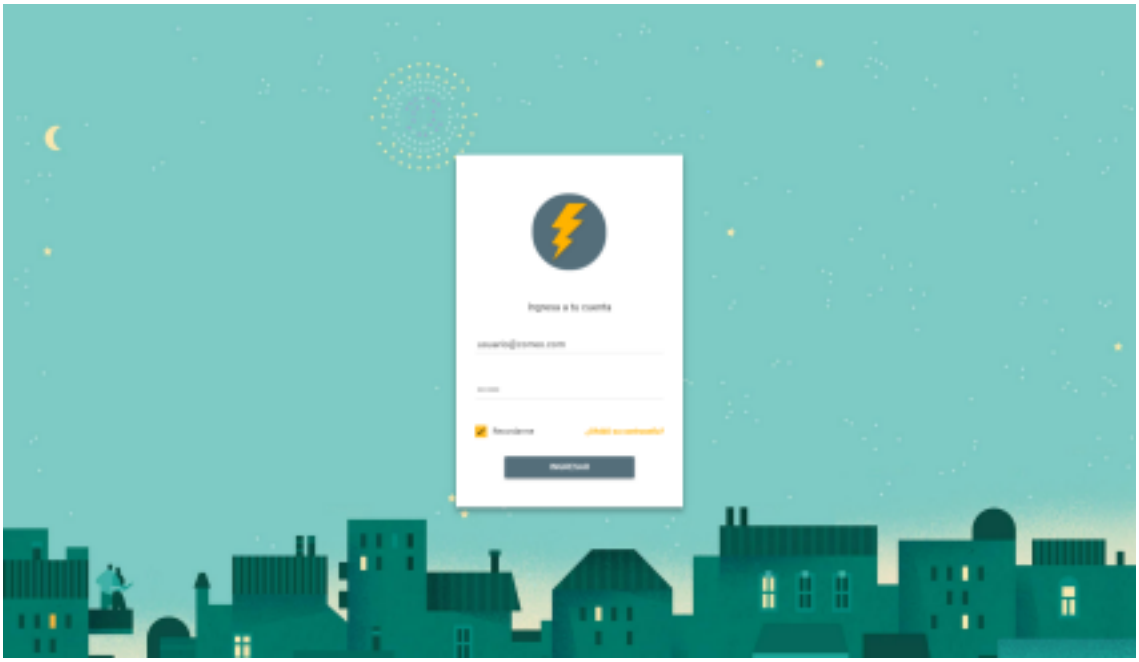


Figura 103: Estudio de Usabilidad II. Tablero principal

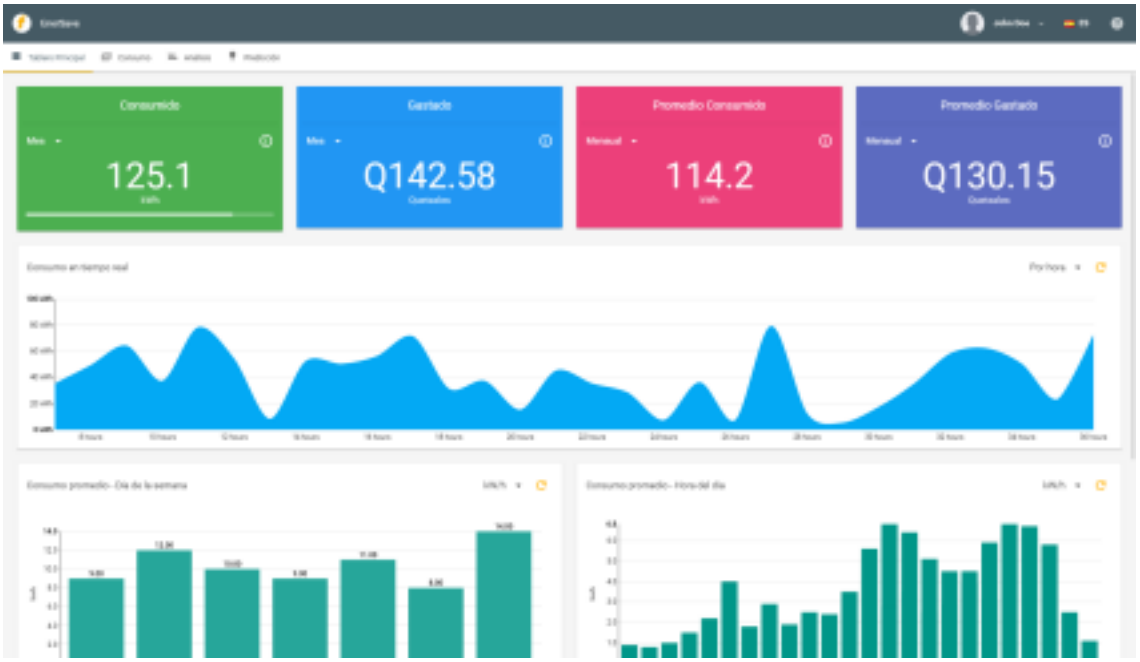


Figura 104: Estudio de Usabilidad II. Tablero principal 2

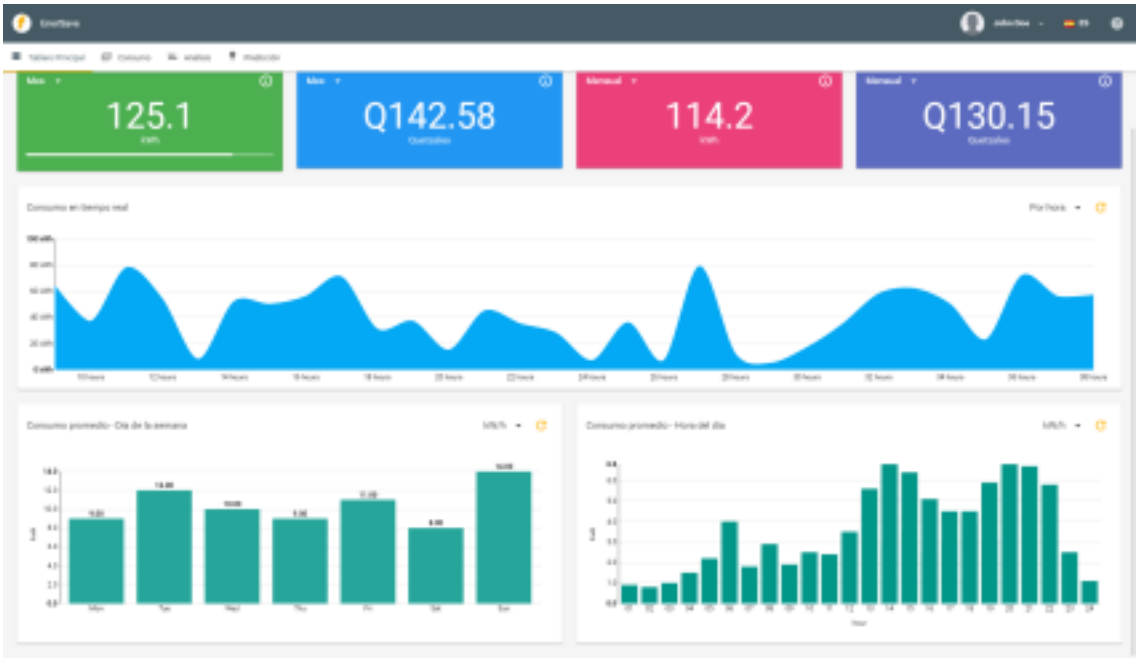


Figura 105: Estudio de Usabilidad II. Descripción del componente

Figura 106: Estudio de Usabilidad II. Consumo

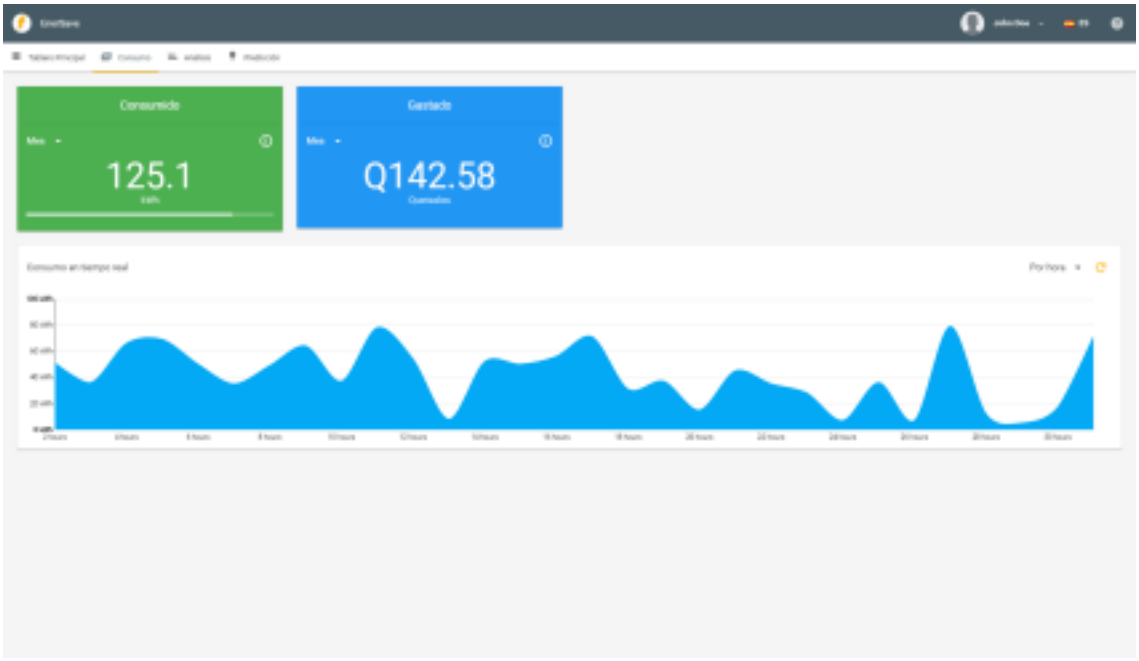


Figura 107: Estudio de Usabilidad II. Análisis

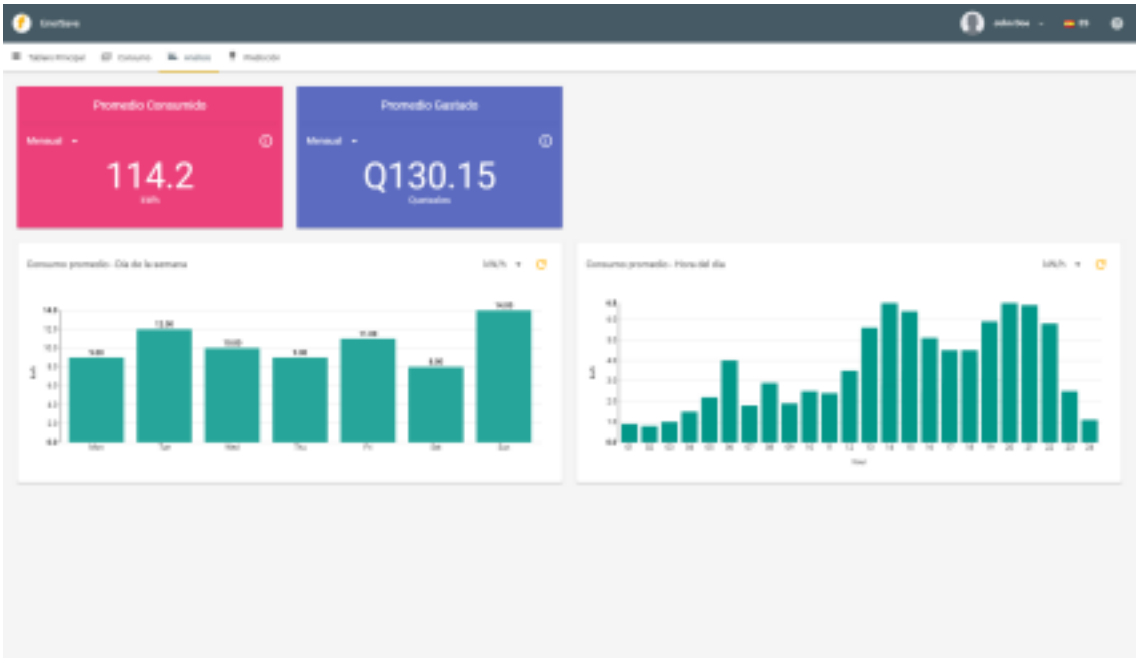


Figura 108: Estudio de Usabilidad II. Predicción

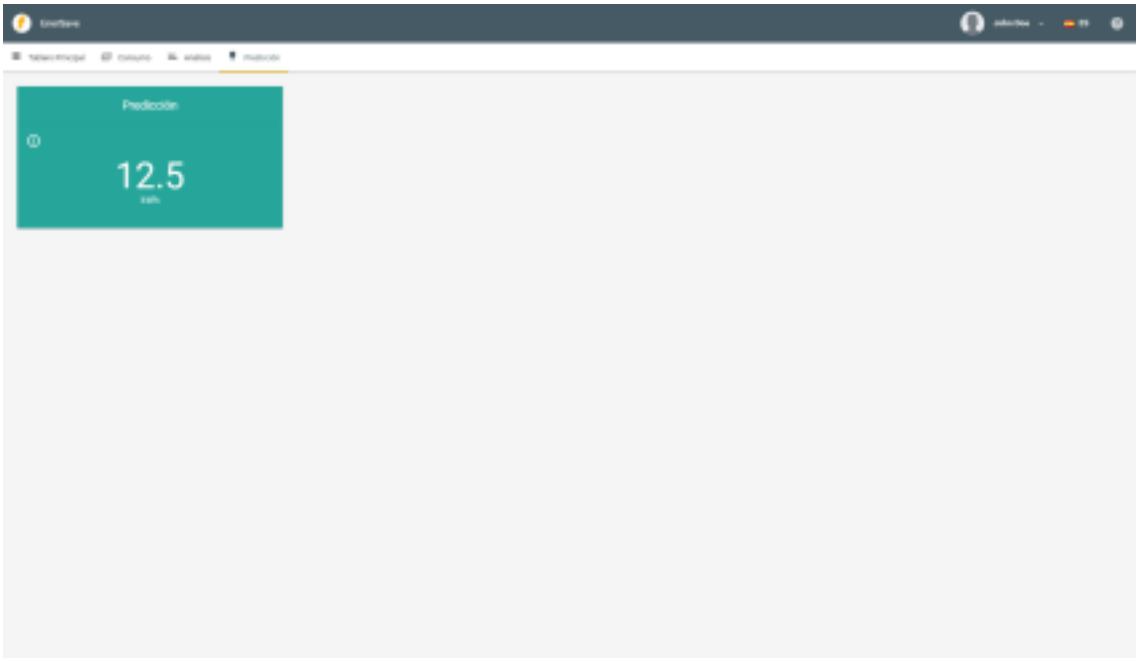


Figura 109: Estudio de Usabilidad II. Perfil de usuario

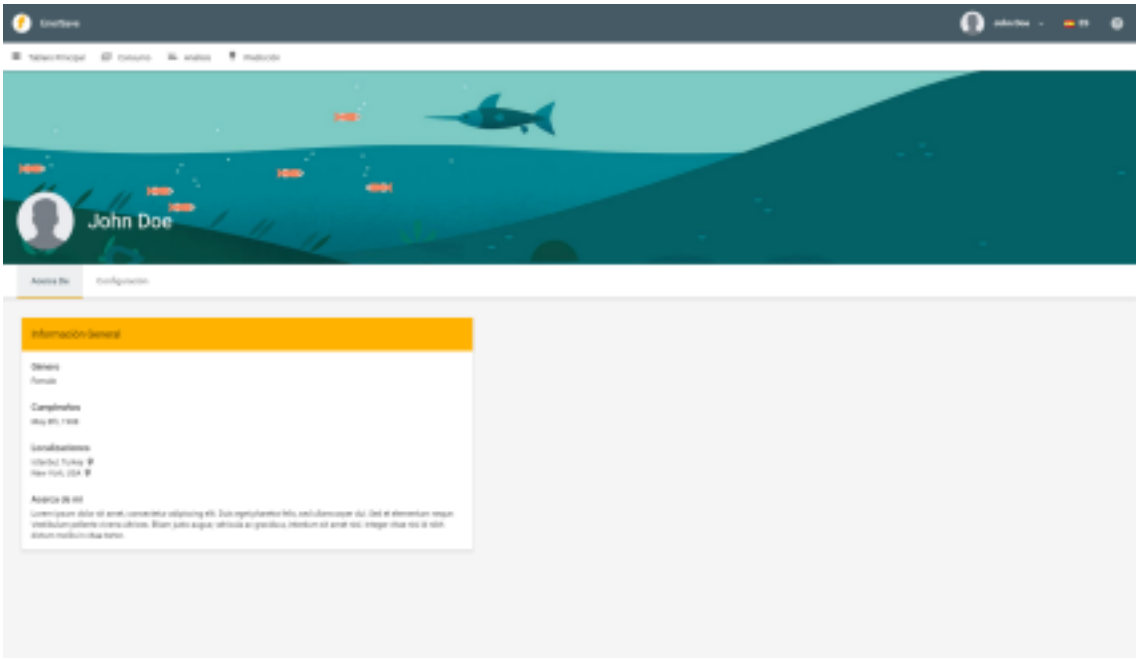


Figura 110: Estudio de Usabilidad II. Perfil de usuario, configuración

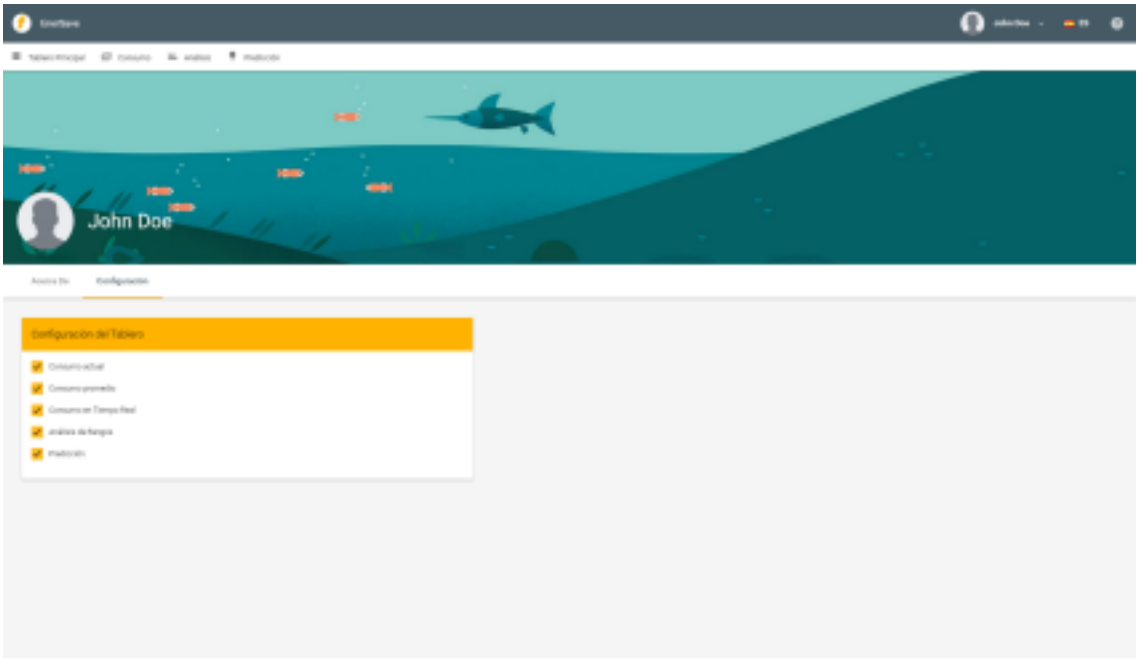
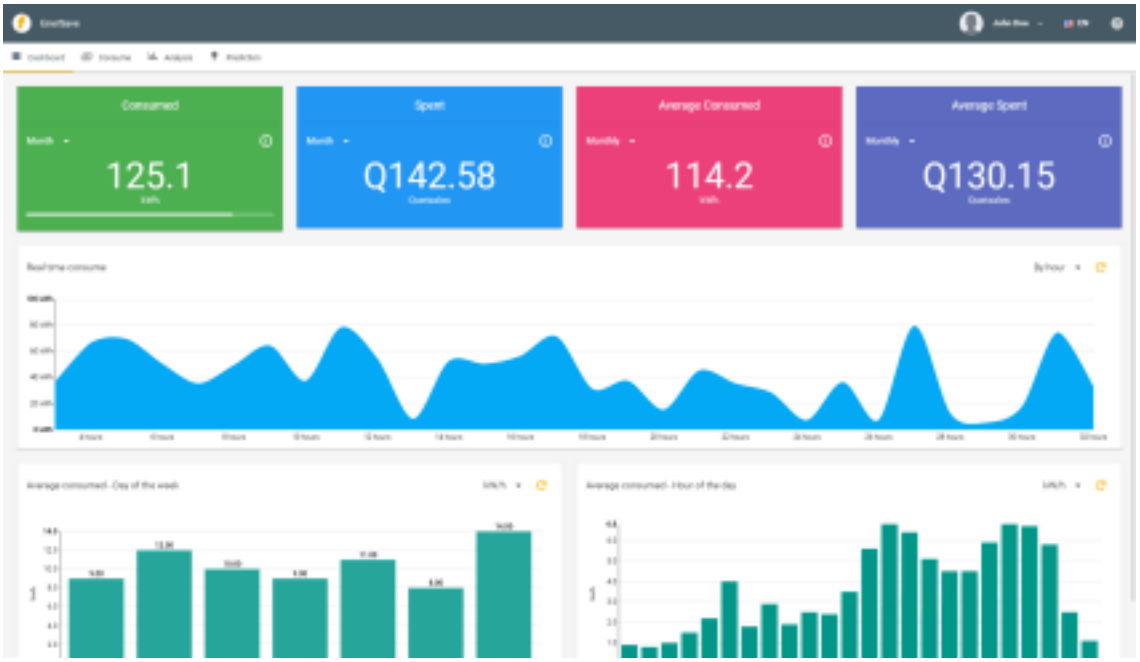


Figura 111: Estudio de Usabilidad II. Interfaz en inglés



VII. Análisis de resultados

A. Sensores y protocolos

1. Selección de sensor de voltaje Se puede observar en la Figura 16 que el sensor cumple con casi la totalidad de los requisitos anteriormente mencionados con la excepción de que su salida es un voltaje entre 1 y 5 V. Esto no ocasionó un problema puesto que el valor más pequeño que se desea medir es de 120 Vrms y el valor mínimo que detecta el sensor es de 100.

Como se observa en el Cuadro 23 las mediciones utilizadas para caracterizar el sensor de voltaje se utilizaron también para calcular el error. Esto fue debido a que esos valores son los estándar en tableros de distribución de electricidad. Del mismo cuadro se puede observar que el error más grande obtenido fue de 0.24.

2. Selección de sensor de corriente Se puede observar en la Figura 18 que el sensor es no invasivo pues realiza la lectura utilizando el campo magnético generado por la corriente que pasa por la línea. También se observa que es de fácil instalación pues basta con abrirlo y colocar el cable a través para luego cerrar y hacer que el núcleo se vuelva un circuito cerrado. A diferencda del sensor de voltaje, este sensor es únicamente un transformador de corriente, por lo que su salida no es un voltaje DC en el rango especificado. Para convertir la salida en un voltaje analógico que pudiera ser interpretado por el microcontrolador fue necesario colocar una resistencia de carga en la salida y agregar un circuito para acondicioiar la señal a un rango entre 0 y 5 V, dicho circuito se presenta en la sección de implementación. Al momento de elegir la resistencia de carga fue necesario tomar en consideración la ley de Ohm, puesto que entre mayor fuese la resistencia, mayor sería la potencia disipada. Por esta razón se escogió una resistencia de 33 Ω aunque eso implicara que al tener el valor más alto de corriente en la salida (50 mA figura 19) no se obtuviera 5 V.

Se puede observar en el Cuadro 24 que se tiene un error máximo de 1.34 %, valor menor al 3 % que indica el fabricante al referirse a el error de linealidad del sensor.

3. Protocolo de comunicación Como se observa en el Cuadro 25 se decidió transmitir 1 byte a la vez a una velocidad de 9600 bits por segundo, se omitió el bit de paridad puesto que la información a intercambiar era únicamente los resultados del módulo de conversión analógica a digital ADC, cuya salida es un valor digital de 8 bits.

4. Implementación del Módulo

a. Lectura de sensores. Para comprender la señal de salida que proveen los sensores fue necesario digitalizarlas, esto corresponde a la sección azul de la Figura 20. Se utilizó un microcontrolador PIC16F88 y el módulo de convertidor analógico a digital ADC incorporado para esta tarea. El microcontrolador se programó utilizando el compilador CCS en lenguaje C, configurado para utilicar el reloj interno a una velocidrd de 8 MHz. Se configuraron dos canales distintos del ADC, los canales AN2 y AN3 a una velocidad 32 veces más lenta que el reloj interno, cambiándose entre sí dependiendo del valor que se desea obtener. Debido a que la salida del sensor de voltaje es DC y va de 1 a 5 V no fue necesario acondicionar la señal, entonces se conectó directamente al canal AN3.

Como se puede observar en la Figura 21, al llamar a la función current_read() se obtienen 90 mediciones del canal de ADC de la señal del sensor de corriente, luego de tener todas las mediciones se obtiene el valor máximo. Este proceso se repite cuatro veces para un total de 360 mediciones y al final se devuelve el valor máximo de todas ellas, equivalente al valor pico de la señal. Conociendo el valor pico de la señal fue posible calcular el voltaje rms y, por ley de Ohm, la corriente rms. Con la corriente rms de salida se calculó la corriente rms de entrada utilizaudo una regla de tres.

Como se observa en la Figura 22 la corriente a la salida del sensor es independiente de la del sistema la señal se verá desfasada 2.5V debido al divisor de voltaje. Para evitar que la corriente AC fluyera por la resistencia R3 hacia tierra, alterando el voltaje de la medición, se colocó un capacitor de 10 μF en paralelo a R3. El amplificador operacional se colocó como seguidor de voltaje para asegurar que no existiera caída de tensión al conectar la señal al canal analógico del microcontrolador.

Luego de obtener la media de la diferencia obtenida para cada medición mostrada en el Cuadro 26 se encontró que, en promedio la medición realizada por el microcontrolador da un error de 0.1 V, este valor se utilizó como ajuste en el procesamiento de la información.

b. Comunicación inalámbrica. Luego de digitalizar la señal de los sensores fue necesario transmitirla de manera inalámbrica hacia el sistema de procesamiento y control central, esto corresponde a la sección verde de la Figura 20. Para esto se utilizó el módulo de radiofrecuencia nRF24L01+.

Para su correcto funcionamiento fue necesario cambiar la sección de la librería que hace referencia al hardware, puesto que la versión utilizada estaba destinada a otro modelo de microcontrolador. Los cambios realizados se muestran en la Figura 23.

Como se puede observar los únicos cambios realizados fueron debido a que las funciones de SPI se encuentran en el puerto B del microcontrolador, a diferencia de la ubicación en el puerto C de la librería original. Cabe mencionar que las funciones RF_CS y RF_CE pueden utilizarse con cualquier pin de entrada o salida general mientras que el resto de funciones deben corresponder a

los puertos especificados en el microcontrolador para la comunicación SPI.

En el circuito mostrado en la Figura 24, NET_18 y NET_8 corresponden a la distribución mostrada en las Figuras 25 y 26, respectivamente. Las líneas de 5 y 3.3 V se obtuvieron utilizando una fuente de 12V fija y reguladores de voltaje. Es importante mencionar que para la conexión del módulo RF al microcontrolador basta unir los pines con el mismo nombre, que el pin MISO debe llevar una resistencia de pull-up y que CE y CSN van conectados a las salidas definidas por el usuario.

Para el módulo receptor se cambió la función RF_CS del pin B5 al pin B6 del microcontrolador, debido a que el primero fue necesario para la comunicación vía UART con la Raspberry Pi 2. Tanto el microcontrolador como el módulo de radiofrecuencia reciben la fuente de alimentación de la Rasberry Pi 2. Esto se puede observar en la Figura 27.

El ciclo del módulo receptor depende del funcionamiento del emisor, puesto que sólo envía información a la Raspberry vía UART si recibe información inalámbricamente, de lo contrario se mantiene esperando a que esto suceda.

c. Procesamiento de la información y almacenamiento en el servidor .

Se puede observar en la Figura 28 que la conversión puede ocurrir únicamente en un sentido, pero dado que la Raspberry no va a transmitir información hacia el microcontrolador esto no representó un problema para la implementación.

Tanto en la ecuación VI..1 como en la ecuación VI..2 se observa que se realiza el ajuste de 0.1 V, debido a la desviación observada durante la toma de mediciones con el microcontrolador. Para deducir estas ecuaciones se despejó la caracterización de cada sensor puesto que en ellas la variable independiente es el valor real RMS. Debido a que en la ecuaciones se debe utilizar un valor de voltaje se realiza una multiplicación utilizando como referencia el rango del ADC para convertir de valor digital a voltaje. Luego de esto ya se utiliza el resto de la ecuación despejada para obtener los valores reales. Como estos valores ya son rms la potencia se calculó multiplicando ambos.

Debido a que el servidor no soporta el almacenamiento de datos cada segundo, la Raspberry almacenó los datos por medio minuto y luego de eso se realizó un promedio del voltaje, la corriente y potencia y eso se almacenó en el servidor. De esta forma se obtuvo un almacenaminnto de datos cada 30 segundos pues se adquiere información de los sensores cada segundo.

Como se buscó calcular la energía consumida y la frecuencia de muestreo fue de 1 segundo entonces se multiplicó la potencia por el equivalente a horas de 1 segundo. Esta operación dio como resultado el consumo de energía, en Watts hora, de 1 segundo, entoncrs bastó con sumar los consumos de energía por segundo para obtener el consumo total del hogar.

En la Figura 31 se muestran las primeras mediciones con el sensor de voltaje. Para la realización de esta gráfica se realizó un ajuste de 6.8 Vrms debido a sue los valores se encontraban centrados alrededor de 126.8Vrms, al hacer esto se pudo apreciar mejor las variaciones y su magnitud puesto

que el valor esperado era de alrededor de 120 Vrms. Cabe mencionar que este ajuste se realizó únicamente en la gráfica y no al momento de calcular la potencia. Se puede observar que los valores obtenidos variaron alrededor de un punto medio, estas variaciones son mucho mayores que las obtenidas durante las mediciones de prueba. Esto puede ser debido a que en la conexión de la red eléctrica de un hogar existe una mayor cantidad de ruido. Las mediciones de prueba se realizaron en un ambiente controlado donde el voltaje es regulado y adicionalmente contaban con supresores de picos, los cuales disminuyen el efecto del ruido. Las variaciones fueron de aproximadamente ± 2.5 Vrms.

Las primeras mediciones con el sensor de corriemte se muestran en la gráfica de la Figura 32. Se puede observar que al inicio y durante aproximadamente la mitad del tiempo de medición se tuvo el consumo de corriente mínimo, de alrededor de 5 amperios rms. En el resto de muestros se observan los picos de aumento en la corriente, algunos llegaron hasta los 36 amperios rms, pero rara vez se mantienen constantes durante un período prolongado de tiempo. Tomando en cuenta que las mediciones iniciaron a las 10 a.m. se puede observar que hubo picos de corriente alrededor de la 1 p.m. El valor máximo de corriente se registró luego de transcurridas doce horas y media de mediciones con un pico que llegó hasta los 36 amperios rms a las 10:30 p.m., esta pudo ser debido a que un miembro de la familia del hogar utiliza un dispositivo para administración de oxígeno en las noches. Luego de eso se observa que la corriente disminuyó hasta un valor mínimo de 5 amperios rms y se mantuvo así hasta aproximadamente las 5 a.m. del día siguiente cuando se registró nuevamente un pico de 36 amperios rms. Luego de esto se observan variaciones pero de magnitud pequeña y aproximadamente a las 7 a.m. se registró un pico de 33 amperios rms. Estos valores altos de corriente en la mañana pueden ser debido a la utilización del calentador de agua.

Para calcular la potencia consumida por el hogar se utilizó la ecuación IV..2, la gráfica obtenida de esos resultados se muestra en la Figura 33. Se puede observar que la gráfica de potencia posee el mismo comportamiento que la gráfica de corriente. Esto se debe a que las variaciones en el voltaje son pequeñas en comparación con las variacioaes en la corriente, debido a esto la potencia consumida por el hogar se vio afectada principalmente por la magnitud de la corriente. A pesar de que se tuvo aue ajustar la medición de voltaje, este ajuste fue un valor constante, por lo que no afectó el comportamiento de la potencia consumida, únicamente su magnitud.

B. Seguridad de la información

1. Selección de algoritmo criptográfico Entre todos los algoritmos listados dentro del marco teórico; entiendase los algoritmos: Rijndael, DES, 3DES, RC2, RC4, RC5, RC6, Blowfish, Twofish, IDEA, MARS, Serpent y CAST. Se seleccionó solo uno para su debida implementación dentro del proyecto. Esta selección involucró distintos criterios para comparación; según su arquitectura, seguridad, ataques conocidos, uso de la comunidad y eficiencia en operaciones según análisis de tiempo y memoria consumida según resultados de la implementación.

a. Arquitectura del algoritmo Esta sección es la comparativa entre la estructura y las operaciones que utiliza el algoritmo. Se determinó si el algoritmo sería de tipo simétrico o asimétrico. Con base en el nivel de criticidad, se dividió la información en información perteneciente a la aplicación y en las llaves criptográficas utilizadas en el cifrado simétrico.

- ***Cifrado y descifrado de la información***: Para el tema de arquitectura, se comparó en elegir por cifrado simétrico o cifrado asimétrico. Según la investigación realizada, se obtuvieron distintas características positivas y negativas, sobre cada uno de estos cifrados.
  - *Cifrado simétrico:*
    - Características positivas
      - Rápido
      - Eficiente
      - Seguro
      - Procesamiento poco costoso
    - Características negativas
      - Intercambio y administración de llave muy riesgosa
  - *Cifrado asimétrico*
    - Características positivas
      - Existencia de dos llaves (pública y privada)
      - Muy seguro
      - Brinda el no repudio
    - Características negativas
      - Lento
      - Procesamiento intensivo

  Con base en la modalidad del proyecto y su uso; se pudo elegir para el cifrado y descifrado de la información propia de la aplicación, emplear por un algoritmo de cifrado simétrico. Se

eligió debido a sus características, además en base a los resultados; se puede determinar que un algoritmo criptográfico simétrico es más rápido que un algoritmo criptográfico asimétrico.

- **Se necesita rapidez para brindar resultados a los usuarios**. Según este algoritmo, es mucho más rápido a comparación a un algoritmo asimétrico, según las gráficas en la sección de resultados en implementación de algoritmos simétricos e implementación de algoritmos asimétricos en tiempo y memoria.

- **Se necesita eficiencia para mejorar el procesamiento de los dispositivos**. Se sacrificó más seguridad por eficiencia, debido al nivel de criticidad de los datos. Al ser un algoritmo simétrico, el procesamiento computacional es muy barato y no consume muchos recursos del sistema. También se tomó en cuenta, si se planea implementar los dispositivos dentro de las casas es necesaria la eficiencia para nivelar el procesamiento en el servidor.

- **No se necesita seguridad extrema**. Según la información que se enviará dentro del canal inseguro; este no tendrá datos extremadamente sensibles al usuario, por lo que no es necesario un algoritmo altamente seguro como el asimétrico.

- ***Intercambio y administración de llaves criptográficas*** Se realizó la separación con el intercambio y administración de las llaves criptográficas simétricas con el cifrado de la información perteneciente a la aplicación, debido a su nivel de importancia y criticidad dentro del proyecto. Para el manejo de llaves, se intentó evitar una mala gestión de llaves criptográficas utilizadas dentro de un canal inseguro físico. Se evitará también, el intercambio por medio de dispositivos electrónicos (USB, papel, CD, etc) y el intercambio por medio de un canal inseguro dentro del internet.

  Se utilizó el algoritmo de cifrado Asimétrico, debido a:

  - **Se cifrará información extremadamente crítica**. Un atacante, al obtener esta llave criptográfica simétrica, fácilmente puede descifrar o alterar la información que es transmitida por un canal inseguro dentro de la red. Es necesario cifrarla con un algoritmo de seguridad fuerte.

  - **Creación de dos llaves**. Este algoritmo, puede transmitir la llave pública generada, a través de un canal inseguro, sin comprometer la aplicación. El actor que reciba esta

llave, podrá cifrar la información (llave utilizada para el cifrado simétrico), y transmite nuevamente el texto cifrado; para que el que posea la llave privada pueda descifrarlo, brindando así integridad al sistema.

- **El procesamiento intensivo de cifrar la llave simétrica de 32 bytes no afectará al sistema**. La llave criptográfica simétrica que se cifra con este algoritmo no tendrá un gran impacto al procesamiento de los dispositivos, debido a que es información es relativamente pequeña; 32 bytes o 32 caractéres, y se realizará únicamente en el momento de instalación y registro de la aplicación.

b. Seguridad de los algoritmos Esta sección se comparó la seguridad de los algoritmos de cifrado asimétrico y simétrico. Para los cifrados simétricos, se tomó como criterio de comparación, la cantidad de posibilidad de llaves por algoritmo, el tamaño de bloque, los ataques actuales para los algoritmos y el nivel de seguridad para cada uno basado en cantidad de rondas comprometidas. La seguridad fue catalogada como segura y no segura. Para los algoritmos asimétricos, el criterio de comparación fue el tamaño de llave, y la complejidad de la implementación; debido a que estos algoritmos se catalogan como muy seguros hasta la actualidad.

- ***Algoritmos simétricos***: Existen cuatro para seleccionar el algoritmo. El primer criterio fue seleccionar el algoritmo criptográfico que posea una llave criptográfica de tamaño de 256 bits. Mientras más grande la llave criptográfica, más posibilidades de llaves existen. El tamaño de la llave está definido en bits, lo que significa que es una cadena de caracteres de 0's y 1's; por lo que existen dos posibilidades para el tamaño de la llave criptográfica, por ejemplo una llave de 256 bits tiene 2^{256} combinaciones de llave. Además según NIST, la empresa de estándares norte americana, mencionan que una llave de 256 bits provee un nivel alto de seguridad.

  El segundo criterio para selección, es el tamaño de bloque como mínimo de 128 bits. Mientras más grande es el tamaño del bloque, es más seguro; debido a que se encontrarán menos probabilidades que los textos cifrados aparezcan repetidos al momento de transformar el texto original a texto cifrado. Además según el reporte de los algoritmos, elaborado por la empresa europea ENISA; mencionan que el tamaño de bloque debe de ser como mínimo de 128bits. (Smart *et al.*, 2014).

  El tercer criterio fue en verificar la cantidad de rondas atacadadas por algoritmo. Según la documentación en el Cuadro 3 dentro de la sección del marco teórico; este muestra algunos

de los ataques que podrían significar un riesgo a los algoritmos y la cantidad de rondas actualmente registradas que han sido atacadas. Se eligieron específicamente los algoritmos que no han sido atacados en su totalidad, donde todas las rondas han sido comprometidas.

Como último criterio, se realizó la comparación de los algoritmos que fueron candidatos dentro del concurso realizado por NIST (National Institute of Standards and Technology), según el Cuadro 4 en la sección de marco teórico. Se eligió tres de los algoritmos para su debida implementación y análisis.

- De los algoritmos investigados, los siguientes poseen llaves iguales a 256 bits, según el Cuadro 2 en la sección del marco teórico.
  - Rijndael
  - RC2
  - RC4
  - RC5
  - RC6
  - CAST
  - MARS
  - Blowfish
  - Twofish
  - Serpent
- De los algoritmos investigados que poseen llave criptográfica de 256 bits o mayor, se seleccionan los algoritmos con bloques de tamaño mayores o iguales a 128 bits, según el Cuadro 2 en la sección de marco teórico.
  - Rijndael
  - RC4 (Flujo)
  - RC6
  - CAST
  - MARS
  - Twofish
  - Serpent
- Con los algoritmos con 256 bits de tamaño de llave y 128 bits de tamaño de bloque, se selecciona los algoritmos que no han tenido ataques exitosos o la cantidad de rondas comprometidas no es completa, según el Cuadro 3 en la sección de marco teórico.

    - Rijndael
    - RC6
    - CAST
    - MARS
    - Twofish
    - Serpent
  - Con base en las selecciones anteriores, los seis algoritmos filtrados han sido los mismos algoritmos que compitieron dentro del concurso para convertirse en el nuevo AES (Advanced Encryption Standard). Se seleccionaron los cinco finalistas de la segunda ronda, según el Cuadro 4 en la sección de marco teórico.
    - Rijndael
    - RC6
    - Twofish
    - MARS
    - Serpent
  - Para los cinco algoritmos filtrados, se tomaron los tres mejores votados según la terna de la competencia para convertirse al nuevo AES. Estos algoritmos son, con base en el Cuadro 4 en la sección de marco teórico.
    - Rijndael
    - Twofish
    - Serpent
  - Utilizando los algoritmos anteriores, se realizaron pruebas de funcionalidad según las especificaciones del proyecto; probandolas dentro del ambiente de trabajo de una Raspberry PI. Se obtuvo la comparación entre la cantidad de memoria y el tiempo que utiliza cada una de sus funciones (cifrar, descifrar y generar llave criptográfica) para los bloques de información de entrada.

- ***Algoritmos asimétricos***: Para los algoritmos asimétricos, se realizó una comparación parecida a los simétricos; dependiendo del tamaño de llave que puede soportar el algoritmo y la complejidad de su implementación, según se muestra en el Cuadro 5 en la sección de marco teórico, dejando los siguientes algoritmos para su implementación.
  - RSA
  - ElGamal

Se eligieron estos algoritmos, debido al tamaño de llaves criptográficas que soportan son mayores de 1024, con 2048 bits para realización de pruebas de los dos algoritmos. Un factor importante de sus elecciónes es la fácil implementación dentro de un nuevo sistema, a comparación del algoritmo criptográfico ECC, debido a que si la implementación es muy compleja y difícil, está más propenso a agregarle errores y vulnerabilidades al sistema.

2. Implementación de los algoritmos Con los distintos algoritmos elegidos del cifrado simétrico y asimétrico; se inició la parte de implementación de cada uno de ellos. Se empezó implementando desde el lenguaje de programación Python. Este lenguaje de programación, al ser interpretado, el manejo de información es relativamente fácil, además de la eficiencia manejando grandes bloques de información para cifrar y descifrar. Python además, es un lenguaje muy común, por lo que existe una gran diversidad de librerías de uso libre.

Se utilizaron algoritmos criptográficos ofrecidos por la librería llamada PyCrypto y PyCrypto-Plus. Estas librerías brindan distintos algoritmos criptográficos simétricos y asimétricos comúnes. Para poder analizar la cantidad de memoria utilizada por los algoritmos; se utilizó una librería llamada *memory_profiler*, y para analizar los tiempos, se realizó una diferencia de tiempos iniciales con finales, al momento de realizar el cifrado/descifrado. Se analizó variando la cantidad de texto para cifrar/descifrar; esta aumentandose en múltiplos de 10 bytes.

a. Algoritmos simétricos : Se analizaron los tiempos y la cantidad de recursos computacionales, entiendase como la cantidad de memoria que utilizaron los tres algoritmos destacados: Rijndael, Serpent y Twofish para la generación de llave criptográfica, el cifrado y descifrado de los datos. Estas pruebas se realizaron dentro de una Raspberry para verificar la habilidad de estos algoritmos en ese ambiente de trabajo. Se ejecutó el análisis de tiempo y recursos para diferentes cantidades de datos de ingreso para determinar cómo reaccionará el algoritmo ante una diferencia de tamaño de datos de entrada. Estas cantidades fueron: 1, 10, 100, 1000, 10000 y 100000 bytes.

Para la cantidad de tiempo utilizada sobre la generación de llave criptográfica simétrica, según la información del Cuadro 27 y la Figura 34 en la sección de resultados, se puede apreciar que el algoritmo más rapido es Twofish con 40.5ms, precediéndolo Rijndael con 3.0ms y finalmente Serpent con 1.5ms.

Twofish fue el algoritmo más tardado para generar el valor de la llave criptográfica simétrica. Este algoritmo tiene la característica que el cifrado y descifrado se realiza a la misma velocidad; pero la generación de llave es mucho más tardada y compleja en base a su arquitectura y cálculos que realiza. Este algoritmo, a diferencia de Rijndael y Twofish, trabaja de forma independiente

al tamaño de llave; por lo que cifra, descifra y genera la llave en tiempos independientes. Para los otros dos algoritmos Rijndael y Twofish, se puede apreciar que la generación de llave es relativamente rápida, con una diferencia mínima de 1.5ms. Se logró confirmar el orden de velocidad de los algoritmos según la documentación previa, en el Cuadro 4 dentro de la sección marco teórico.

Luego para el análisis de tiempo de descifrado de los algoritmos, según la información del Cuadro 29 y la Figura 37 en la sección de resultados, se puede mostrar que el algoritmo más rápido es Twofish con 1182.66ms, por consiguiente Rijndael con 1205.83ms y finalmente Serpent, siendo el más lento por una diferencia significativa, con 386210.5ms.

Serpent, como se puede visualizar en la Figura 35, es el algoritmo más lento de los implementados. Tiene una diferencia de 385027.84ms siendo este aproximadamente 6 minutos con 41 segundos más lento a comparación a Twofish. La diferencia de tiempos es notable debido a que este algoritmo utiliza más rondas de cifrado y descifrado internamente, enfocandolo más en seguridad que en velocidad. Con base en esto, este algoritmo es más tardado al momento de cifrar y descifrar una cantidad grande de datos, en comparación a Twofish y Rijndael según el Cuadro 4 en la sección del marco teórico. Además según Joan Daemen y Vincent Rijmen mencionan que Serpent ha sido uno de los algoritmos más problematicos conforme a la lentitud de sus operaciones (Daemen y Rijmen, 2001). Twofish y Rijndael por otra parte, según la Figura 38, son algoritmos más eficientes que Serpent, tuvieron una diferencia significativa de solo 100ms, siendo Twofish el más rápido. Rijndael quedó en la segunda posición debido a que el algoritmo es dependiente de la llave criptográfica; su cifrado y descifrado es más lento que Twofish.

Por último, para el análisis de tiempo de cifrado, según la información del Cuadro 28 y la Figura 35 de la sección de resultados, se puede apreciar que Twofish es el algoritmo más rápido con 1188.0ms, seguido de Rijndael con 1206.33ms y finalmente Serpent con 415775.5ms.

Serpent nuevamente en su tiempo de cifrado, fue uno de los más lentos teniendo una diferencia con Twofish de 414587.5ms siendo este 6.9 minutos aproximadamente. El tiempo de cifrado y descifrado para este algoritmo es muy extenso debido a la cantidad de operaciones que realiza al poseer más criterios de seguridad que Twofish y Rijndael. Twofish y Rijndael a diferencia de Serpent, según la Figura 36, mostró ser de los más rápidos al momento de cifrar la cantidad de datos, teniendo una diferencia no significativa de 18.33ms.

Después de haber elaborado el análisis de tiempo sobre las diferentes fases que realizan los algoritmos criptográficos; se realizó el análisis de memoria consumida para estas mismas fases con los mismos algoritmos.

Para el análisis de memoria al momento de generar la llave simétrica, según la información del Cuadro 30 y la Figura 39 en la sección de resultados, se puede apreciar que el algoritmo que consumió más memoria, es el algoritmo de Serpent con 91.8255mb, precediéndolo Rijndael con 91.8266mb y finalmente Twofish con 91.8287mb.

De igual manera, para el análisis de memoria al momento de descifrar, según la información del Cuadro 32 y la Figura 41 en la sección de resultados, Serpent fue el algoritmo que consumió más memoria con 91.8255mb, precediéndolo Rijndael con 91.8261mb y finalmente Twofish con 91.8274mb.

Por último, el análisis de memoria al momento de cifrar, según la información del Cuadro 31 y la Figura 40 en la sección de resultados, se puede apreciar que el algoritmo que consumió más memoria, es Serpent con 91.7451mb, precediéndolo Rijndael con 91.8255mb y finalmente Twofish con 91.8274mb.

En los tres distintos análisis de consumo de memoria que se realizaron para los algoritmos criptográficos simétricos, se puede determinar que la cantidad de memoria no varía de una manera muy significativa. En los tres casos analizados, Serpent ha sido el algoritmo que ha requerido menos consumo de memoria. Este algoritmo, a pesar de ser uno de los algoritmos más lentos en cifrado/descifrado, su consumo de recursos es más pequeño, variando por apenas 0.12mb con su sucesor Rijndael. Con base en este análisis, se puede concluir que los tres algoritmos sí pueden ser implementados dentro de un ambiente de trabajo que posea pocos recursos computacionales como la Raspberry sin sufrir consecuencias de procesamiento intenso en sus operaciones. Ahora, conforme al análisis de tiempo, la diferencia de tiempos para Twofish y Rijndael varían muy poco, menos de 100 milisegundos. Serpent por otra parte ha tenido una diferencia muy significativa. Se descartará a Serpent para la implementación, debido a que se necesita más eficienca para el cifrado de información, tal y como brinda Twofish o Rijndael.

Con base en este análisis de tiempo y recursos consumidos de los algoritmos simétricos, finalmente se eligió a AES Rijndael sobre Twofish y Serpent, como algoritmo para implementar dentro del proyecto. Se eligió este algoritmo debido a que a pesar de quedar en segundo lugar dentro del análisis de tiempo, y estar de igual manera en segundo lugar con un uso de recursos computacionales, este algoritmo ha sido uno de los más utilizados dentro del mercado desde años atrás. Diversas empresas tecnológicas que ofrecen servicios muy similares y entidades gubernamentales, utilizan este algoritmo para el cifrado de los datos debido a los márgenes de seguridad que posee y la eficiencia en sus operaciones. Actualmente, se han registrado varios ataques contra este algoritmo,

el cuál no han logrado comprometer hasta la fecha. Rijndael además ofrece mucha documentación de uso e implementación a nivel de desarrollo, aparte que existe una multitud de librerías de uso libre dentro del internet.

AES Rijndael se implementó con las siguientes características: llave criptográfica de 256 bits, tamaño de bloque de 128 bits y modo de operación CTR. Se utilizó este modo de operación debido a la eficiencia en el software y hardware; además que la salida de texto cifrado no es la misma para el mismo texto de entrada, como lo realiza el modo usual ECB (Electronic Code Book).

b. Algoritmos asimétricos : Para estos algoritmos, se realizó el análisis conforme a dos criterios; los tiempos que utilizan los algoritmos para el cifrado, descifrado y generación de las llaves criptográfica (pública y privada). Segundo se analizó la cantidad de memoria que utiliza para el cifrado, descifrado y generación de llave. Finalmente se escogió el que posea las mejores características; la eficiencia en tiempo y recursos para los tres diferentes aspectos utilizando un tamaño de llave de 2048bits.

Con base en los resultados de la Figura 42, se puede apreciar que el tiempo de generación de llaves públicas y privadas para los algoritmos, RSA se clasificó como el más eficiente a comparación de ElGamal, con una diferencia de 7762550ms, en minutos 129.37min y en horas 2.15hrs. Esta diferencia de tiempos es muy drástica y se puede comprobar conforme a los procesos que realizan los distintos algoritmos para generar las llaves. Para ElGamal, el proceso de generar la llave criptográfica se basa en seleccionar un número primo aleatorio p basado en la operación de $(k-1)$ donde k es el tamaño de la llave criptográfica y comprobar hasta que la operación $2q+1$ sea un número primo. Al ser un tamaño de llave criptográfica de 2048, existen millones de posibilidades de que el valor de la operación $2q+1$ sea un número primo; por lo que el algoritmo estará buscando entre todas estas posibilidades, generando así un tiempo de busqueda muy excesivo. A diferencia del algoritmo criptográfico RSA, este genera las llaves únicamente buscando aleatoriamente dos números primos relativamente grandes, dentro del rango del tamaño de llave; por lo que la busqueda se reduce de gran manera.

En el análisis de tiempo de cifrado, según la Figura 43 en la sección de resultados, se puede visualizar que los tiempos para los dos algoritmos fue exáctamente el mismo, siendo este 2ms; un tiempo relativamente rápido.

Sobre el resultado de tiempo de descifrado que se muestra en la Figura 43 en la sección resultados, se puede visualizar que los tiempos de los dos algoritmos varían por 81ms, encabezando

el algoritmo RSA y proseguido de ElGamal. Estos tiempos están conforme a las operaciones que realizan los algoritmos. Para RSA, el proceso de descifrado se basa principalmente en realizar una operación exponencial y módulo. Mientras que para ElGamal su operación de descifrado se basa en dividir el mensaje en grupos de bits, computar la llave compartida y en base a esa llave compartida generar el texto descifrado por pedazos, para después unirlo y obtener el texto descifrado. RSA al tener menos operaciones, el tiempo de ejecución es menor.

Para el análisis de memoria para la generación de llave mostrada en la Figura 45 dentro de la sección de resultados, se destaca que RSA quedó primero con 57.68mb y procediendola ElGamal con 57.6953mb. Estos algoritmos tienen una diferencia mínima de 0.01mb al momento de generar la llave pública y privada, causando ninguna criticidad al proyecto.

En el análisis de memoria para el cifrado mostrada en la Figura 46 dentro de la sección de resultados, se destaca que RSA quedó primero con 57.50mb y procediendola ElGamal con 57.69mb. Para el análisis de memoria en descifrado mostrada en la Figura 47 dentro de la sección de resultados, se destaca que RSA quedó primero con 57.51mb y procediendola ElGamal con 57.69mb.

Estos algoritmos tienen una diferencia de memoria en cifrado de 0.19mb y una diferencia en descifrado de 0.18mb, causando ningúna criticidad al proyecto. Los dos algoritmos funcionan muy bien dentro del ambiente de trabajo de una Raspberry PI.

Luego de realizar las pruebas de tiempo y recursos computacionales, se eligió como algoritmo asimétrico a implementar RSA. Se tomó este algoritmo para cifrado asimétrico en el intercambio de llaves criptográficas simétricas debido a que el tiempo y consumo de memoria para las operaciones que realiza, no muestran un riesgo a las operaciones dentro de la raspberry independientes al cifrado. El algoritmo ofrece una gran velocidad de generación de llave pública y privada a comparación de ElGamal. RSA además es uno de los más utilizados para el intercambio de llaves criptograaáficas de algunas empresas tecnológicas que ofrecen servicios similares. El algoritmo ofrece documentación para implementación en nuevos sistemas.

RSA se implementó con una llave criptográfica de tamaño 4096bits.

3. Pruebas de confidencialidad Para poder realizar las pruebas de confidencialidad, se utilizaron distintas herramientas descritas dentro del marco teórico. Estas herramientas se dividieron en dos; herramientas para monitoreo y Spoofing de paquetes de red, y herramientas para crackeo de valores Hash. No se tomó mucho énfasis en ataques de diccionario, ni fuerza bruta

para los textos cifrados; debido a la cantidad de posibilidad de llaves que los algoritmos de Rijndael y RSA generan en base a su arquitectura.

Para el monitoreo de paquetes que se transmiten dentro de un segmento de red, se verificó que la información no se transmita en texto plano, sino que se transmita de forma cifrada y segura. Para el crackeo de valores Hash, se verificó que fueran verdaderamente robustos.

a. Wireshark se utilizó únicamente Wireshark para poder monitorear los paquetes dentro del segmento de red, debido a que las funcionalidades que ofrece, son muy parecidas a las que ofrece ettercap y ARPSpoof. La prueba se realizó en una computadora portátil conectada dentro de la misma red y analizando la información cada segundo. Este análisis se dividió en dos partes; verificar la confidencialidad para el intercambio de llave y verificar la confidencialidad del envío de información cifrada.

Para verificar la confidencialidad del envío de llave; como se puede ver en la Figura 48 en la sección de resultados, Wireshark pudo obtener el envío de la solicitud para el intercambio de llaves. Según esta figura, se logró obtener la llave pública del algoritmo asimétrico de RSA enviada en texto plano. Esta llave pública no se considera un riesgo que personas externas puedan visualizarlas u obtenerlas, de igual manera se obtuvo el valor del Hash, pero debidamente cambiado su valor. El valor del hash fue enviado para verificar integridad dentro del envío de información; si el valor del Hash del receptor es diferente con el remitente, no se acepta la información. Nuevamente, como se puede ver en la Figura 49 en la sección de resultados, Wireshark pudo obtener los valores de respuesta sobre esta solicitud, siendo estos la llave simétrica generada por el servidor en su forma cifrada y su respectivo Hash. En la figura se puede visualizar que la información no es legible para cualquier persona que esté utilizando esta ataque.

Para verificar la confidencialidad del envío de la información cifrada; como se puede ver en la Figura 50 en la sección de resultados, el texto cifrado *cypheredContent* está correctamente cifrado e ilegible para personas externas, de igual manera su valor Hash. El atacante no podrá saber que es lo que se transmite en esta comunicación, pero el servidor al momento de descifrar la información, según la Figura 51, este responde de manera correcta mencionando que la información ha sido almacenada exitósamente. Esta verificación se realizó dos veces para mostrar que la información está correctamente cifrada e ilegible, según las Figuras 52 y 53 de la sección de resultados.

b. FindMyHash se utilizó FindMyHash para poder verificar si los valores Hash que se transmiten a través de la red pueden significar algún riesgo para el proyecto. Se verificó si el valor Hash para cada uno de las solicitudes hacia el servidor puede ser encontrado. Esta herramienta intenta encontrar el valor de forma interna, si no lo logra, posteriormente realiza una busqueda en diferentes ubicaciones de la red para determinar si existe algún patrón o si previamente fue encontrado.

Según las Figuras 54, 55, 56 y 57 en la sección de resultados, muestra que no encontró ningún valor para el Hash dado; por lo que se puede concluir que los valores Hash no brindan riesgo al proyecto.

C. Almacenamiento de información y servicios web

1. Elección del framework de desarrollo La REST API del módulo de almacenamiento de información y servicios web fue desarrollada utilizando el framework loopback. Este framework tiene varios beneficios, pero los dos más importantes, en términos de este proyecto, son la generación automática de documentación y su flexibilidad para trabajar con distintos DBMS dentro de un mismo proyecto. La generación automática de documentación permite una integración rápida con otros módulos del proyecto ya que no es necesario actualizar manualmente la documentación cada vez que se realiza un cambio. Una gran parte del proyecto puede ser modelada con un esquema relacional, pero existe la posibilidad de que una base de datos relacional vea afectado su rendimiento con grandes cantidades de datos. Para lidiar con las posibles consecuencias negativas en el rendimiento se pensó en utilizar una base de datos no relacional que trabajara junto a una base de datos relacional, de aquí nace la necesidad de un framework, como loopback, que pudiera trabajar fácilmente con distintos DBMS dentro de una misma implementación.

2. Modelado del sistema de información En la Figura 58 se muestra el diagrama entidad relación utilizado en la implementación del proyecto. Los diagramas entidad relación se utilizan generalmente cuando se quiere diseñar una base de datos relacional. A pesar de ello se diseñó el sistema utilizando un diagrama entidad relación, ya que describe claramente la relación entre los distintos componentes del sistema de información. Se definieron cinco entidades: Client, Sensor, Calculation, Key y Measurement.

La entidad Client representa al consumidor dentro del sistema. Los atributos que corresponden a esta entidad representan información personal del usuario. Client se relaciona con Sensor con una relación de uno a muchos. Es necesario que Client pueda relacionarse con varios Sensor para proveer al consumidor la posibilidad de monitorear el consumo energético de varias residencias. La entidad Sensor representa el hardware dentro del sistema, únicamente posee atributos que describen las características del hardware.

Sensor se relaciona con tres entidades: Key, Measurement y Calculation. La relación de Sensor con Measurement puede que es la más importante ya que vincula las mediciones registradas en el sistema con la instancia de Sensor que corresponde a la residencia donde se tomaron las medidas. La relación de Sensor con Calculation existe para poder conectar los cálculos provistos por el módulo de análisis con el Sensor que haya producido las mediciones. Finalmente la entidad Key existe para que el consumidor pueda cifrar la información antes de su envío al servidor y que el servidor al recibir la información pueda descifrarla.

3. Elección de los manejadores de bases de datos Los DBMS que se seleccionaron fueron PostgreSQL y HBase. Ya que no se cuenta con presupuesto para la realización de este proyecto se decidió buscar soluciones gratuitas. Como soluciones SQL gratuitas se consideraron MySQL y PostgreSQL. A continuación se presenta el cuadro comparativo de ambos gestores que se usó para decidir.

Cuadro 71: Comparación de PostgreSQL vs MySQL (Gorbatchev, 2014)

Restify	
PostgreSQL	**MySQL**
Conformidad SQL que facilita la migración entre DBMS.	Su falta de dificultad con el estándar SQL dificulta la migración entre DBMS.
Comunidad pequeña	Comunidad y documentación amplia
Rendimiento concurrente estable	Problemas con alta concurrencia
Integridad de datos garantizada	Para garantizar la integridad de información es necesaria configuración adicional
Falta de soporte comercial	Soporte comercial
Escalabilidad horizontal y vertical	Escalabilidad horizontal y vertical

En base a la comparación anterior se elige a PostgreSQL sobre MySQL debido a que es necesario garantizar la integridad de la información y también que es necesario un DBMS que pueda manejar, sin problemas de concurrencia, un volumen grandes de solicitudes.

Tomando en cuenta investigaciones similares, una casa puede llegar a producir alrededor de 53 MB (Mega Byte) de información al mes si se toman mediciones cada segundo. En este proyecto la taza de muestreo utilizada por el módulo de sensores es de una medición por segundo lo que significaría almacenar alrededor 600 MB de datos por casa al año. Esto implicaría que por cada diez usuarios el sistema deberá estar preparada para almacenar más 5 GB (Gyga Bytes) al año. (Barker, 2012)

En la documentación de PostgreSQL se describe que el tamaño de la base de datos puede ser ilimitado lo que asegura escalabilidad vertical. Pero por el volumen de datos es necesario

que el sistema pueda escalar horizontalmente. Actualmente PostgreSQL posee algunas facilidades para poder manejar bases de datos distribuidas pero son relativamente nuevas y no se asegura que el rendimiento del DBMS no se vea afectado por grandes volúmenes de datos. Por ello se seleccionó HBbase para manejar la información de las mediciones, un DBMS no relacional orientado a columnas que funciona sobre el framework Hadoop. HBase proveerá la escalabilidad horizontal necesaria para almacenar grandes cantidades de datos y consultar los mismos sin tener problemas con el rendimiento.

4. Comparación de la versión híbrida con la versión simple Como se mencionó en la sección de resultados existen dos implementaciones distintas de la REST API, una con PostgreSQL como único DBMS (versión simple) y otra que utiliza PostgreSQL y HBASE (versión híbrida). Para elegir una implementación sobre la otra es necesario analizar los resultados de las pruebas, ver de la Figura 60 a la 62.

Las pruebas realizadas con la herramienta JMeter fueron realizadas desde una computadora personal con sistema operativo Windows 10 con procesador Intel core i7 y 8GB de memoria RAM con acceso a internet por medio de un Modem USB Claro con una velocidad de descarga de 6.71Mb por segundo y una velocidad de subida de 1.79 Mb por segundo. Para reducir el efecto del ambiente sobre los resultados de las pruebas la única aplicación en ejecución, aparte de los procesos necesarios para el funcionamiento del sistema operativo, fue la herramienta JMeter. Las peticiones ejecutadas por JMeter fueron dirigidas hacia las dos implementaciones de REST API alojadas en un servidor de Amazon Web Services.

En la Figura 60 se observa que al iniciar la prueba la versión híbrida tiene tiempos de respuesta mayores a un segundo, mientras que la versión simple se mantiene estable entre los 300 y 500 milisegundos. Que varias transacciones, para la versión híbrida, tuvieran tiempos de respuesta superiores a un segundo puede tener distintas causas. Una de ellas es que para hacer una inserción es necesario establecer comunicación con dos DBMS distintos. Otra causa puede ser que la secuencia de inicio de los diferentes procesos de HBase tarde más en iniciar.

En las Figuras 61 y 62 se observa que ambas versiones tienen tiempos de respuesta debajo de 500 milisegundos. Aunque la versión híbrida tiene problemas entre las 14 y 77 transacciones por segundo, donde alcanza tiempos de respuesta superiores a un segundo. Aunque observando las tendencias que siguen las gráficas la versión hibrida tiende a estabilizarse conforme las transacciones por segundo aumentan en contraste al alza que sufren los tiempos de respuesta para la versión simple.

Para asegurarse que los resultados de la prueba no se vieron afectados fuertemente por el ambiente de prueba se sugiere utilizar un servicio como BlazeMeter para realizar las pruebas. Este servicio no se utilizó dentro del proyecto porque es necesario pagar una tarifa de membresía para hacer pruebas con más de 50 usuarios concurrentes.

Para la prueba de 3600 solicitudes en una hora, ver Figura 63, se observaron tiempos de respuesta similares para las dos versiones. Al igual que la prueba pasada la versión híbrida registra sus tiempos de respuesta más altos al inicio pero luego muestra tiempos de respuesta similares, y en algunos casos menores, a los de la versión simple. Al calcular el promedio de los tiempos de respuesta se observa una diferencia de 6 milisegundos. Aunque la desviacón estándar de la versión simple es menor, la diferencia es apenas de 20 milisegundos. Dado que ambas versiones se comportaron de manera similar bajo esta prueba y que la versión híbrida maneja mejor una creciente cantidad de transacciones por segundo se elige la versión híbrida por sobre la simple.

D. Integración

1. Acuerdos sobre protocolos de comunicación Para establecer los protocolos de comunicación se realizaron acuerdos, uno por módulo que se debe comunicar. Al momento de realizar estos acuerdos fue necesario reunirse con el encargado de cada módulo. En estas reuniones se comunicó qué información se necesita y qué formato se utilizó. A partir de esto se hizo un acuerdo donde formalmente se estableció el protocolo para la comunicación.

El primer acuerdo que se hizo fue el acuerdo para el módulo de análisis de datos. En este acuerdo se estableció qué información necesita y el formato que se utilizó para pasar la información. Debido a que el módulo de análisis inició sus procesos de análisis recibiendo un CSV fue necesario adaptarse a esta forma de trabajar, esto implicó que a partir de la información recibida de la base de datos se debe crear un CSV. Durante el desarrollo del proyecto se han realizado cambios al acuerdo original. Hay cambios que son sencillos, pero otros tienen más complejidad. Los cambios considerados complejos son aquellos que no son independientes, que involucran cambios en otros módulos. Un ejemplo de esto es que el módulo de análisis solicite más datos. Esto implica que el módulo de sensores debe capturar estos datos, y que el módulo de almacenamiento de información debe almacenar los datos. Por otro lado hay cambios que son independientes, por lo que basta con comunicar cuál es el cambio y realizar el cambio.

El otro acuerdo que se hizo fue el acuerdo para el módulo de Almacenamiento de la Información y Servicios Web. Debido a que el enfoque de este módulo incluye los servicios web por los cuales se obtiene la información de la base de datos, la información que se necesitaba es la descripción de los servicios web a utilizar. Luego de esto se discutió sobre si la información transmitida era la necesaria para realizar los análisis para hacer los cambios necesarios. Este acuerdo tuvo una mayor participación del módulo involucrado ya que este módulo fue el que realizó los servicios web a consumir. En este acuerdo se establece las URLs a consumir, los parámetros que espera la solicitud y el formato de la respuesta.

2. Flujo de información El módulo de integración debe transmitir información desde el móduo de Almacenamiento de Información y Servicios Web al módulo de Análisis. El

módulo de análisis tiene tres procesos que se deben ejecutar. Estos procesos son: entrenamiento de predicción, predicción y agrupación (clustering). Luego de ejecutar los procesos la respuesta es recibida por el módulo de integración y se consume un servicio web para almacenarla en la base de datos. Para demostrar que sí se realiza la comunicación se realizaron pruebas para cada uno de los proceso de análisis en distintas condiciones. Estas pruebas generan un historial de los mensajes que se envían y reciben. Este historial es un archivo CSV para poder utulizar después herramientas para cuantificar los resultados de las pruebas. Cabe mencionar que se creó un archivo por cada proceso de análisis, esto es uno para clustering, otro para predicción y uno para entrenamiento de predicción.

Las primeras condiciones que se probaron se muestran en los Cuadros 42, 45 y 48. Debido a que se realizaron las pruebas en un ambiente ideal, todas las pruebas fueron exitosas. Pero esto no implica que este módulo es perfecto.

En los Cuadros 43, 46 y 49 no se utilizaron condiciones ideales. Se realizarón estas pruebas para demostrar la continuidad del programa. Estas condiciones lo que causaban es fallos en el envío de datos a los servicios web y fallos en los análisis de clustering. Los Cuadros 46 y 49 muestran que se realizó el proceso exitosamente un 98 % de las veces. El otro 2 % no fue exitoso porque al iniciar el proceso, al enviar la información a los servicios web, hubo problemas de conexión. Por esta razón solo se puede observar que se envió la información pero no se recibió alguna respuesta, no se estableció la conexión con el servicio web. En cuanto al Cuadro 43 se muestra un total de 37 %. Se obtuvo este resultado porque los datos almacenados de consumo no se cumplían los requisitos mínimos para ejecutar el proceso de clustering. Entonces lo que muestra este cuadro es que se le envía información al módulo de análisis y este proceso falla por lo que no se recibe respuesta. Esto implica que la plataforma de integración tiene la capacidad para manejar este la falla del proceso de análisis y no detener ejecución.

Las últimas condiciones que se probaron se orientaron en hacer fallar el módulo de integración. Para esto se tuvo control del acceso de internet de la computadora que estaba ejecutando el este módulo. Estas pruebas se hicieron en las mismas condiciones que las pruebas anteriores, con la diferencia que se desconectaba el internet a propósito para observar fallos.En los Cuadros 44, 47 y 50 se obtuvieron los porcentajes más bajos. Estos resultados demuestran que el internet es un factor importante en la integración, ya que sin él no se pueden consumir los servicios web y no se puede iniciar el flujo de información para realizar los análisis.

El módulo de integración propone las siguientes etapas para transmitir la información entre estos dos módulos.

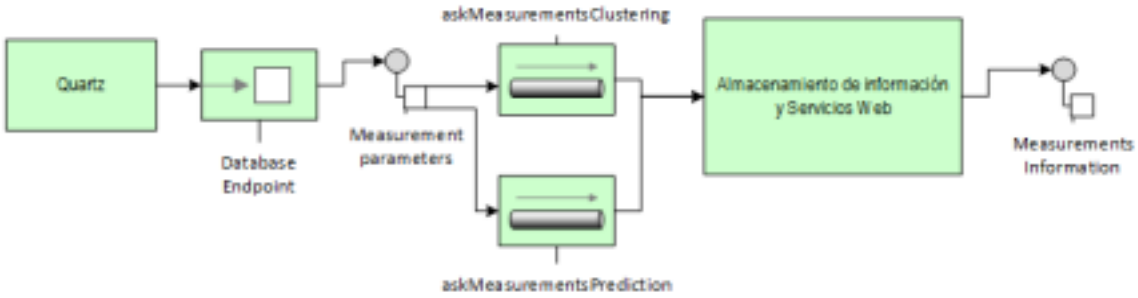
- Se solicita información del consumo o sensores a la base de datos
- La información de consumo o sensores se transforma y transmite al módulo de análisis

- El módulo de análisis realiza su proceso y devuelve sus resultados
- Los resultados del análisis se guardan en la base de datos

Para compreder a mayor detalle este proceso a continuación se detalla estas etapas:

1. Solicitar información Para el proceso de entrenamiento de predicción y clustering, se siguen los mismos pasos. La diferencia son los parámetros que se generan, ya que el proceso de clustering solicita información de consumo por día y el de entrenamiento de predicción solicita información de consumo de un mes.

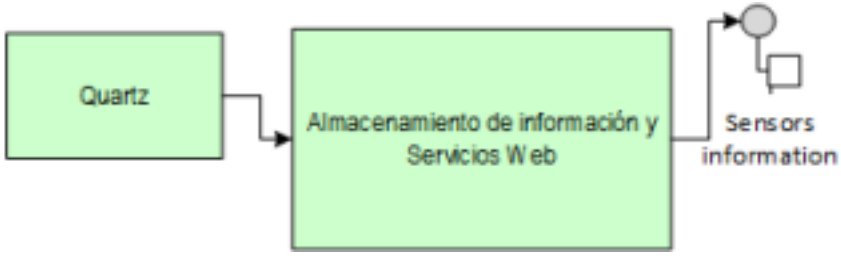
Figura 112: Diagrama solicitar información de consumo



Esta etapa es iniciada por el componente de Quartz, el flujo para el análisis de clustering se debe iniciar una vez por día y para el entrenamiento de predicción se debe iniciar una vez por mes. Quartz se encarga de iniciar el flujo a través de una sintaxis llamada cron que permite ejecutar procesos en base al tiempo. Debido a que la solicitud de consumo requiere parametros para filtrar la información, el flujo sigue por "Database Endpoint". El cual es el encargado en crear los parámetros, esto incluye los parámetros para clustering y/o para entrenamiento de predicción. Dependiendo de que proceso se quiera ejecutar de análisis, "Database Endpoint' deposita los parámetros contruidos en un canal específico. El canal del mensaje para el proceso de clustering se llama "askMeasurementsClustering" y para el proceso de entrenamiento de predicción se llama "askMeasurementsPrediction". Luego se hace la solicitud con los parámetros a un servicio web del módulo responsable de esto, representado por el componente "Almacenamiento de Información y Servicios Web". De esta solicitud se recibe una respuesta llamada "Measurements Information".

Debido a que el inicio del flujo para el proceso de predicción es diferente, se propone el siguiente diagrama para este inicio.

Figura 113: Diagrama solicitar información de sensores

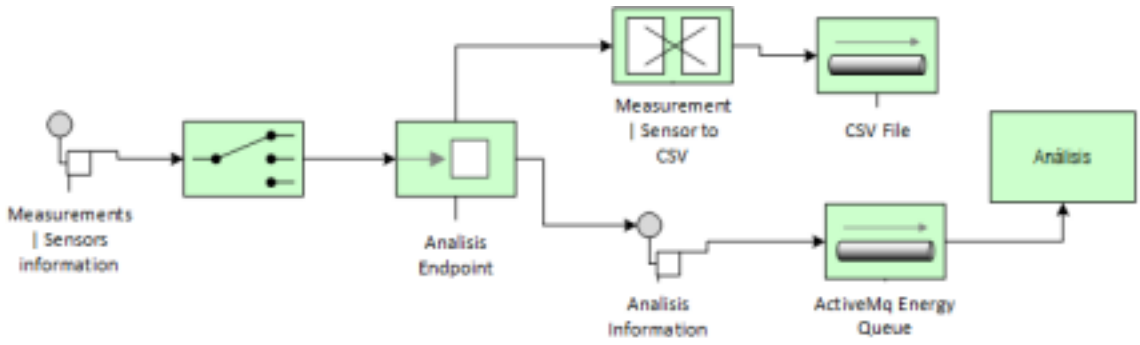


Esta etapa inicia igual por el componente de Quartz, quien se encarga de iniciar el flujo de predicción una vez por hora. Debido a que no se necesitan parámetros para esta solicitud, Quartz inicia la etapa al consumir un servicio web del componente mostrado. Esto genera una respuesta, el mensaje, con la información de los sensores.

Los resultados de las pruebas para esta etapa se muestran en las primeras dos filas de los Cuadros 42, 43, 44, 45, 46, 47, 48, 49 y 50. Estos resultados muestran la cantidad de veces que se hizo una solicitud al módulo de Almacenamiento de Información y Servicios Web y la cantidad de respuestas que se obtuvieron de esta solicitud.

2. Enviar información al módulo de Análisis Esta etapa es el momento en el flujo de información en el cual el módulo de integración traduce la información de los servicios web al protocolo que se estableció con el módulo de análisis de datos.

Figura 114: Diagrama enviar información al módulo de análisis



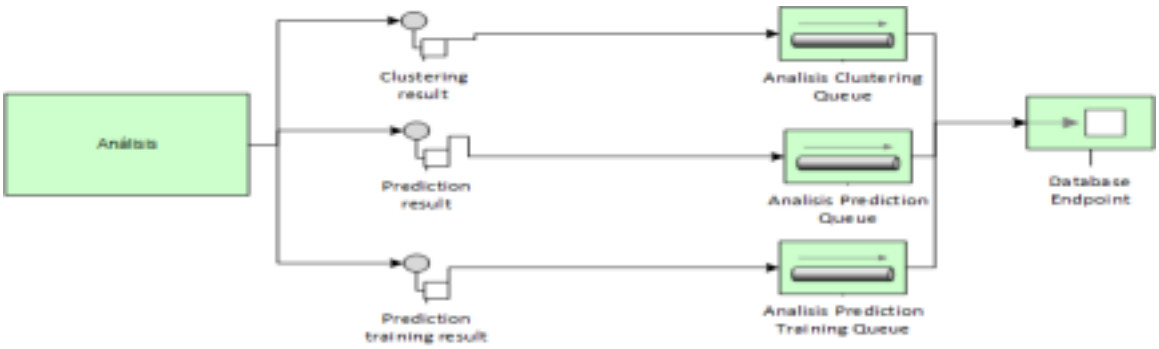
Esta etapa inicia al finalizar la etapa anterior. Para simplificar un poco el mensaje con el que se inicia puede ser información de consumo o información de sensores, el final de las dos posibilidades de la etapa anterior. Este mensaje, “Measurements | sensors Information”, se envía al router quien se encarga de enviarlo al método correcto al punto extremo llamado “Analisis Endpoint”. Para cada uno de los procesos de análisis se crea un CSV. Para crear este CSV se traduce la información que se recibió del servicio web, para seguir el protocolo del CSV para el tipo de análisis que se este

realizando. Esta traducción se coloca en un canal para luego ser enviado a un archivo dentro de la carpeta de python. Una vez creado el CSV que necesita el módulo de Ańalisis de Datos, se crea un mensaje que se envía a python para transmitir esta información al módulo de análisis. Esta es la parte del flujo que utiliza ActiveMq. Una vez creado el mensaje, este se coloca en una canal de mensaje llamdo "ActiveMq Energy Queue". Este mensaje se mantendrá dentro de este canal hasta que sea leído en Python. Una vez se leyó el mensaje en Python por el módulo de Integración se transmite la información al módulo de análisis que se encuentra en este lenguaje.

Los Cuadros 42, 43, 44, 45, 46, 47, 48, 49 y 50 también muestran los resultados obtenidos de las pruebas para esta etapa. La tercera fila muestra la cantidad de mensajes que se enviaron al módulo de análisis, estos mensajes inlcuyen el nombre del archivo CSV que se creó, lo que implica que sí se creó el CSV.

3. Recibir resultados de análisis En esta etapa se pueden recibir tres posibles mensajes del módulo de análisis. Es un mensaje por cada proceso que ejecuta el módulo de análisis: clustering ("clustering result"), predicción ("prediction result") y entrenamiento de predicción ("Prediction training result"). Esta etapa da inicio cuando el módulo de análisis envía una respuesta. Depen-

Figura 115: Diagrama recibir información del módulo de análisis



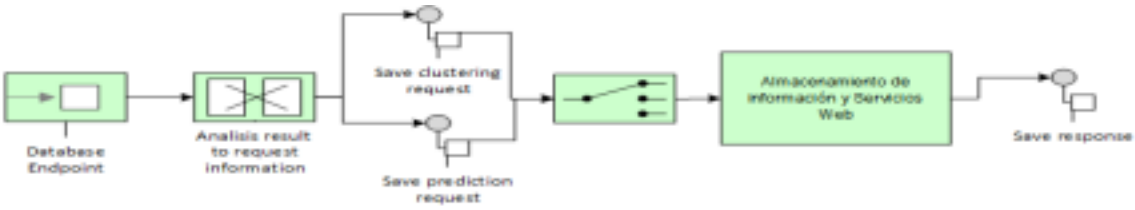
diendo del tipo de proceso que se ejecutó, se envía su respuesta a un canal específico. El resultado de clusterint se envía al canal "Analisis Clustering Queue", el de predicción se envía a "Analisis Prediction Queue" y el de entrenamiento de predicción se envía a "Analisis Prediction Training Queue". Cada uno de estos mensajes envía la información a un método distinto en el punto extremo llamado "Database endpoint". Cabe mencionar que debido a que el análisis envía su respuesta desde Python, se vuelve a utilizar ActiveMq para envíar esta respuesta al ambiente de Apache Camel. Los tres canales antes mencionados son de ActiveMq y se leen desde el ambiente de Apache Camel. Este es el final del flujo de información del proceso de entrenamiento de predicción ya que no se almacenan sus resultados en la base de datos.

Esta etapa también se encuentra en los Cuadros de resultados 42, 43, 44, 45, 46, 47, 48, 49 y 50. Esta etapa esta representada por la cuarta fila de cada proceso, que es la información que envía

el módulo de análisis. Si hubo una respuesta por parte del módulo de análisis quiere decir que se cumplió la comunicación, ya que no se estableció el manejo de errores. Se obtiene una respuesta del módulo de análisis si logró realizar alguna acción con la información que se le envió.

4. Guardar resultados de análisis Esta es la última etapa del flujo de información de la comunicación entre módulos para el proceso de predicción y clustering, el proceso de entrenamiento de predicción ya no llega a esta etapa.

Figura 116: Diagrama guardar resultados de análisis



Esta etapa inicia en “Database Endpoint”, donde se tiene el mensaje con la respusta del módulo de análisis. Para almacenar la información se realiza un proceso de traducción que se encarga de traducir el formato de la respuesta del módulo de análisis a los parámetros para consumir un servicio web. En el diagrama se muestran dos tipos de mensaje con el objetivo de ilustrar que es el mismo proceso para ambas respuestas. Los parámetros obtenidos se envían al router quien se encarga de enviarlos al servicio web correcto para realizar el almacenamiento en la base de datos. Luego de esto se recibe la respuesta de la solicitud del servicio web que indica que se guardó la información.

Esta etapa es la penúltima fila de los Cuadros 42, 43, 44, 48, 49 y 50. En esta fila se muestra la cantidad de mensajes que se enviaron al servicio web. No se midió la cantidad de respuestas obtenidas debido a que no influye en el flujo de información, solo es el paso final.

E. Análisis de datos

Se consideraron diversos lenguajes, llegando a reducir las opciones a tres de los más populares usados en el campo, siendo R, Julia y Python. Julia a pesar de ser un lenguaje relativamente nuevo en comparación de Python, ha sido catalogado por la comunidad como un lenguaje bastante rápido y escalable. Pero al ser relativamente nuevo para el momento en que se llevó a cabo la elección, su comunidad aún no es demasiado fuerte y tampoco cuenta con suficientes librerías para poder llegar a realizar los diferentes aspectos que se buscan dentro del proyecto. R, por su parte cuenta con una comunidad bastante grande, además de ser bastante popular en el campo del análisis de datos, pero la sintaxis de este puede llegar a ser bastante engorrosa, además de no ser tan escalable como otros lenguajes, Python incluido en estos. Python por otro lado es una de las opciones que

mejor se acopla para poder llevar a cabo las tareas necesarias no sólo para el análisis de los datos sino también el manejo de los datos y la visualización de los mismos. De igual manera Python cuenta con diversas librerías bien soportadas para poder hacer uso de estos confiando mejor en sus resultados.

Al haberse decidido el uso de Python para la implementación de los métodos seleccionados se decidió el uso una librería principalmente, la cual tiene su objetivo en el área de aprendizaje de máquina, siendo esta Sckitlearn. Esta es una de las librerías más populares para la implementación de algoritmos de aprendizaje de máquina, teniendo la facilidad de poder ser personalizable a modo de lograr lo que se busca dentro de los alcances del proyecto, este de igual manera tiene implementaciones disponibles de varios de los algoritmos de aprendizaje de máquina más populares como KMeans e incluso Máquina de Vectores de Soporte (o Support Vector Machines, en inglés).

Se deben considerar las complejidades de tiempo de los algoritmos usados dentro del área de análisis de este proyecto. Por ello estas se resumen en el cuadro que se presenta a continuación.

Cuadro 72: Complejidades Big O de los Métodos usados.

Árboles de decisión	$O(f_{features} * n_{muestras} * log(n_{muestras}))$ (Pedregosa, 2012)
Máquina de vectores de soporte	$O(f_{features} * n^2_{muestras})$ (Chang & Lin, 2013)
KMeans	$O(n_{muestras} * d_{dimensiones} * k_{centroides})$ (Eliasson & Rosén, 2013)
Clústeres jerárquicos	$O(n^2_{muestras} * log(n_{muestras})$ (Rokach & Maimon, 2010)
Método de la silueta	$O(n^2_{muestras})$ (Petrovic, 2006)
Pham *et al*	$O(k_{iteraciones} * 2^{n_{muestras}})$ Ecuación (VI..3)

Como ya se mencionó, aspectos como la eficiencia y complejidad deben ser considerados fuertemente dado el tipo de proyecto que se realizó debido a que este debe buscar ser lo más rápido posible en responder, así como también no ser demasiado pesado al momento de ser entrenado dado que deberá servir de apoyo para el monitoreo de energía eléctrica de un usuario. De igual manera se consideró la factibilidad de escalabilidad de los algoritmos, es decir, se consideró cuan fácilmente adaptables serían para las diferentes circunstancias del problema, sin tener que llegar a depender de una arquitectura específica dentro de su implementación como suele pasar con algoritmos basados en redes neuronales, o bien que se dependa demasiado de la forma en que los datos se encuentran distribuidos. Dado que estos deben ser capaces de ser entrenados de la forma más fácilmente posible así como también de poder realizar una predicción o clasificación, según sea el caso, con una alta probabilidad de acierto. Tras haber evaluado los aspectos anteriormente dichos, así como la popularidad de uso en estudios similares se llegó a determinar que los algoritmos a

utilizar serían Regresión con Máquina de Vectores de Soporte (SVR por sus siglas en inglés) y Árboles de Decisión (DT por sus siglas en inglés) para el área de predicción, y para la parte de clustering se eligió a KMeans y Clústeres Jerárquicos en su versión aglomerativa.

Uno de los aspectos importantes a tomar en cuenta para las implementaciones es la limpieza de datos, para esto se buscaron aquellas casillas donde el valor de consumo eléctrico no tuviera valor alguno o contara con un argumento no convertible a número decimales, en tal caso se decidió deshacerse de estos datos dado que no aportan ningún tipo de valor al proceso. Nótese que la limpieza de los datos es importante debido a que es gracias a esta que se busca que los algoritmos se comporten de mejor manera y puedan acercarse a tener un mejor resultado al realizar el procesamiento correspondiente.

La literatura (Brownlee, 2013) recomienda que se haga una estandarización de los datos, debido a que los algorimos de aprendizaje de máquina pueden comportarse de forma inadecuada si no se parecen a la distribución normal. El proceso se llevó a cabo usando el método *StandardScaler* de la librería Scikit Learn. Además se recomienda hacer una reducción de características con las cuales se está trabajando, con el fin de evitar un sobreajuste del modelo, aumentar la exactitud de los métodos y reducir el tiempo de entrenamiento. Para esto, como bien se ha mencionado en la parte teórica de este trabajo, existen diferentes aproximaciones de las cuales la más recomendable es la que incluye los métodos del tipo envolventes, dado que estas ayudan a detectar las posibles interacciones entre las variables. Nótese que se eligió este tipo de método dado que se consideró que el número de observaciones era lo suficiente como para poder evitar el sobreajuste del modelo, y además que el número de características a evaluar era pequeño como para no caer en tiempos de ejecución altos. Se seleccionó el método conocido como Árboles Aleatorios Extremos. El mismo método ofrece la ventaja de no sólo usar un subconjunto de datos, los umbrales son extraídos de forma aleatoria para cada característica candidata y el mejor de estos es usado como partición. Esto tiene como ventaja que no solamente reduce la varianza del modelo un poco, sino que permitía la comparación entre los métodos dado que no dependía de alguna característica de los algoritmos que se estaban usando.

A pesar que se consideró tener un acercamiento basado en la eliminación recursiva de características en los métodos, esto tuvo que ser descartado dado que al variar entre modelos se hubiera podido llegar a tener diferentes variables para cada método. Como se puede observar en el Cuadro 35, las variables elegidas a través de las distintas iteraciones con diversa cantidad de datos fue variando, llegando a trabajar con un máximo de 3 variables como máximo y como mínimo 2. Esta variación no sólo en cantidad sino en tipo de datos puede deberse a que al cambiar la cantidad de datos, se hacía más perceptible para el método cuales eran las características de los datos que más peso llegaban a tener dentro de la aplicación. Al observar la figura 83, se hace perceptible

que la variable que permaneció en todas las iteraciones y que tuvo mayor influencia en el modelo fue la que hace referencia a la hora, seguido por el día; de forma intuitiva se hace claro que estas variables pueden llegar a ser las que más peso tienen dentro del modelo, puesto que el gasto de energía eléctrica va ir cambiando conforme avanza el día; por su parte la variable día tiene sentido que haya sido elegida como una variable con importancia dentro del modelo dado que la rutina de las personas puede variar según el día, sin embargo, la variable que describe si es fin de semana o no, no parece hacer un aporte significativo al modelo, como se puede observar en la figura 83, es una de las menos importantes de acuerdo al peso. De igual manera, es importante resaltar que el haber divido el conjunto de datos de forma aleatoria para las partes de entrenamiento y prueba, fue de suma importancia dado que de esta manera los métodos se acoplaron de mejor manera a los datos y quedaron mejor preparados para funcionar de manera adecuada ante un dato no visto antes.

Ahora bien, al comparar el desempeño en términos de velocidad y exactitud (figuras de 65 la a la 80), de parte de los métodos de Árboles de Decisión, llamado de aquí en adelante como DT (por sus siglas en inglés) y Máquina de Vectores de Soporte, llamado de aquí en adelante como SVR (por sus siglas en inglés), se puede apreciar como el método de DT resulta hacer un mejor trabajo al adaptarse a los datos y con ello ofrece una mejor regresión y por ende una mejor precisión en términos de exactitud en comparación a los resultados ofrecidos por SVR. Esta última para la mayoría de ejecuciones realizó predicciones con puntaje de exactitud por debajo de cero. Esto se puede observar de forma más clara en el cuadro 34, donde el puntaje de exactitud r^2 de DT permanece sobre cero y algunas veces supera el 0.70. Entonces, el desempeño tan pobre que tuvo SVR en las diferentes pruebas utilizadas se puede deber al hecho que se haya utilizado un núcleo lineal en lugar de un núcleo de función gaussiana de base radial (rbf por sus siglas inglés), este último se recomienda en varios estudios, pero tuvo que ser descartado por la cantidad con la que se trabajó. Puesto que para grandes volúmenes de información este método sería muy lento porque trabaja con una función exponencial.

Es importante notar que el método de SVR se basa principalmente en dos hiperparámetros, C y ϵ, que a pesar de haber sido buscados de forma automática usando un método de validación cruzada con búsqueda por rejilla, pareciera ser que estos no se terminaron de acoplar de forma adecuada a los datos, pues como se puede observar en el cuadro 36, para casos como el donde se trabajó con 1048575 datos, se obtuvieron valores de 0.5 y 1 para ϵ y C respectivamente; lo cual se a que la cantidad de error que se está aceptando es de 0.5, y que se va a permitir que hayan varias clasificaciones erróneas en la fase de entrenamiento. Al ser C un valor relativamente pequeño en comparación a los datos usados, hará que el optimizador del método busque en un hiperplano con mayor margen de separación, esto lleva a que se clasifiquen erróneamente más puntos en el

hiperplano; obsérvese que dicha falla en la elección de hiperparámetros puede deberse a una mala elección del diccionario con los posibles valores para la parte de la validación cruzada, haciendo que no se consideren algunos parámetros que pudieran haber mejorado el método.Una forma de mejorar este aspecto pudiera ser observar el comportamiento de los datos previamente y basados en la experiencia poder determinar un mejor rango de parámetros, pero este no es factible para las finalidades del proyecto dado que lo que se desea es dejar la funcionalidades lo más automatizadas posibles por lo que una evaluación de forma visual sería para nada escalable.

En la figura 81 y el cuadro 33, se puede apreciar como el tiempo de ejecución del DT resulta ser más rápido que el del SVR, esto se refuerza al observar las ecuaciones del cuadro 72, se puede ver que justamente el DT al tener una complejidad temporal de orden lineal es más rápido que tener una complejidad de orden cuadrático, como es el caso del método SVR; de igual manera dicho comportamiento se puede apreciar en la figura 81, donde justamente se puede observar como las gráficas presentan una conducta que se acopla a los tiempos de ejecución dados anteriormente.

Con los datos obtenidos desde el sensor del proyecto, se probaron estas observaciones con los métodos de DT y SVR. Con lo cual al observar el cuadro 38, se aprecia que ambos métodos tuvieron una exactitud bastante buena al ser sobre el 95 %. Aunque ambos tuvieron una efectividad superior al 99 %, podría decirse que el algoritmo de Árboles de Decisión tuvo una efectividad mayor por 0.56 %. A pesar que ese valor no es significativo, en el cuadro 37 se puede comprobar que también tuvo mejor rendimiento en tiempo que la SVR.

Cabe mencionar que para esta ejecución de los métodos, el SVR usó un C bastante alto, el cual pudo haber sido uno de los factores que ayudaron a que se lograrán un puntaje de r^2 tan alto dado que permitió que el método usara un margen más pequeño en el hiperplano. Esto permitió que se clasificaran erróneamente menos valores. Por otra parte el hecho de que el valor ϵ fuera de 0.1 ayudó a que no se aceptara un margen de error mayor en los valores dados por el método. Se pudo comprobar que las variables día, hora y numeroDia, son consideradas como las características más importantes para el modelo (véase cuadro 39). Considerándose los aspectos mencionados previamente en áreas de velocidad y exactitud basada en el coeficiente r^2, se puede decir que el método de DT presenta ser un mejor candidato para ser implementado dentro del servidor de prueba dado que resulta más rápido y con una mejor exactitud en comparación que el método de SVR.

Los métodos de KMeans y Hierarchical Clustering fueron seleccionados dada su popularidad en estudios similares donde se buscaba clasificar el consumo de energía eléctrica de hogares y empresas, como en las investigaciones de Flath y sus colegas(Flath *et al*, 2012), Gabe-Thomas y su equipo (Gabe-Thomas, Walker, Verplanken, & Shaddick, 2016) así como en el trabajo de Lavin

y Klabjan (Lavin & Klabjan, 2014). Otra de los aspectos tomados en cuenta para su elección es que formaran parte de la librería ScikitLearn.

Cabe destacarse que para el método de Hierarchical Clustering, se usó una versión conocida como Agrupación de Aglomeración o Agglomerative Clustering, que este se caracteriza por tener una aproximación "de abajo hacia arriba", haciendo que cada observación empiece en su propio clúster. Este se usó con un criterio de enlace basado en la distancia promedio entre cada una de las observaciones, el cual definitivamente fue un factor determinante para los resultados obtenidos. Puesto que si las observaciones tendían a tener una distancia promedio lo suficientemente grande como para poder separarse en un clúster diferente esto causaría que se generara un nuevo clúster que podría llegar a tener un solo elemento. Debe tomarse en cuenta que este método inicia con un clúster para cada observación, esto lleva a que este algoritmo se base fuertemente en la cantidad de grupos que se le especifica previamente.

En lo que respecta a KMeans, se eligieron K observaciones de manera aleatoria para ser los centroides de cada uno de los clústeres. Para asegurarse de que se generen siempre los mismo números aleatorios, con el fin que no queden siempre grupos diferentes se utilizó el método "seed" de la librería "random" de Python. Debe hacerse énfasis en que este método también es dependiente del número de clústeres que se especifica previo a su ejecución, generando tantos clústeres como se diga, lo cual provoca que se tenga que usar otro método para elegir la cantidad de clústeres más adecuada para el modelo.

Por otra parte, al observar las figuras 88, 89 y 90, se puede apreciar que el método de Clústeres Jerárquicos con el acercamiento utilizado, resulta ser más rápido que el método de KMeans, a pesar que la cantidad de clústeres pequeña. En el caso en el que se debían buscar 10 clústeres como máximo, la diferencia de rendimiento entre ambos algoritmos fue bastante poca (0.027 segundos). Sin embargo, al ampliar el número máximo de grupos hasta 184, la brecha entre ambos métodos fue de 43.45 segundos aproximadamente. Esto se puede observar más claramente en la figura 91, donde KMeans presenta un comportamiento casi lineal, mientras que para el caso de Clústeres Jerárquicos pareciera tener una comportamiento acorde a su tiempo de ejecución. En esta investigación no se consideraron cantidades elevadas de datos por lo que los algoritmos de clústeres jerárquicos tuvo un buen desempeño. Sin embargo dada su complejidad exponencial a medida que aumente el número de datos, su rendimiento irá decayendo. Esto no sucederá con el método de KMeans debido a su naturaleza lineal. Si se considera que el proyecto en sí debe ser capaz de escalar sin mayor contratiempo, el hecho que la rapidez se mantenga conforme aumentan los datos, es un factor determinante en la elección del método a usar puesto que el servicio se debe mantener lo más rápido posible.

Como se ha mencionado en el párrafo anterior, ambos métodos necesitan saber en cuántos clústeres deben agrupar los datos dados, para esto existen diferentes métodos, siendo unos basados en la observación de gráficos y otros basados en el cálculo de diferentes aspectos de los clústeres. Uno de ellos es conocido como "la regla del pulgar" (o rule of thumb), que no es más que una simple operación matemática, pero este puede que no dé el mejor número de clústeres dado que solamente se basa en una división sobre la cantidad de datos. Por ello se estudiaron los métodos propuestos en la investigación de Pham y su grupo (Pham *et al*, 2004), y el método conocido como el de la silueta. Como se puede notar en la gráfica 86 el algoritmo de Pham *et al*, presenta un tiempo exponcial al iterar sobre una cantidad de clústeres mayor a 30, esto también se vio reflejado en la complejidad de este algoritmo experesada en la ecuación (VI..3). Esto lleva a tener que descartar este algoritmo dado que para la plataforma propuesta se necesita que los métodos que se implementen trabajen lo más rápido posible, y considerando que el sistema puede llegar a tener más de 100 usuarios, este método tardaría demasiado tiempo en encontrar la cantidad óptima de agrupaciones.

Por otro lado al observar la gráfica 87 se puede constatar que el método de la silueta presenta una complejidad de orden n^2, lo que hace que tenga un mejor desempeño en comparación al método de Pham *et al*. Con este método se busca que los grupos sean lo más densos y separados posibles. Si los datos no cuentan con similitudes suficientes entre sus características, puede llevar a que el número de grupos propuesto tienda al límite dado para la cantidad de agrupaciones. Esto se observa en el cuadro 41, donde sólo el caso de 10 clústeres se determinó que la mejor cantidad de grupos era de 2; mientras que para los casos donde el máximo era de 100 y 184, resultó que la cantidad de clústeres era de 99 y 183 de forma respectiva. Esto se puede deber al hecho que los datos fueron generados de forma aleatoria basados en un rango que fue dado por las observaciones tomadas de una base de datos pública sobre el consumo de energía eléctrica en países de América Latina. Al tener datos aleatorios, estos no tenían suficientes similitudes para ser considerados dentro del mismo clúster. Otro aspecto importante a mencionar es que para el uso del método de la silueta, se especificó que la métrica para distancia que debía ser usada era la distancia Euclidiana, lo cual pudo afectar en la determinación de cuán bien esparcidos estaban los grupos. Puesto que el uso de los cuadrados haya hecho que se aumenten las distancias y si los datos son muy poco similares. Esto lleva a que las distancias sean aumentadas, lo cual provocaría la generación de grupos que tienden al número máximo de posibles clústeres especificado. En este caso sería interesante realizar pruebas con una métrica de decisión dada por una distancia de Manhattan, con el fin de tener solamente las distancias sin mayor arreglo como factor para determinar la proximidad entre una observación y los clústeres definidos.

Es necesario realizar pruebas con estos métodos pero usando métodos reales, mapeados desde un sensor en el cuál se confíe, pudiendo ser este un factor de decadencia en las pruebas realizadas

durante esta investigación. Así mismo, es recomendable realizar más investigaciones para determinar de mejor manera cuál método resulta más recomendable para determinar la cantidad de clústeres en base al patrón de consumo.

Entonces, dado que según los tiempos de ejucución teórico encontrados y bien representados en las ecuaciones del cuadro 72 para el caso de KMeans y Clústeres Jerárquicos, se puede decir que para cuando se tenga una mayor cantidad de datos el método de Clústeres Jerárquicos será más lento que el de KMeans, y dado que para cantidades pequeñas de datos, como se puede ver en la figura 91 la diferencias de tiempo, no pasa a mayores cantidades como horas, y que además los resultados son bastante parecidos en cuanto a las agrupaciones formadas por ambos métodos variando solamente en uno o dos datos colocados en diferentes agrupaciones, se puede decir que el método KMeans es un mejor candidato para ser usado dentro del servidor de prueba del sistema.

F. Interfaz de usuario

Analizando los resultados, se puede notar que el proceso la creación de una interfaz de usuario usable y útil es continua y cambiante, como se demostró en este proyecto, y que aún hay espacio para mejoras aplicables a la interfaz.

Los resultados 1 y 2 muestran los datos recolectados por los estudios de usabilidad y el rendimiento de los usuarios para cada tarea, así como observaciones en su reacción al ejecutar la tarea. Analizando los resulados, rápidamente se pueden detectar posibles mejoras en la interfaz entre cada iteración. Según se observa en el Cuadro 69, el promedio de tiempo mejoró para el segundo estudio de usabilidad respecto del primero, aunque también esta mejora se pudo deber a la familiaridad que ya tenían los usuarios con la interfaz, al haberla utilizado en el primer estudio de usabilidad. De cualquier manera, el proceso de usabilidad sigue siendo iterativo y muy frecuentemente, los primeros estudios de usabilidad son los que más mejoras inducen a la eficiencia en el uso de la interfaz.

Las personas que realizaron la prueba de usabilidad corresponden a una mayoría de adultos hombres entre 35 a 55 años, quienes en su mayoría son quienes pagan la factura de luz en sus hogares. Este es el grupo objetivo de esta interfaz, pues son quienes más se preocupan por el consumo energético; y por ende, quienes más utilizarán esta interfaz. Entre los entrevistados, la mayoría nunca ha utilizado una interfaz para monitoreo de consumo eléctrico, probablemente por la escasez de servicios de este tipo en Guatemala. La mayoría entiende los conceptos básicos sobre energía eléctrica, lo que simplifica el nivel de comprensión necesario para saber cuánto consumieron y deben pagar. La mayoría también indicó que se le facilita aprender a utilizar una nueva interfaz web, lo cual, no necesariamente garantiza que sepan utilizar esta interfaz con facilidad ya que todas las interfaces tienen sus peculiaridades. Por último, los usuarios respondieron que no utilizarían

esta aplicación diariamente, y que probablemente la utilicen una o menos de una vez al mes. Este descubrimiento indica que se debe de atrapar al usuario de otra forma para que la utilice, como un sistema de notificaciones y alertas automáticas, lo cual se incluyó en las recomendaciones para el futuro de esta interfaz.

El resultado 3 presenta las funcionalidades que se definieron para el sistema disponibles a través de la interfaz de usuario. Cabe resaltar que esta es una versión final para fines del presente módulo. Sin embargo, esto no significa que no esté abierto a mejoras para el futuro del proyecto. Las funcionalidades se encuentran validadas por una variedad de usuarios, aunque la mayoría estaba conforme con solo saber el dato sobre su consumo hasta la fecha y cuánto debían pagar. Por esta razón, quizás el usuario no fue lo suficientemente exigente en cuanto a la utilidad de la interfaz. Siempre existen usuarios avanzados que necesiten saber sobre este análisis y por eso también se pidió su opinión a lo largo del diseño de la interfaz.

El resultado 4 muestra las capturas de la interfaz en su última versión. Cabe resaltar que, como en el caso anterior, la versión actual es una versión final para fines de este módulo, pero no dictamina una puesta en producción inmediata. Aún se pueden realizar más estudios de usabilidad siguiendo la metodología utilizada en este trabajo para descubrir aún más mejoras, tanto en usabilidad como en eficiencia y utilidad. Estas imágenes muestran todas las pantallas a las que se puede acceder a través de la interfaz, empezando desde la pantalla de carga, pantalla de ingreso, pantalla principal, secciones y configuración de usuario. Esta versión ya contiene herramientas propias de una aplicación web, como consumo de APIs lo cual facilita su integración con el proyecto general. Finalmente, este es el resultado más importante, porque es lo que servirá en la integración.

VIII. Conclusiones

Tras haber concluido el proyecto, de parte del módulo de sensores y protocolos de comunicación se concluye, que se implementó un sensor YHDC TCVH para obtener mediciones de voltaje en un hogar con un porcentaje de error de aproximadamente 0.24 %. También que se implementó un sensor YHDC SCT130-000 para obtener mediciones de corriente en un hogar con un porcentaje de error máximo de 1.34 %. Así mismo que se utilizó un microcontrolador PIC16F88 para digitalizar las señales de salida de los sensores de voltaje y corriente. Además que para procesar la información digitalizada, realizar el cálculo de potencia y transmitir la información a un servidor en línea se utilizó la plataforma Raspberry Pi 2. Y que se diseñó un circuito para proteger los niveles de voltaje de la Raspberry Pi y lograr establecer comunicación vía el protocolo UART con el microcontrolador encargado de la recepción de datos por radiofrecuencia.

En el módulo de seguridad de la información se determinó que para poder transmitir mensajes privados a través de canales inseguros dentro de la red; es necesario utilizar algoritmos de cifrado con seguridad robusta y eficiencia de sus operaciones. En base a los criterios o enfoque que se le dará, se debe de implementar un algoritmo que cumpla con los requisitos del proyecto, ya sea más eficiencia o más seguridad. Al momento de utilizar un algoritmo criptográfico simétrico o asimétrico, es necesario que los algoritmos posean lo mínimo requerido de seguridad, siendo estos para los algoritmos simétricos; llave criptográfica preferiblemente de 256 bits, tamaños de bloque de 128 bits como mínimo y conocer si ha sido vulnerado por algún ataque. Para los algoritmos asimétricos es necesario una llave de tamaño como de 2048 hasta 4096 bits en múltiplos de 16, verificando que la implementación sea fácil y no muy compleja.

Además, se concluyó que al momento de implementar algoritmos de cifrado; es necesario verificar si la información no ha sido alterada, o si el usuario o dispositivo que envía la información sea perteneciente al proyecto. Para verificar esto, se deben de utilizar las funciones Hash como parte de la integridad de los datos. Otro aspecto importante para seleccionar el algoritmo, es la disponibilidad. Este factor es muy importante para sistemas más grandes, el cuál este asegurará que la información estará disponible a todo momento a los usuarios. Disponibilidad se puede relacionar con la eficiencia y bajo procesamiento de recursos computacionales, realizando una menor carga al dispositivo y al servidor. Dependiendo nuevamente del enfoque del proyecto, se puede optar a brindarle más importancia al pilar de confidencialidad, como al pilar de integración o disponibilidad.

También se concluye que se elaboraron implementaciones de tiempo y consumo de memoria para tres algoritmos simétricos y dos algoritmos asimétricos. En base a los algoritmos simétricos, brindaron un consumo de recursos muy similar para las operaciones de generación de la llave criptográfica, el cifrado y descifrado. Para el consumo de tiempo, AES Rijndael fue uno de los algoritmos más rápidos y con buenos márgenes de seguridad, además de ser utilizado por muchas empresas tecnológicas de la actualidad. Para los algoritmos asimétricos, RSA ha sido el más rápido y común para el intercambio seguro de llaves.

En lo que respecta al módulo de almacenamiento de información y servicios seb se seleccionó para el almacenamiento de información los DBMS HBase y PostgreSQL. Como framework de desarrollo para la REST API se seleccionó loopback. Con esto se da cumplimiento al objetivo de de seleccionar las herramientas a utilizar para el almacenamiento de los datos recabados. Además para la implementación del sistema se plantearon dos modelos distintos, uno utilizando únicamente el DBMS PostgreSQL (versión simple), y otro utilizando PostgreSQL y HBase versión (híbrida). Se realizaron pruebas de carga sobre ambas versiones, donde los tiempos de respuesta de la versión híbrida se mantenían estables a medida que aumentaban las transacciones por segundo. En cambio los tiempos de respuesta de la versión simple se hacían más largo a medida que aumentaban las transacciones por segundo. El sistema necesita mantener un rendimiento estable para una alta cantidad de transacciones por segundo, por ello se eligió la versión híbrida por sobre la versión simple.

Dentro de este módulo se estableció una REST API por la cual el módulo de Integración puede consultar y almacenar información de mediciones y cálculos de una base de datos. Por medio de esta misma REST API el módulo de sensores puede guardar la información de las mediciones que obtuvo. Esta REST API se encuentra alojada en un servidor de Amazon Web Services. La documentación de su uso se encuentra alojada en el mismo servidor. La REST API maneja tiempos de respuestas menores a los 600 milisegundos para un tráfico de 80 transacciones por segundo. Con esto se da cumplimiento al objetivo de implementar las bases de datos necesarias para almacenar la informacion de consumo energetico de los usuarios de la herramienta.

En la parte de integración se desarrolló una plataforma en la cual se establece la comunicación necesaria entre el módulo de Almacenamiento de Información y Servicios Web y el módulo de Análisis de Datos. Esta comunicación sucede en diferentes momentos: una vez por hora para el proceso de predicción, una vez por día para el proceso de clustering y una vez por mes para el proceso de entrenamiento de predicción. Esta comunicación se puede realizar gracias a que se siguieron los acuerdos encontrados en los anexos C y B, donde se definen los protocolos de comunicación entre los módulos involucrados. Estos acuerdos se establecieron tomando en cuenta las necesidades de cada módulo y se realizaron pruebas con las que se confirma que se siguieron

los acuerdos. Estas pruebas demuestran no solo que se siguen los protocolos, sino que se establece exitosamente la comunicación entre el módulo de Almacenamiento de Información y Servicios Web y el módulo de Análisis de datos. También indican que para realizar la comunicación es necesario tener acceso a internet, ya que sin internet no se pueden consumir los servicios web donde se obtiene la información para los procesos de análisis.

Referente al módulo de ánalisis de datos, se aplicaron algoritmos para predecir el consumo de energía eléctrico, siendo los algoritmos elegidos para este proyecto los conocidos como Árbol de Decisión y Máquina de Vectores de Soporte, ambos en su versión de regresión. Con ello se encontró que el método de Árbol de Decisión fue más rápido que el de Máquina de Vectores de Soporte, con un tiempo de 0.67 segundos y 10.54 segundos, respectivamente. Este método también mostró superioridad en el puntaje de exactitud con 0.9992, mientras que el segundo obtuvo un puntaje de 0.9936. Por lo cual es un mejor cantidato para ser aplicado en el sistema propuesto. Esto con datos tomados desde el sensor del proyecto.

También se ejecutaron algoritmos de análisis de datos para clasificar el consumo de energía, siendo estos KMeasn y Clústeres Jerárquicos, y con esto se encontró que el uso del método de KMeans es un mejor candidato para ser usado en el sistema propuesto, puesto a pesar que es un poco más lento que Clústeres Jerárquicos para pocos datos, dado el tiempo de ejecución teórico, esté será más rápido al tener más datos por lo que es más escalable para las finalidades del sistema, además que los resultados son similares a los de Clústres Jerárquicos, en cuanto a las agrupaciones formadas por ambos métodos.

Dentro del módulo de interfaz de usuario se cumplió con el objetivo de diseñar una interfaz de usuario que cumpliera con las características de usabilidad, utilizando la metolodología de diseño Ëstudio de Usabilidadÿ aplicando un diseño iterativo, centrado en el usuario. Esto no descarta la posibilidad de mejoras y refinamiento de esta característica a través de la evaluación de más estudios de usabilidad.
También se cumplió con el objetivo de mostrar los datos de consumo de energía eléctrica a través de una interfaz útil, seleccionando un tipo de interfaz y plataforma de desarrollo adecuada según las necesidades del proyecto. Esto garantiza flexibilidad, soporte, eficiencia y una estructura modular. Además se definieron las funcionalidades del sistema, cumpliendo con el objetivo del módulo, a través de componentes gráficos que muestran información relevante al usuario. Esta definición está abierta a mejoras utilizando una metodología de diseño centrado en el usuario.

IX. Recomendaciones

Respecto a los sensores y protocolos de información se recomienda que el sensor de voltaje tenía un rango máximo de 600 Vrms por lo que se recomienda buscar un sensor cuyo rango sea más adecuado para los voltajes comunes en hogares y así obtener una mejor digitalización. También el investigar el consumo máximo de potencia de un hogar promedio para implementar un sensor de corriente cuyo rango máximo no esté tan alejado del valor máximo que se medirá, evitanao así que pequeñas variaciones a la salida del sensor resulten como errores grandes en las mediciones.

El módulo de seguridad de la información, para la implementación de un sistema de seguridad de un proyecto, recomienda utilizar algoritmos de cifrado simétrico para el intercambio de información privada y algoritmos de cifrado asimétrico para el intercambio de llaves. Se debe de realizar además verificaciones de fuente de origen para los sistemas y usuarios internos; además de la administración de roles y asilamiento de conexiones utilizando diferentes canales de comunicación como VPN. Para poder implementar algoritmos, se recomienda que se utilicen librerías conocidas de uso libre. Al tener algoritmos libres y siendo utilizadas por la comunidad, este puede brindar más documentación, resolución de problemas y dudas; que un algoritmo no muy común. Si se implementará, se debe de tener en cuenta que esto está más propenso a generar errores y por lo tanto se pueden involucrar más vulnerabilidades por cualquier error de programación o de diseño.

El mismo módulo aconseja que conforme a futuros análisis, se recomienda utilizar diferentes técnicas y metodologías para poder comparar la funcionalidad de los algoritmos. Realizar comparaciónes implementandolas dentro de diferentes sistemas con la misma arquitectura. Utilizar técnicas de comparación, para eliminar los tiempos de uso del procesador del dispositivo, y realizar pruebas de penetración o monitoreo de tráfico para verificar la confidencialidad del algoritmo. Se recomienda también elaborar distintas políticas de administración, eliminación, rotación y generación de llaves criptográficas, y por supuesto; su buena almacenamiento e intercambio. Elaborar mecanismos de seguridad contra ataques principalmente para el servidor; elaborar restricciones de direcciones de IP o direcciones físicas de dispositivos para evitar ataques no inesperados.

Para el almacenamiento de información y servicios web se sugiere el utilizar herramientas de pruebas de rendimiento con clientes distribuidos, puesto que esta acción que no se pudo realizar debido a que es necesario pagar una tarifa de membresía para hacer pruebas con más de 50 usuarios concurrentes. De igual forma se sugiere usar métodos alternativos para almacenamiento de llaves,

dado que actualmente las llaves simétricas para cifrar información durante su transferencia entre el sensor y el servidor están almacenadas de forma persistente lo que supone un riesgo. Se proponen soluciones alternativas como almacenamiento temporal de las llaves en memoria o llaves que expiren luego de una cantidad determinada de tiempo.

Para el módulo de integración se se aconseja el mejorar el manejo de errores, para la comunicación con estos módulos pueden existir errores que no se manejan de la mejor manera. Con el módulo de Almacenamiento de la Información y Servicios Web se recomienda realizar una cantidad mínima de intentos al consumir un servicio web o utilizar mensajería para establecer comunicación. Esto se recomienda gracias a que en base a la investigación, uno de los problemas encontrados con integración remota es lo poco confiable que son las redes de internet. Una cantidad mínima de intentos ayuda a disminuir errores que pueden ocurrir con la conexión a internet. Por otro lado, la mensajería se puede utilizar para publicar el mensaje en un canal y luego enviarlo para consumir un servicio web. El canal conservaría el mensaje hasta que se obtenga una respuesta del servicio web. Esto asegura que se consuma el servicio web, ya que si no da una respuesta, el canal no elimina el mensaje y se intentará volver a enviar hasta que se obtenga una respuesta.

Además que con la utilización de mensajería se debe tomar en cuenta que puede llegar a atrasarse mucho el flujo de información. Esto causaría que se acumulen varios procesos de análisis, lo que puede retrasar de manera significativa los resultados para el usuario. Por esta razón es necesario establecer un limite de mensajes que puede acumular el canal que consume un servicio web. Y con el módulo de Análisis se recomienda no crear archivos porque puede llegar a ser muy lento. Este proceso involucra que el módulo de integración, cree el archivo y el módulo de análisis lo lea, lo cual resulta ineficiente.

De igual modo se aconseja el mejorar comunicación con proceso de clustering, pues los cuadros 43 y 44 demuestran que no siempre se lleva a cabo la comunicación con el proceso de clustering. Para solucionar esto se recomienda discutir con el módulo de Análisis de Datos sobre los requisitos que debe tener la información transmitida para que se logren realizar los análisis. Por el momento se obtiene la información del día, para el análisis de clustering, se formatea según protocolo y se envía, pero se podrían realizar procesos para cumplir con las condiciones mínimas para que este análisis no falle.

Para el módulo de ánalisis de datos, se aconseja que en lo referente a la parte de predicción de consumo de energía de parte de los usuarios, se recomienda el tomar en consideración métodos estadísticos como ARIMA y ARMA que ha sido bastante populares en el pasado y con buenos resultados en la literatura. De igual manera se recomienda el uso de métodos mezclados como Máquina de Vectores de Soporte juntamente con una Red Neuronal para mejorar la exactitud

de las predicciones. Además se recomienda el uso de tecnologías para big data, como el uso de tecnologías basada en clústeres como BOINC o bien el uso de Hadoop para mejorar la velocidad de procesamiento de los datos tanto en las fase de búsqueda de parámetros como también en las de entrenamiento, buscando con esto que sin importar la complejidad de los datos o de los métodos implementados, se obtenga una velocidad aceptablemente rápida para obtener resultados.

Dentro del módulo anterior también se sugiere que para la sección relacionada con clústeres, se recomienda la consideración de otros métodos mencionados en la literatura como Máquina de Vectores de Soporte, puesto que esta puede devolver grupos mejor definidos y mejorar la exposición de patrones, dado que ha mostrado mejores resultados en otras investigaciones. De igual manera se recomienda el uso de datos reales tomados desde un sensor confiable para realizar la selección de algoritmos a ser puestos en el área de producción, dado que esto pudo significar cierto desacierto en la selección de métodos de este trabajo. Así mismo se recomienda la implementación de visualización de datos basada en localidad de los mismos para poder observar de mejor manera los clústeres formados y no solamente los las líneas de tendencia de los diferentes consumos de los usuarios.

En lo que respecta a la parte de interfaz de usuario se recomienda la realización de más iteraciones a través estudios de usabilidad, para el refinamiento de la interfaz y su característica de usabilidad. También se recomienda actualizar la interfaz de usuario por lo menos cada seis meses, tanto en diseño, como en funcionalidades para presentar siempre información relevante para el usuario. Así mismo la aplicación podría integrar en el futuro una conexión con redes sociales para compartir información sobre su consumo a sus contactos. De esta forma, más personas pueden conocer sobre la plataforma y se está brindando una herramienta útil para el usuario. Además, para brindar un servicio más personalizado, se recomienda la inclusión de un sistema de notificaciones periódicas y eventuales, que indique al usuario información relevante sobre su consumo, datos atípicos y límites establecidos.

X. Bibliografía

Abiteboul, S., Hull, R., & Vianu, V. (1995). *Foundations of databases. Computers & Mathematics with Applications.* United States of America: Addison Wesley.

A Comparison Of NoSQL Database Management Systems And Models. (2014).Recuperado de https://www.digitalocean.com/community/tutorials/a-comparison-of-nosql-database-management-systems-and-models

Ahmad, S., Rizwan, M., Ahmad, J., Barua, N. (2010). *Meet in the middle attack: A cryptoanalysis approach*, CiteSeerX. Recuperado el 2 de septiembre de 2016 de: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.184.7025&rep=rep1&type=pdf

Alanazi, H., Zaidan, B., Zaidan, A., Jalab, H., Shabbir, M. and Al-Nabhani, Y. (2016). *New Comparative Study Between DES, 3DES and AES within Nine Factors.* 2nd ed. Recuperado el 1 de abrir de 2016 de: https://arxiv.org/ftp/arxiv/papers/1003/1003.4085.pdf

Alanazi, H., Zaidan, B., Zaidan, A., Jalab, H., Shabbit, M., Al-Nabhani, Y. (2010). *New comparative study between DES, 3DES and AES within Nine Factors*, Journal of Computing. Recuperado el 16 de junio de 2016 de: https://arxiv.org/ftp/arxiv/papers/1003/1003.4085.pdf

Anastasi, G. Corucci, F. Marcelloni, F. (2010) *Intelligent System for Electrical Energy Management in Buildings.* 11th International Conference

Anderson, R., Biham, E., Knudsen, L. (2000). *Serpent: A Proposal for the Advanced Encryption Standard*, CryptoSoft. Recuperado el 13 de junio de 2016 de: http://cryptosoft.net/docs/Serpent.pdf

Antonio, J. (2011). *Rompiendo hashes con findmyhash.py.* Recuperado el 1 de octubre de 2016 de: http://www.flu-project.com/2011/12/rompiendo-hashes-con-findmyhashpy_25.html

Apache HBase. (2016). http://hortonworks.com/apache/hbase/

Ávila, J. (2013). *Confidencialidad de la información*, INNSZ. Recuperado el 26 de febrero de 2016 de: innsz.mx/opencms/contenido/investigacion/comiteEtica/confidencialidadInformacion.html

Bakken. *Middleware.* http://www.eecs.wsu.edu/∼bakken/middleware.pdf [16 de octubre del 2016]

Barker Sean. (2012).*Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes.* http://lass.cs.umass.edu/papers/pdf/sustkdd12-smart.pdf

Biham, E., Dunkelman, O., Keller, N. (2002). *Linear Cryptanalysis of Reduced Round Serpent*, Springer Berlin Heidelberg. Recuperado el 12 de julio de 2016 de: https://www.cosic.esat.kuleuven.be/nessie/reports/phase1/tecwp3-008_1.pdf

Biryukov, A., Kushilevitz, E. (2007). *From Differential Cryptanalysis to Ciphertext-Only Attacks*, CiteSeerX. Recuperado el 3 de septiembre de 2016 de: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.2939&rep=rep1&type=pdf

BlazeMeter . (2016). https://guide.blazemeter.com/hc/en-us

Bogomonly, A. *The Distance Formula.* http://www.cut-the-knot.org/pythagoras/DistanceFormula.shtml [Marzo 25, 2016]

Boneh, D. (1998). *Twenty years of attacks on the RSA cryptosystem*, Standford University. Recuperado el 1 de agosto de 2016 de: https://crypto.stanford.edu/d̃abo/papers/RSA-survey.pdf

Boylestad, R. (2004). *Introducción al Análisis de Circuitos*. México D.F. Pearson Educación. 1228 págs.

Breiman, L., Friedman, J.H., Olshen, R., & Stone, C.J., 1984. *Classification and Regression Tree.* Pacific California: Wadsworth & Brooks/Cole Advanced Books & Software. 358 págs.

Brownlee, J. *A Gentle Introduction to Scikit-Learn: A Python Machine Learning Library.* http://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/ [Agosto 10, 2016]

Brownlee, J. *Feature Selection For Machine Learning in Python.* http://machinelearningmastery.com/feature-selection-machine-learning-python/ [Agosto 10, 2016]

Brownlee, J. *How to Prepare Data For Machine Learning.* http://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/ [Agosto 10, 2016]

Brumley, D., Boneh, D. (2010). *Remote Timing Attacks are Practical*. Recuperado el 12 de octubre de 2016 de: https://crypto.stanford.edu/d̃abo/papers/ssl-timing.pdf

Bruschi, D., Cavallo, L., Lanzi, A., Monga, M. (2010). *Replay Attack in TCG Specification and Solution*, International Secure Systems Lab. Recuperado el 5 de octubre de 2016 de: http://old.iseclab.org/people/andrew/download/acsac05.pdf

Burwick, C., Coppersmith, D., D'Avignon, E., Gennaro, R., Halevi, S., Jutla, C., Matyas, M,. Connor, L., Peyravian, M., Safford, D. y Zunic, N. (1999). *MARS - a candidate cipher for AES*,

IBM. Recuperado el 5 de junio de 2016 de: http://www.nada.kth.se/kurser/kth/2D1449/99-00/mars.pdf

Canga, R. (2012). *Introducción a la Difusin de Seales de Radio y Televisión.* http://serbal.pntic.mec.es/srug0007/archivos/radiocomunicaciones/120INTRODUCCI%D3N/1%20Radiofrecuencia.pdf.pdf [Consultado: 20 de junio 2016]

Center, F. D. and I. D. *Data Analysis.* https://ori.hhs.gov/education/products/n_illinois_u/datamanagement/datopic.html [Marzo 18, 2016]

Chandler, Daniel. *Whats wasting power at home? Ask your app!.* http://news.mit.edu/2016/wasting-power-home-app-0801 [24 de octubre de 2016]

Chandra, Usha. Shi, Jinyu, Chandra, Namas. *Design and Implementation of IBIDS An Internet Based Integrated Design System.* ACM-SE 37 Proceedings of the 37th annual Southeast regional conference

Chang, C. & Lin, C. 2013. *LIBSVM: A Library for Support Vector Machines.* ACM Transactions on Intelligent Systems and Technology (TIST), 2, 139. https://doi.org/10.1145/1961189.1961199

Chicco, G., Napoli, R., & Piglione, F. 2006. *Comparisons among clustering techniques for electricity customer classification.* IEEE Transactions on Power Systems, 21(2), 933940. http://doi.org/10.1109/TPWRS.2006.873122

Christensson, P. (2009, Mar 31). *User Interface Definition.* Recuperado en Mar 31, 2016, de http://techterms.com

Chujai, P., Kerdprasop, N., & Kerdprasop, K. 2013. *Series Analysis of Household Electric Consumption with ARIMA and ARMA Models.* Proceedings of the International MultiConference of Engineers and Computer Scientists, I.

Conrad, E., Misenar, S., Feldman, J. (2012). *CISSP Study Guide*, Waltham, USA.

Corporate News. *S&C Named As Part of the Innovative Team Awarded Department of Energy Grant for Chicago Microgrid Project.* http://es.sandc.com/news/2014/09/23/sc-on-team-awarded-department-of-energy-grant-for-chicago-microgrid-project/ [6 de agosto de 2015]

Díaz, J. *LOS SISTEMAS DE INFORMACIÓN Y LA INTEGRACIÓN DE SUS SISTEMAS DE GESTIÓN NORMALIZADOS.* http://eprints.rclis.org/16116

Daemen, J., Rijmen, V. (2001). *The Design of Rijndael.* Recuperado el 7 de octubre de 2016 de: http://jda.noekeon.org/JDA_VRI_Rijndael_2002.pdf

Dai, Q., Zhang, C. & Wu, H. 2016. *Research of Decision Tree Classification Algorithm in Data Mining*. International Journal of Database Theory and Application. 9(5), 18.

Damgard, I., Nielsen, J. (2001). *From Know-Plaintext Security to Chosen-Plaintext Security*, BRICS. Recuperado el 7 de octubre de 2016 de: http://www.brics.dk/RS/01/43/BRICS-RS-01-43.pdf

Daniel (2015). *Arduino usando SCT013*. http://cms.35g.tw/coding/arduino-using-sct013-measure-current/ [Consultado: 14 de septiembre 2016]

Dayan, P. 2009. *Unsupervised learning.* The MIT Encyclopedia of the Cognitive Sciences, 1 7. http://doi.org/10.1007/BF00993379

DB-Engines Ranking of Relational DBMS (2016). Recuperado de http://db-engines.com/en/ranking/relational+dbms

De Luz, S. (2010). *Criptografía: Algoritmos de cifrado de clave simétrica*, Redes Zone. Recuperado el 5 de marzo de 2016 de: http://www.redeszone.net/2010/11/04/criptografia-algoritmos-de-cifrado-de-clave-simetrica

Demirci, H., Aydin, A., Ture, E. (2004). *A New Meet-in-the-Middle Attack on the IDEA Block Cipher*, Springer Berlin Heidelberg. Recuperado el 11 de julio de 2016 de: http://www.cs.bilkent.edu.tr/šelcuk/publications/IDEA_SAC03.pdf

Desiree, N., Edit, J. (2004). *Seguridad informática y criptografía*, Universidad Nacional de Nordeste. Recuperado el 2 de marzo de 2016 de: http://exa.unne.edu.ar/informatica/SO/Criptografia04.pdf

Development, W. G. for C. H. and. *Collecting and Analyzing Data*. http://ctb.ku.edu/en/table-of-contents/evaluate/evaluate-community-interventions/archival-data/main [Marzo 28, 2016]

deVille, B. 2006. «Decision Trees - What are they?» *Decision Trees for Business Intelligence and Data Mining: Using SAS Enterprise Miner*. Cary, Carolina del Norte: SAS Institute Inc. págs. 1-16.

Douglas, S. *Advanced Encryption Standard*, RIVIER. Recuperado el 3 de mayo de 2016 de: https://www.rivier.edu/journal/ROAJ-Fall-2010/J455-Selent-AES.pdf

Durda, F. (2014). *Serial and UART Tutorial*. https://www.freebsd.org/doc/en/articles/serial-uart/ [Consultado: 22 de junio 2016]

Edwards, R. E., New, J., & Parker, L. E. 2012. *Predicting future hourly residential electrical consumption: A machine learning case study.* Energy and Buildings, 49, 591603. http://doi.org/10.1016/j.enbuild.2012.03.010

Eguiluz Javier. (2015).*New in Symfony 2.8: Symfony as a Microframework* . http://symfony.com/blog/new-in-symfony-2-8-symfony-as-a-microframework

Eichinger, F., & Pathmaperuma, D. 2013. «Analysis Challenges in the Future Energy Domain».*Computational Intelligent Data Analysis for Sustainable Development.* Londres: Chapman and Hall/CRC. págs. 181223.

Eliasson, P., & Rosén, N. 2013. *Efficient K-means clustering and the importance of seeding.* Tesis KTH Royal Institute of Technology. Suiza, Estocolmo: School of Computer Science and Communication. 17 págs.

El modelo Entidad-Relación. (2013). Recuperado de http://www.cs.us.es/cursos/bd-2001/temas/diseno.html

Fabio, M. (2010). *Lecture 4: Data Encryption Standard (DES)*, Laboratorie de Recherche en Informatique. Recuperado el 7 de abril de 2016 de: https://www.lri.fr/ fmartignon/documenti/systemesecurite/4-DES.pdf

Fabio, M. (2010). *Rijndael Algorithm (AES)*, Laboratorie de Recherche en Informatique. Recuperado 6 de mayo de 2016 de: https://www.lri.fr/ fmartignon/documenti/systemesecurite/5-AES.pdf

Farrell, Susan. (2015, Oct 4). *Utility Navigation: What It Is and How to Design It.* Recuperado de https://www.nngroup.com/articles/utility-navigation/

Ferguson, N. (1999). *Impossible differentials in Twofish*, Schneier on Security. Recuperado el 10 de julio de 2016 de: https://www.schneier.com/academic/paperfiles/paper-twofish-impossible.pdf

Fernández Alberto. (2013). *Servicios web RESTful con HTTP. Parte I: Introduccin y bases toricas.* Recuperado de http://www.adwe.es/general/colaboraciones/servicios-web-restful-con-http-parte-i-introduccion-y-bases-teoricas

Flath, C., Nicolay, D., Conte, T., Van Dinther, C., & Filipova-Neumann, L. 2012. *Cluster analysis of smart metering data: An implementation in practice.* Business and Information Systems Engineering, 4(1), 3139. https://doi.org/10.1007/s12599-011-0201-5

Fluhrer, S., Mantin, I., Shamir, A. (2001). *Weaknesses in the Key Scheduling Algorithm of RC4*, Cisco Systems. Recuperado el 29 de junio de 2016 de: http://www.crypto.com/papers/others/rc4_ksaproc.pdf

Gabe-Thomas, E., Walker, I., Verplanken, B., & Shaddick, G. 2016. *Householders mental models of domestic energy consumption: Using a sort-and-cluster method to identify shared concepts of appliance similarity* PLoS ONE, 11(7), 115. https://doi.org/10.1371/journal.pone.0158949

Galaz, J. (2015). *nRF24L01 Control de 2 servos + 1 servo o motor.* http://seta43.blogspot.com/2015/07/nrf24l01-control-de-2-servos-1-servo-o_78.html [Consultado: 5 de septiembre 2016]

Gangan, S. (2015). *A Review of Man-in-the-Middle attack*. Recuperado el 30 de agosto de 2016 de: https://arxiv.org/ftp/arxiv/papers/1504/1504.02115.pdf

Geisler, M. (2013). *Python open built-in function: difference between modes a, a+, w, w+ and r+.* http://stackoverflow.com/questions/1466000/python-open-built-in-function-difference-between-modes-a-a-w-w-and-r [Consultado: 8 de octubre 2016]

Gelbstein, Ed. (2016). *La integridad de los datos: el aspecto más relegado de la seguridad de la información*, ISACA. Recuperado 29 de febrero de 2016 de: http://www.isaca.org/Journal/archives/2011/Volume-6/Pages/Data-Integrity-Information-Securitys-Poor-Relation-spanish.aspx

Germano, T *Self Organizing Maps*. http://davis.wpi.edu/ matt/courses/soms/ [Marzo 20, 2016]

Gilbert, H., Handschuh, H., Joux, A., Vaudenay, S. (2001). *A Statistical Attack on RC6*, Springer Berlin Heidelberg. Recuperado el 4 de julio de 2016 de: https://www-almasty.lip6.fr/j̃oux/pages/papers/RC6Stat.pdf

González, J. (2012). *Cripto. Cifrado de flujo*, Tecnologías de Software. Recuperado 18 de marzo de 2016 de: http://pepgonzalez.blogspot.com/2012/10/cripto-cifrado-de-flujo.html

Google (2013). *Color - Material Design Guidelines* Google Inc. Recuperado de https://material.google.com/style/color.html

Google Trends. (2016, Ago 10). *Compare: angularjs, ember.js, backbone.js* Google Trends (www.google.com/trends). Recuperado de https://www.google.com/trends/explore?date=all&q=ember.js,angularjs,backbone.js&hl=en-US

Gorbatchev Alex. (2014). *Comparing Express, Restify, hapi and LoopBack for building RESTful APIs* . Recuperado de https://strongloop.com/strongblog/compare-express-restify-hapi-loopback/

Gregory, P. (2015). *CISSP Guide to Security Essentials*. Boston, USA.

Grusin, M. (2013). *Serial Peripherial Interface (SPI).* https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi [Consultado: 22 de junio 2016]

Gunasundari, T. Elangovan, K. (2014). *A comparative survey on Symmetric Key Encryption Algorithms*, ISI. Recuperado el 17 de mayo de 2016 de: http://isindexing.com/isi/papers/1393092856.pdf

Gururaja, H., Seetha, M., Koundinya, A., Shashank, A., Prashanth, C. (2014). *Comparative Study and Performance Analysis of Encryption in RSA, ECC and GoldwasserMicali Cryptosystems*, IJAIEM. Recuperado el 5 de agosto de 2016 de: http://www.ijaiem.org/volume3issue1/IJAIEM-2014-01-15-028.pdf

Handschuh, H., Howard, M. (1999). *A Timing Attack on RC5*, Springer Berlin Heidelberg. Recuperado el 30 de junio de 2016 de: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.9908&rep=rep1&type=pdf

Harris, S. (2013). *CISSP All-in-One Exam Guide*. Estados Unidos: McGraw-Hill

Heys, H. (2004). *A Tutorial on Linear and Differential Cryptanalysis*, Memorial University of Newfoundland. Recuperado el 26 de agosto de 2016 de: https://www.engr.mun.ca/~howard/PAPERS/ldc_tutorial.pdf

Heys, M., Tavares, S. (1994). *On the Security of the CAST Encryption Algorithm*, Memorial University. Recuperado el 6 de junio de 2016 de: http://www.engr.mun.ca/ howard/PAPERS/ccece_94.pdf

Hibernate - ORM Overview. (2016). Recuperado de https://www.tutorialspoint.com/hibernate/ormoverview.htm

Hohpe, Gregor & Woolf, Bobby. *Messaging Patterns*. http://www.enterpriseintegrationpatterns.com /patterns/messaging/toc.html [16 de octubre de 2016]

Hohpe, Gregor & Woolf. (2015). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions* Addison-Wesley. 13 - 74.

Huguenard, J. *Error Sum of Squares*. http://hlab.stanford.edu/ brian/error_sum_of_squares.html [Marzo 30, 2016]

Hyndman, R., & Athanasopoulos, G.*Dynamic regression models*. https://www.otexts.org/fpp/9/1 [Marzo 18, 2016]

Hyndman, R., & Athanasopoulos, G. *Moving average models*. https://www.otexts.org/fpp/8/4 [Marzo 18, 2016]

IBM Analytics, *Data Science*. http://www.ibm.com/analytics/us/en/technology/data-science [Julio 30, 2016]

Institute Inc., S. 1999. *Chapter 7 The ARIMA Procedure*. SAS/ETS Users Guide, Version 8, 191299.

Integración de sistemas de información - Conceptos básicos http://solucionesdeprogramacion.blogspot.com/2011/08/integracion-de-sistemas-de-informacion.html [20 de abril de 2016]

Interface Definition Language. http://csis.pace.edu/∼marchese/CS865/Papers/interface-definition-language.pdf [16 de octubre del 2016]

IPA (2003). *Analysis of RC2*, IPA. Recuperado el 10 de mayo de 2016 de: https://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1042_rc2.pdf

ISSY Grid. *IssyGrid, first operational district level smart grid in France.* http://www.issy.com/en/home/issy-a-smart-city/issygrid [6 de agosto de 2015]

Jain, M., Agrawal, A. (2014). *Implementation of hybrid cryptography algorithm*, iJCEM. Recuperado el 12 de junio de 2016 de: http://ijcem.in/wp-content/uploads/2014/06/Implementation-Of-Hybrid-Cryptography-Algorithm.pdf

Jones Don. (2012). *Pros and cons of SQL Server 2012.* Recuperado de http://searchsqlserver.techtarget.com/tip/Pros-and-cons-of-SQL-Server-2012

Junod, P. (2001). *On the Complexity of Matsui's Attack*, Pascal Junod. Recuperado el 22 de junio de 2016 de: http://crypto.junod.info/sac01.pdf

Karthik, S., Muruganandam A. (2014). *Data Encryption and Decryption by Using Triple DES and Performance Analysis of Crypto System*, IJSER. Recuperado el 18 de abril de 2016 de: http://www.ijser.in/archives/v2i11/SjIwMTM0MDM=.pdf

Katsov Ilya. (2012).*Apache JMeter.* https://highlyscalable.wordpress.com/2012/03/01/nosql-data-modeling-techniques/

Kelman, T. *Julia - a Fresh Approach to Numerical Computing and Data Science.* https://www.mapr.com/blog/julia-fresh-approach-numerical-computing-and-data-science [Julio 20, 2016]

Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D. (2000). *Improved Cryptanalysis of Rijndael*, Fast Software Encryption. Recuperado el 26 de junio de 2016 de: https://www.schneier.com/academic/paperfiles/paper-rijndael.pdf

Kelsey, J., Schneier, B., Wagner, D. (1997). *Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X NewDES, RC2, and TEA*, Schneier on Security. Recuperado el 27 de junio de 2016 de: https://www.schneier.com/academic/paperfiles/paper-relatedkey.pdf

Kelsey, J., Schneier B. (2000). *MARS Attacks! Preliminary Cryptanalysis of Reduced-Round MARS Variants*, Schneier on Security. Recuperado el 5 de julio de 2016 de: https://www.schneier.com/academic/paperfiles/paper-mars-attacks.pdf

Kilian, J., Rogaway, P. (2000). *How to Protect DES Against Exhaustive Key Search (An Analysis of DESX)**, UCDAVIS. Recuperado el 24 de agosto de 2016 de: http://web.cs.ucdavis.edu/r̃ogaway/papers/desx.pdf

Kodinariya, T. M., & Makwana, P. R. 2013. *Review on determining number of Cluster in K-Means Clustering*. International Journal of Advance Research in Computer Science and Management Studies, 1(6), 23217782.

Koscielny, C. (2004). *A new approach to the ElGamal Encryption Scheme*, amcs. Recuperado el 15 de agosto de 2016 de: http://www.zbc.uz.zgora.pl/Content/2572/14kosciel.pdf

Krstajic, D., Buturovic, L. J., Leahy, D. E., & Thomas, S. 2014. *Cross-validation pitfalls when selecting and assessing regression and classification models.* Journal of Cheminformatics, 6(1), 115. http://doi.org/10.1186/1758-2946-6-10

Kumar S., Suneetha, C., ChandresekhAR, A. (2012). *Encryption of Data using Elliptic Curve Over Finite Fields*, Cornell University. Recuperado el 6 de agosto de 2016 de: https://arxiv.org/ftp/arxiv/papers/1202/1202.1895.pdf

Lavin, A., & Klabjan, D. 2014. *Clustering Time Series Energy Data from Smart Meters*. Energy Efficiency, 8(4), 681-689.

Lebanidze, E. 2011. *Securing Enterprise Web Applications at the Source: An Application Security Perspective*. Recuperado el 14 de noviembre de 2016 de https://www.owasp.org/images/8/83/Securing_Enterprise_Web_Applications_at_the_Source.pdf

Lichman, M. *UCI Machine Learning Repository*. http://archive.ics.uci.edu/ml [Julio 30, 2016]

Lipmaa, H., Rogaway, P. (2000). *CTR-Mode Encryption*. Recuperado el 17 de mayo de 2016 de: http://web.cs.ucdavis.edu/r̃ogaway/papers/ctr.pdf

Liu. *The Common Object Request Broker Architecture (CORBA)* . http://www.mcs.csueastbay.edu/∼cclee/cs6580/corba-liu.pdf [2 de abril de 2016]

Lucks, S. (1998). *Attacking Triple Encryption*, Serge Vaudenay. Recuperado el 25 de junio de 2016 de: http://dx.doi.org/10.1007/3-540-69710-1_16

Manku, S., Vasanth, K. (2015). *Blowfish encryption algorithm for information security*, ARPN. Recuperado el 7 de junio de 2016 de: http://www.arpnjournals.com/jeas/research_papers/rp_2015/jeas_0615_2157.pdf

Martínez, S., Mateu, V., Tomás, R., Valls, M. 2012. *Criptografía ordenable para bases de datos.* Recuperado el 10 de noviembre de 2016 de: http://gef2012.mondragon.edu/recsi2012/es/programa/recsi2012_submission_62.pdf

MathWorks. *Autoregressive Moving Average Model.* http://www.mathworks.com/help/econ/arma-model.html?requestedDomain=www.mathworks.com [Marzo 18, 2016]

McKenzie, G.*How to Calculate Hamming Distance.* http://classroom.synonym.com/calculate-hamming-distance-2656.html [Abril 7, 2016]

McLaughlin, J. (2015). *Chapter 2 - Differential cryptanalysis.* Recuperado el 27 de agosto de 2016 de: http://vanilla47.com/PDFs/Cryptography/Cryptoanalysis/Chapter%202%20-%20Differential%20cryptanalysis.pdf

Medina, Y. y Miranda, H. (2015). *Comparación de Algoritmos Basados en la Criptografía Simétrica.* Mundo FESC, Edición 9. 14 - 21.

Meier, A. (2005). *The ElGamal Cryptosystem*, TUM. Recuperado el 10 de agosto de 2016 de: http://wwwmayr.in.tum.de/konferenzen/Jass05/courses/1/papers/meier_paper.pdf

Meko, D. *Autoregressive-Moving-Average Modeling.* http://www.ltrr.arizona.edu/dmeko/geos585a.html#cLesson5 [Abril 7, 2016]

Mellon. *Software System Integration* http://www.sei.cmu.edu/productlines/frame_report/softwareSI.htm [12 de abril de 2016]

Menezes, A., Oorschot, P., Vanstone, S. (2001). *Handbook of Applied Cryptography.* Recuperado el 11 de agosto de 2016 de: http://cacr.uwaterloo.ca/hac/about/chap4.pdf

Menezes, A., van Oorschot, P., Vanstone, S. (1996). *Handbook of Applied Cryptography*, CRC Press. Recuperado 17 de marzo de 2016 de: http://cacr.uwaterloo.ca/hac/about/chap7.pdf

Merino, B. (2011). *Análisis de Tráfico con Wireshark*, INTECO. Recuperado el 19 de septiembre de 2016 de incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_analisis_trafico_wireshark.pdf

Michaels Russ. (2014). *MySQL Pros and Cons.* Recuperado de http://www.myhostsupport.com/index.php?/News/NewsItem/View/58

Microchip (2013). *PIC16F87/88 - 18/20/28-Pin Enhanced Flash MCUs with nanoWatt Technology.*

Miles E. (2000). *AES Issues*, CygnaCom. Recuperado el 3 de mayo de 2016 de: http://csrc.nist.gov/archive/aes/round2/comments/20000523-msmid-2.pdf

Misfud, Elvira. (2012). *Introducción a la seguridad informática*, Observatorio Tecnológico: Software. Recuperado el 14 de febrero de 2016 de: http://recursostic.educacion.es/observatorio/web/ca/software/software-general/1040-introduccion-a-la-seguridad-informatica?start=1.

Moore, S. (2010). *Meet-in-the-Middle Attack*. Recuperado el 1 de septiembre de 2016 de: http://stephanemoore.com/pdf/meetinthemiddle.pdf

Morton, J.L. (1999). *Basic Color Theory* Color Matters. Recuperado de http://www.colormatters.com/color-and-design/basic-color-theory

Muñoz, M., Salazar, S., Yang, P. (2014). *Seguridad de la información: ARP Spoofing*, Universidad Técnica Federico Santa María. Recuperado el 22 de septiembre de 2016 de: http://profesores.elo.utfsm.cl/ agv/elo322/1s14/projects/reports/G13/InformeARP.pdf

Mulesoft Inc. *What is Mule ESB?*. https://www.mulesoft.com/resources/esb/what-mule-esb [7 de junio de 2016]

Murtagh, F., & Legendre, P. 2011. *Wards Hierarchical Clustering Method: Clustering Criterion and Agglomerative Algorithm.* arXiv Preprint arXiv:1111.6285, (Junio), 20. http://doi.org/10.1007/s00357-014-9161-z

National Instruments (2006). *Comunicación Serial: Conceptos Generales.* http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1 [Consultado: 22 de junio 2016]

Nechvatal, J., Barker, E., Dodson, D., Dworkin, M., Foti, J., Roback, E. (1999). *STATUS REPORT ON THE FIRST ROUND OF THE DEVELOPMENT OF THE ADVANCED ENCRYPTION STANDARD*. Recuperado el 13 de julio de 2016 de: http://www.famaf.unc.edu.ar/p̃enazzi/nechvatal99statusAESfirstround.pdf

Nechvatal, J., Barker, E., Dodson, D., Dworkin, M., Foti, J., Roback, E. (2000). *Report on the Development of the Advanced Encryption Standard (AES)*. Recuperado el 20 de julio de 2016 de: http://csrc.nist.gov/archive/aes/round2/r2report.pdf

Nicolau, A. *The 9 Best Languages For Crunching Data.* https://www.fastcompany.com/3030716/the-9-best-languages-for-crunching-data [Julio 20, 2016]

Nielsen, Jakob. (2012, Ene 4). *Usability 101: Introduction to Usability.* Recuperado de https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Nizar, a H., Member, A., Dong, Z. Y., Zhao, J. H., & Member, S. 2006. *Electricity Deregulated Market.* Power Engineering Society General Meeting, 2006. IEEE, 17. http://doi.org/10.1109/PES.2006.1709335

Nocedal, J. (2006). *RF Jamming. Capítulo 1: Conceptos de Radiofrecuencia*. Tesis Universidad de las Américas Puebla. Puebla, México. págs. 4-5.

Norman, D., Nielsen, J. *The Definition of User Experience.* Recuperado en Abr 1, 2016, de https://www.nngroup.com/articles/definition-user-experience/

Noura, A. (2015). *A comparison of the 3DES and AES encryption standards*, SERCSC. Recuperado el 28 de abril de 2016 de: http://www.sersc.org/journals/IJSIA/vol9_no7_2015/21.pdf

Nurseitov *et al. Comparison of JSON and XML DAta Interchange Formats: A Case Study*

Real Academia Espñola. *Integrar*. http://dle.rae.es/?id=LqKFoJI [16 de octubre del 2016]

Open Source Load Testing Tools: Which One Should You Use?. (2015). https://www.blazemeter.com/blog/open-source-load-testing-tools-which-one-should-you-use

Pérez, I. (2014). *Cómo funciona ARPspoof?*. Recuperado el 23 de septiembre de 2016 de: http://www.welivesecurity.com/la-es/2014/02/11/como-funciona-arpspoof/

Papazoglou. *SOAP: Simple Object Access Protocol*. http://www.cs.colorado.edu/~kena/classes/7818/f08/lectures/lecture_3_soap.pdf [12 de abril de 2016]

Parker Clayton. (2013). *Python Web Frameworks Compared*. Recuperado de http://www.sixfeetup.com/blog/4-python-web-frameworks-compared

Pedregosa, F. *et al*. 2012. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 28252830. https://doi.org/10.1007/s13398-014-0173-7.2

Peng, W., Chen, J., & Zhou, H. 2009. *An Implementation of IDE3 Decision Tree Learning Algorithm.* University of New South Wales, School of Computer Science & Engineering 1, 120.

Perone, C. *Machine Learning: Cosine Similarity for Vector Space Models (Part III).* http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/ [Abril 5, 2016]

Petrovic, S. 2006. *A Comparison Between the Silhouette Index and the Davies-Bouldin Index in Labelling IDS Clusters.* The 11th Nordic Workshop on Secure IT-Systems, NORDSEC 2006, 5364.

Pham, D. T., Dimov, S. S., & Nguyen, C. D. 2004. *Selection of K in K-means clustering*. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 219(1), 103119. https://doi.org/10.1243/095440605X8298

Phanouriou Constantinos. (1995).*Distributed DBMS model*. http://courses.cs.vt.edu/ cs5204/fall00/distributedDBMS

Pieterse, V., & Black, P. *Hamming distance*. https://xlinux.nist.gov/dads/HTML/HammingDistance.html [Marzo 30, 2016]

Pieterse, V., & Black, P.*Manhattan Distance*. https://xlinux.nist.gov/dads/HTML/manhattanDistance.html [Marzo 30, 2016]

Pivotal. *Spring*. https://spring.io/[7 de junio de 2016]

Pivotal. *Spring Integration*. https://projects.spring.io/spring-integration/ [6 de junio de 2016]

Plattner, H. *An Introduction to Design Thinking PROCESS GUIDE*. https://dschool.stanford.edu/sandbox/groups/designresources/wiki/36873/attachments/74b3d/ModeGuideBOOTCAMP2010L.pdf?sessionID=9435c2b6ec2fd3386cee3ca7946c8a5290fb90bb [17 de octubre de 2016]

Plumley, S. (2004). *Home Networking Bible*. Nueva Jersey: John Wiley & Sons. 850 págs.

Prahastono, I., King, D., & zveren, C. S. 2007. *A review of electricity load profile classification methods.* Proceedings of the Universities Power Engineering Conference, (Abril), 11871191.

Purpura, S. *Introduction to Applied Supervised Learning* http://faculty.washington.edu/jwilker/tft/UWTextToolsConferencePurpura.pdf [Marzo 30, 2016].

Quin, Liam. *XML Essentials*. https://www.w3.org/standards/xml/core [05 de Octubre del 2016]

Quisquarter, J., Standaert, F. (1998). *Exhaustive Key Search of the DES: Updates and Refinements*, UCL Crypto Group. Recuperado el 20 de agosto de 2016 de: http://www.hyperelliptic.org/tanja/SHARCS/talks/JJQ_FXS.pdf

Proyectos Robóticos, (2005). *16F876 & nRF24L01 Robótica*. https://sites.google.com/site/proyectosroboticos/nrf24l01/16f876- nrf24l01 [Consultado: 8 de septiembre 2016].

Raymond, Erick S., Landley, R. (2004). *The Art of Unix Usability.* Chapter 2. History: A Brief History of User Interfaces. Recuperado de http://www.catb.org/ esr/writings/taouu/html/ch02.html.

Razzouk & Shute. *What is Design Thinking and Why Is it Important?*. http://myweb.fsu.edu/vshute/pdf/designthinking.pdf [17 de octubre de 2016]

Real Academia Española. *Red*. http://dle.rae.es/?id=VXs6SD8 [12 de abril de 2016]

Riggio, R., Sicari, S. 2015. *Secure Aggregation in Hybrid Mesh/Sensor Networks*. Recuperado el 09 de noviembre de 2016 de: https://pdfs.semanticscholar.org/b93e/5d501dc2d8c1175d3b66fce957879ccbc9ed.pdf

Rijmen, V. (1997) *Cryptoanalysis and Design of Iterated Block Ciphers*, K.U.Leuven. Recuperado el 8 de julio de 2016 de: https://www.cosic.esat.kuleuven.be/publications/thesis-4.ps

Riveiro, M., Johansson, R., & Karlsson, A. 2011. *Modeling and analysis of energy data: state-of-the-art and practical results from an application scenario.* Tech. Rep. HS-IKI-TR-11-002, 2011.

Rivest, R., Robshaw, J., Sidney, R., Yin, Y., (1998). *The RC6 Block Cipher*, CSAIL. Recuperado el 22 de mayo de 2016 de: http://people.csail.mit.edu/rivest/Rc6.pdf

Rivest, R., Robshaw, J., Sidney, R., Yin, Y., (1998). *The Security of the RC6 Block Cipher*, CSAIL. Recuperado el 30 de mayo de 2016 de: https://people.csail.mit.edu/rivest/ContiniRivestRobshawYinTheSecurityOfTheRC6BlockCipher.pdf

Rivest, R., Shamir, A., Adleman, L. (2001). *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Recuperado el 5 de octubre de 2016 de: http://people.csail.mit.edu/rivest/Rsapaper.pdf

Robot Electronics (1999). *I2C tutorial*. http://www.robot-electronics.co.uk/i2c-tutorial [Consultado: 22 de junio 2016]

Rokach, L. & Maimon, O. 2010. «Clustering methods». *Data Mining and Knowledge Discovery Handbook*. Nueva York: Springer US. págs. 321-352.

Romer, M. *Autoregressive Models*. https://onlinecourses.science.psu.edu/stat510/node/79 [Marzo 18, 2016]

Romer, M.*Moving Average Models (MA models)*. https://onlinecourses.science.psu.edu/stat510/node/79 [Marzo 18, 2016]

Ruggieri, S. 2002. *Efficient C4. 5 [classification algorithm]*. Knowledge and Data Engineering, IEEE Transactions on, 14(2), 438444.

Sage, A & Rouse, W. 1999. *Handbook of Systems Engineering and Managment*. illustrated. Estados Unidos, Michigan. 1256 págs.

Sakal, Marton. (2009, Abr 24). *GUI vs. WUI Through the Prism of Characteristics and Postures.* Management Information Systems, Vol. 5. Recuperado de http://www.ef.uns.ac.rs/mis/archive-pdf/2010%20-%20No1/MIS2010_1_1.pdf

Sasi, S., Dixon, D., Wilson, J. (2014). A General Comparison of Symmetric and Asymmetric Cryptosystems for WSNs and an Overview of Location Based Encryption Technique for Improving Security, IOSR. Recuperado el 21 de marzo de 2016 de: http://www.iosrjen.org/Papers/vol4_issue3%20(part-3)/A04330104.pdf

Schapire, R. *Theoretical Machine Learning.* http://www.cs.princeton.edu/courses/archive/spr08/cos511/scribe_notes/0204.pdf [Marzo 20, 2016]

Schneider, J. *Cross Validation.* https://www.cs.cmu.edu/ schneide/tut5/node42.html [Marzo 18, 2016]

Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., Ferguson, N. (1998). *Twofish: A 128-bit block cipher*, Schneier. Recuperado el 8 de junio de 2016 de: https://www.schneier.com/academic/paperfiles/paper-twofish-paper.pdf

Schneier, B., Whiting, D. (2000). *A Performance Comparison of the Five AES Finalist*, Schneider on Security. Recuperado el 25 de julio de 2016 de: https://www.schneier.com/academic/paperfiles/paper-aes-comparison.pdf

Schramm, K., Leander, G., Felke, P., Paar, C. (2004). *A Collision-Attack on AES.* Recuperado el 17 de septiembre de 2016 de: https://www.iacr.org/archive/ches2004/31560162/31560162.pdf

Shafranovich, Y. *Common Format and MIME Type for Comma-Separated Values (CSV) Files.* https://tools.ietf.org/html/rfc4180 [05 de octubre del 2016]

Shaked, Uri. (2013). *AngularJS vs Backbone.js vs Ember.js.* Airpair. Recuperado de https://www.airpair.com/js/javascript-framework-comparison

Shamir, A., Zinger, E. (2012). *Practical Polynomial Time Known Plaintext Attacks on a Stream Cipher Proposed by John Nash.* Recuperado el 4 de septiembre de 2016 de: https://eprint.iacr.org/2012/339.pdf

Shan, Paul. (2013, Nov 12). *Why AngularJS is generally better in Angular vs Ember vs Backbone.* Void Canvas. Recuperado de http://voidcanvas.com/why-angularjs-is-generally-better-than-emberjs-and-backbonejs/

Sheetal C., Sandeep S. (2014). *A comparative study o Rivest Cipher Algorithms*, RIP. Recuperado el 11 de mayo de 2016 de: http://www.ripublication.com/irph/ijict_spl/ijictv4n17spl_13.pdf

Shetty, A., Shetty, S., Krithika, K. (2014). *A Review on Asymmetric Cryptography - RSA and ElGamal Algorithm*, IJIRCCE. Recuperado el 3 de agosto de 2016 de: http://ijircce.com/upload/2014/sacaim/13_Paper%208.pdf

Singh, S., Garg, A., Sachdeva, A. (2013). *Comparison of Cryptographic Algorithms: ECC & RSA*, IJSCE. Recuperado el 6 de agosto de 2016 de: http://static.ijcsce.org/wp-content/uploads/2013/06/IJCSCESI045113.pdf

Slonneger. *XML-RPC.* http://homepage.cs.uiowa.edu/∼slonnegr/xml/10.XML-RPC.pdf [10 de abril de 2016]

Smart, N., Rijmen, V., Gierlichs, B., Paterson, K., Stam, M., Warinschi, B., Watson, G. (2014). *Algorithms, key size and parameters report, 2014.* Recuperado el 10 de octubre de 2016 de: https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014/at_download/fullReport

Souppaya, M., Scarfone, K. 2016. *Guidelines for Managing the Security of Mobile Devices in the Enterprise.* Recuperado el 16 de noviembre de 2016 de http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-124r1.pdf

Souza, C. *Kernel Functions for Machine Learning Applications.* http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/ [Agosto 15, 2016]

Stevens Sander. (2010).*Apache JMeter.* http://www.methodsandtools.com/tools/tools.php?jmeter

Structured Query Language (2016). Recuperado de https://msdn.microsoft.com/en-gb/library/windows/desktop/ms714670%28v=vs.85%29.aspx

Susan Landau, Find Me a Hash Notices of the American Mathematical Society, 53(3), March. 2006, 330 332

Sykes, A. 2000. «An Introduction to Regression Analysis». *Chicago Lectures in Law and Economics.* Nueva York: Eric A. Posner. págs. 133.

System Properties Comparison MySQL vs. Oracle NoSQL vs. PostgreSQL. (2016).Recuperado de http://db-engines.com/en/system/MySQL%3BOracle+NoSQL%3BPostgreSQL

Talens-Oliag, Sergio. (1999). *Seguridad en JAVA*, Universidad de Valencia. Recuperado el 17 de marzo de 2016 de: http://www.uv.es/sto/cursos/seguridad.java/html/sjava-12.html

TC 9241-210:2010. (2010). *Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems.* Recuperado de http://www.iso.org/iso/catalogue_detail.htm?csnumber=52075

Teknomo, K. (2015). *Mahalanobis Distance.* http://people.revoledu.com/kardi/tutorial/Similarity/MahalanobisDistance.html [Marzo 25, 2016]

The Apache Software Foundation. *ActiveMQ*. http://activemq.apache.org/ [05 de octubre del 2016]

The Apache Software Foundation. *Apache Camel*. http://camel.apache.org/ [7 de junio de 2016]

The National Academy of Sciences *How We Use Energy?*. http://needtoknow.nas.edu/energy/energy-use/ [17 de octubre de 2016]

Tibshinari, R. *Clustering 2: Hierarchical Clustering.* http://www.stat.cmu.edu/ ryantibs/datamining/ lectures/05-clus2-marked.pdf [Agosto 15, 2016]

Top 8 Java RESTful Micro Frameworks. (2015).Recuperado de http://www.gajotres.net/best-available-java-restful-micro-frameworks/2/

Tsiounis, Y., Young, M. (2001). *On the Security of ElGamal Based Encryption*, CiteSeerX. Recuperado el 13 de agosto de 2016 de: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.367.681&rep=rep1&type=pdf

Tyson, J. (2008). *SPP/EPP/ECP How Parallel Ports Work*. http://computer.howstuffworks.com/parallel-port2.htm [Consultado: 22 de junio 2016]

Understanding REST And RPC For HTTP APIs. (2016). https://www.smashingmagazine.com/2016/09/understanding-rest-and-rpc-for-http-apis/

Universidad de Sevilla. *Patrones de Integración*. https://www.lsi.us.es/docencia/get.php?id=6717 [16 de octubre del 2016]

University of California Berkeley. *Why Integrate Systems*. http://integration-services.berkeley.edu/integrating-systems/why-integrate-systems [24 de octubre de 2016]

Urserey. *Why Design Thinking Should Be At The Core Of Your Business Strategy Development*. http://www.forbes.com/sites/lawtonursrey/2014/06/04/14-design-thinking-esque-tips-some-approaches-to-problem-solving-work-better-than-others/#5dee78174528 [25 de octubre de 2016]

Use Cases for Choosing Your Next NoSQL Database. (2011).Recuperado de http://highscalability.com/blog/2011/6/20/35-use-cases-for-choosing-your-next-nosql-database.html

Valentino, V. (2013). *Kali Linux Man In The Middle Attack*. Recuperado el 24 de septiembre de 2016 de: http://www.hacking-tutorial.com/hacking-tutorial/kali-linux-man-middle-attack/

Van den Bossche, F., Wets, J.,& Brijs, T. & Brijs, T. 2004. *A Regression Model with ARIMA Errors to Investigate the Frequency and Severity of Road Traffic Accidents.* Proceedings of the 83rd annual Meeting of the Transportation research Board.

Wagner, D. (1999). *The boomerang attack*. Recuperado el 15 de septiembre de 2016 de: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.3427&rep=rep1&type=pdf

Vásquez Vargas, Julio Ismael (2016). entrevistado por: Cristian Pinelo, 22 de septiembre 2016. Potencia Alterna Monofásica, Universidad del Valle de Guatemala, 18 Av. Zona 15, Vista Hermosa III.

Villamendy. *Masdar Showcases the Concept of the Smart City.* http://www.atelier.net/en/trends/articles/masdar-showcases-concept-smart-city_432033 [6 de agosto de 2015]

Wang, M., Wang, X., Hu, C. (2009). *New Linear Cryptoanalytic Results of Reduced-Round of CAST-128 and CAST-256*, Springer Berlin Heidelberg. p: 429-441.

Web Design. (2012, Abr 15). *The 10 principles of mobile interface design.* Creative Bloq. Recuperado de http://www.creativebloq.com/mobile/10-principles-mobile-interface-design-4122910

What is an API?. (2015). https://developer.ibm.com/apiconnect/docs/what-is-an-api/

What Is Apache Hadoop?. (2016). http://hadoop.apache.org/

White, J. *Julias Role in Data Science.* https://www.oreilly.com/ideas/julias-role-in-data-science [Julio 20, 2016]

Wilson, B. *Induction of Decision Trees.* http://www.cse.unsw.edu.au/ billw/cs9414/notes/ml/06prop/id3/id3.html [Agosto 30, 2016]

World Bank. *World Development Indicators.* http://databank.worldbank.org/data/reports.aspx?source=world-development-indicators [Julio 30, 2016]

YHDC (2013). *100A Split core current transformer SCT 013.* http://www.yhdc.com/en/product/320/ [Consultado: 14 de septiembre 2016]

Zhang, J. *660V din rail ac voltaje transmitter*. https://yhdc.en.alibaba.com/product/60317950605-213988356/660V_din_rail_ac_voltage_transmitter.html [Consultado: 13 de septiembre 2016]

Zhou, Y., Feng, D. (2005). *Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing*. Recuperado el 10 de octubre de 2016 de: https://eprint.iacr.org/2005/388.pdf

s.f. (2011). *Conexiones Protocolo Serie/Paralelo*. https://bloggalex.wordpress.com/2011/12/15/conexiones-protocolo-serieparalelo/ [Consultado: 20 de junio 2016]

XI. Anexos

A. Figuras patrones de integración

Canal del mensaje :

Figura 117: Canal del mensaje

Componente :

Figura 118: Componente

Extremo de mensajería (Endpoint) :

Figura 119: Endpoint del mensaje

Mensaje :

Figura 120: Mensaje

Router :

Figura 121: Router

Traductor :

Figura 122: Traductor

Conector :

Figura 123: Conector

B. Acuerdo almacenamiento de la información y servicios web

El megaproyecto de Sistema de Monitoreo de energía eléctrica para hogares, el módulo de Integración tiene como objetivo comunicarse con el módulo de Almacenamiento de Información y Servicios Web. En el siguiente documento se establece el protocolo en que esta comunicación se va a establecer. El módulo de integración tendrá comunicación con este módulo para obtener información de los usuarios y sus consumos de energía. Este proceso de obtención de información se ejecutará antes de los procesos de análisis de datos, ya que se necesitan los datos de los usuarios para estos procesos. El módulo de integración constará de los siguientes procesos para la obtención de datos:

Solicitar información de consumo de sensores Este proceso realizará una solicitud al servidor para solicitar la información de los usuarios. La solicitud se realizará de la siguiente manera:

- Url: /measurements?filter=:filter
  - "filter": filtro para solicitar datos, JSON con los siguientes atributos
- Método: GET

Filter es un JSON con la siguiente estructura:

- "order": Atributo por el cual ordenar los resultados
- "where": Objecto sobre el cual filtrar los resultados
  - "and": Lista de parámetros de filtros
    - "timestamp": Filtro para timestamp, en este caso puede ser gte o lt.
      - "gte": Filtro para condición mayor o igual que
      - "lt": Filtro para condición menor que

Lo parámetros de esta solicitud son parte de la url. Ver ejemplo 1.
La respuesta es un JSON con la siguiente información:

- Una lista:
  - "value"(decimal): potencia del consumo
  - "latitude"(decimal): latitud donde se realizó el consumo
  - "longitude"(decimal): longitud donde se realizó el consumo
  - "timestamp"(cadena): fecha cuando se realizó el consumo

- "id"(entero): identificador del consumo
- "sensorId"(entero): identificador del sensor que mide el consumo

Ver ejemplo 2.

Solicitar información de sensores Esta solicitud no contiene parámetros y se hace para obtener la información de los sensores. La solicitud se realizará de la siguiente manera.

- Url: /sensors
- Método: GET

La respuesta de esta solicitud se espera un JSON con la siguiente información.

- "name" (cadena): nombre del sensor
- "description" (cadena): descripción del sensor
- "id" (entero): identificador del sensor
- "clientId" (entero): identificador del usuario al que pertenece el sensor

Ver ejemplo 3.

Guardar resultado de análisis Esta solicitud se realizará para los resultados de análisis que se obtenga, a excepción del entrenamiento de análisis que no se tiene que guardar. La solicitud tendrá el siguiente formato:

- Url: /calculations
- Método: POST

En esta solicitud espera un JSON con los parámetros. Este JSON tiene el siguiente formato:

- "name"(cadena): nombre de lo que se va a guardar
- "sensorId"(entero): identificador del sensor
- "value"(decimal): resultado del análisis
- "date"(cadena): Fecha a la que se esta prediciendo, o para el clustering la fecha del día que se tomó en cuenta para el clustering
- "type"(cadena): Tipo de análisis del cual se obtuvo el resultado

Ver ejemplo 4

Como respuesta de esta solicitud se esepera un JSON con los siguientes atributos:

- "id"(entero): Identificador de lo que se envió para guardar

- “name”(cadena): nombre de lo que se va a guardar
- “sensorId”(entero): identificador del sensor
- “value”(decimal): resultado del análisis
- “date”(cadena): Fecha a la que se esta prediciendo
- “type”(cadena): Tipo de análisis del cual se obtuvo el resultado

Ver ejemplo 5

Ejemplos

a. Ejemplo 01 - Solicitar información de consumo de sensores :


```
{
  “order”:“sensorId”,
  “where”:{
    “and”:[
      {
        “timestamp”:{
          “gte”:“2016-10-12”
        }
      },
      {
        “timestamp”:{
          “lte”:“2016-10-13”
        }
      }
    ]
  }
}
```


b. Ejemplo 02 - Información de consumo de sensores :


```
  [
  {
    “value”: 10,
    “latitude”: 0,
    “longitude”: 0,
    “timestamp”: “2016-10-07T03:24:00.000Z”,
    “id”: 0,
```

```
    “sensorId”: 1
  },
  {
    “value”: 40,
    “latitude”: 0,
    “longitude”: 0,
    “timestamp”: “2016-10-09T03:24:00.000Z”,
    “id”: 1,
    “sensorId”: 1
  }
]
```


c. Ejemplo 03 - Información sensores :


```
[
  {
    “name”:“sensor1”,
    “description”:“GT”,
    “id”:1
    “clientId”:1
  },
  {
    “name”:“sensor2”,
    “description”:“otro”,
    “id”:2
    “clientId”:1
  },
]
```


d. Ejemplo 04 - Solicitar guardar resultado de análisis :


```
{
  “name”:“”,
  “value”:50.69,
  “date”:“2016-08-30 00:01”,
  “type”:“prediction”,
  “sensorId”:1
}
```

e. Ejemplo 05 - Resultado de guardar análisis

{
"id": 1,
"name":"",
"value":50.69,
"date":"2016-08-30 00:01",
"type":"prediction",
"sensorId":1
}

C. Acuerdo análisis de datos

En el megaproyecto de Sistema de Monitoreo de energía eléctrica para hogares, el módulo de integración tiene como objetivo comunicarse con el módulo de análisis de datos. En el siguiente documento se establece el protocolo en que esta comunicación se va a establecer. El módulo de integración tendrá comunicación con dos procesos del módulo de análisis de datos: agrupamiento de datos y predicción. El proceso de agrupamiento de datos se ejecutará una vez por día. El proceso de predicción se divide en dos partes: entrenamiento y predicción. El entrenamiento se ejecutará una vez por mes, mientras que la predicción se ejecutará una vez cada hora. El módulo de integración se necarga de solicitar los datos necesarios para ejecutar los procesos del módulo de análisis, este proveé una respuesta que el módulo de integración guardará.

Agrupación Para el proceso de agrupación de datos el módulo de análisis de datos necesita los datos de consumo de energía en un archivo CSV y el nombre de dicho archivo. Con estos dos componentes ya se ejecuta el proceso de este módulo y este proceso devuelve cada dato del CSV con una etiqueta que sería la agrupación a la que pertenece este dato. El CSV que recibe de entrada tiene el siguiente formato:

- La primera fila contiene el encabezado de los datos, estas son sus columnas:
    1. “datetime” (cadena): fecha en la cual se realizó el consumo
    2. “id” (entero): identificador de la medición de consumo
    3. “latitude” (decimal): latitud de donde se realizó la medida
    4. “longitud” (decimal): longitud de donde se realizó la medida
    5. “power” (decimal): potencia consumida
- De la segunda fila en adelante, cada fila contiene los valores de cada consumo

Se creará un archivo CSV que contiene todo el consumo en un día de todos los sensores. Ver ejemplo 1.
Para la respuesta, debido a que es más de un dato, el módulo de integración espera de respuesta un JSON con el siguiente formato (ver ejemplo 2):

- Una lista con los ids y sus resultados
    1. Por cada id en el CSV se espera lo siguiente:
        - “id” (entero): Identificador del sensor de consumo
        - “label” (entero): Etiqueta a la que pertenece el dato

Esta respuesta se espera como una cadena para que el módulo de integración tenga más control sobre la respuesta.

Entrenamiento de predicción Para el entrenamiento de la predicción, el módulo de análisis de datos espera de entrada un CSV con el siguiente formato:

- La primera fila contiene el encabezado de los datos
    1. "datetime" (cadena): Fecha del consumo medido
    2. "id" (entero): Identificador del sensor
    3. "power" (decimal): Potencia del consumo
- La segunda fila en adelante contiene los valores de cada consumo

Este formato está diseñado para realizar el entrenamiento de predicción de un sensor a la vez. Ver ejemplo 3.
La respuesta de este análisis tiene un formato similar que la respuesta del análisis de agrupamiento. En este caso se espera un JSON con los siguientes atributos:

- "id" (entero): Identificador del sensor de consumo
- "result" (boleano): Si fue exitoso o no el entrenamiento

Ver ejemplo 4.

Predicción Para la predicción el módulo de análisis necesita un CSV con el siguiente formato:

- En la primera fila se tiene el encabezado de los datos:
    1. "datetime" (cadena): fecha a la cual se desea la predicción
    2. "id" (entero): Identificador del sensor de consumo
- En la segunda fila en adelante se tienen los valore

Ver ejemplo 5.
Para la respuesta se espera un JSON con los siguientes atributos:

- "date" (cadena): La fecha a la que se predijo
- "id" (entero): Identificador del sensor
- "prediction" (decimal): Consumo que se predice que el sensor va a tener

Ver ejemplo 6

Para todas las respuestas de análisis se espera un JSON como cadena de carácteres.

Ejemplos

a. Ejemplo 1 - Muestra de CSV Agrupación :

‘datetime”,“id”,“latitude”,“longitude”,“‘power”

“08/16/2016 05:47”,“1”,“70.35”,“80.68”,“42.2”

“08/16/2016 05:47”,“1”,“70.35”,“80.68”,“42.2”

b. Ejemplo 2 - Muestra de respuesta JSON, agrupación :

```
[
  {
    “id”: 1,
    “label”: 36
  },
  {
    “id”: 2,
    “label”: 58
  },
  {
    “id”: 3,
    “label”: 90
  }
]
```

c. Ejemplo 3 - Muestra CSV entrenamiento predicción :

“datetime”,“id”,“power”

“19/09/2016 04:22:53”,“1”,“0.9”

“20/09/2016 05:42:13”,“1”,“2.3”

d. Ejemplo 4 - Muestra de JSON de respuesta entrenamiento de predicción :

```
{
  “id”: 1,
  “result”: true
}
```

e. Ejemplo 5 - Muestra de CSV de predicción :

“datetime”,“id”

“08/16/2016 05:00”,“1”

f. Ejemplo 6 - Muestra de JSON de respuesta de predicción :

{
“date”: “08/16/2016 05:00”,
“id”: 1,
“prediction”: 50.69
}

D. Interfaces de competidores

Figura 124: Interfaz de competidor. Schneider-electric embedido

Figura 125: Interfaz de competidor. Schneider-electric 2 embedido

Figura 126: Interfaz de competidor. Efergy

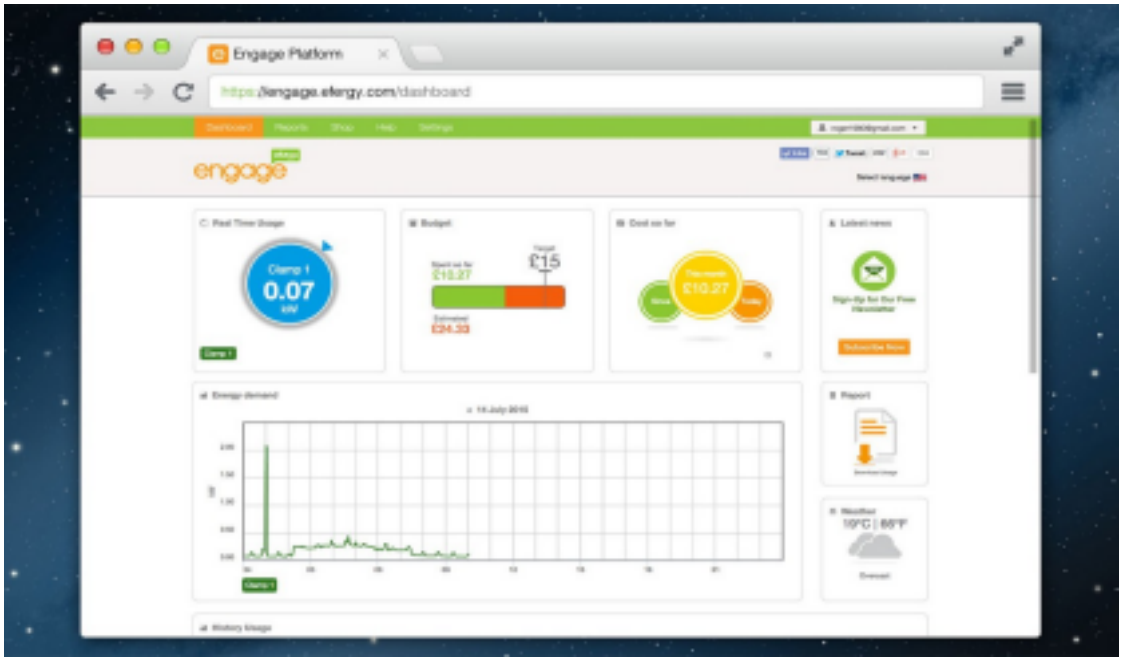


Figura 127: Interfaz de competidor. Efergy 2

Figura 128: Interfaz de competidor. Efergy embedido

E. Bosquejos en papel

Figura 129: Diseño de intefaz. Dashboard

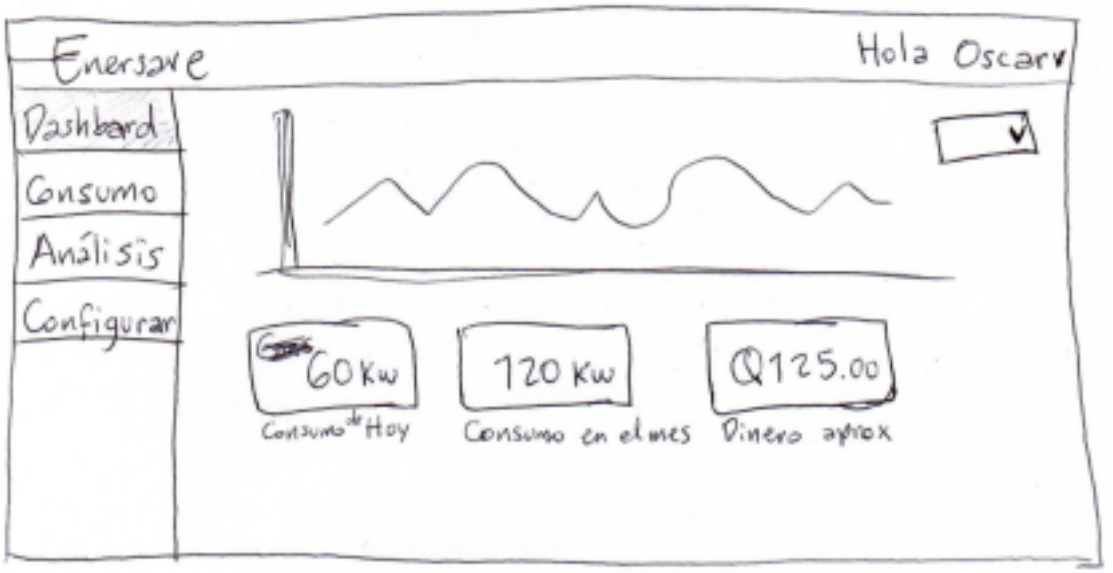


Figura 130: Diseño de intefaz. Consumo

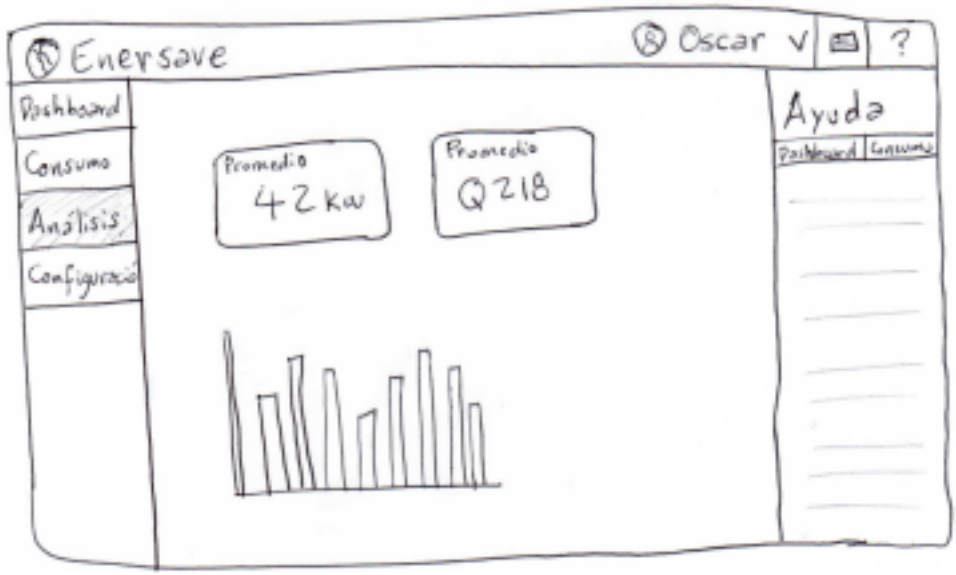


Figura 131: Diseño de intefaz. Análisis estadístico

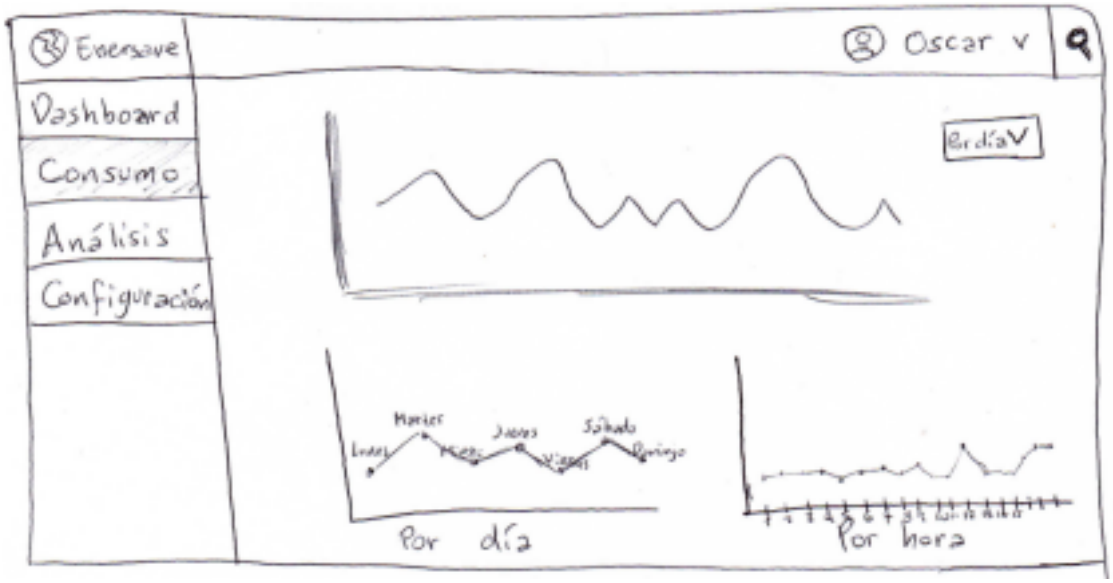


Figura 132: Diseño de intefaz. Configuración

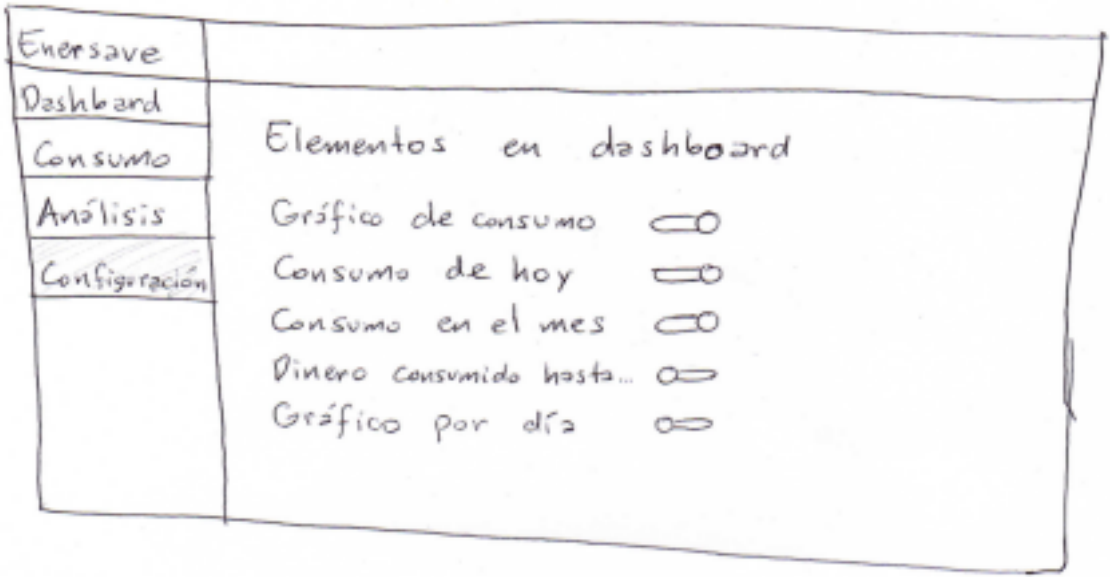


Figura 133: Diseño de intefaz. Análisis predictivo

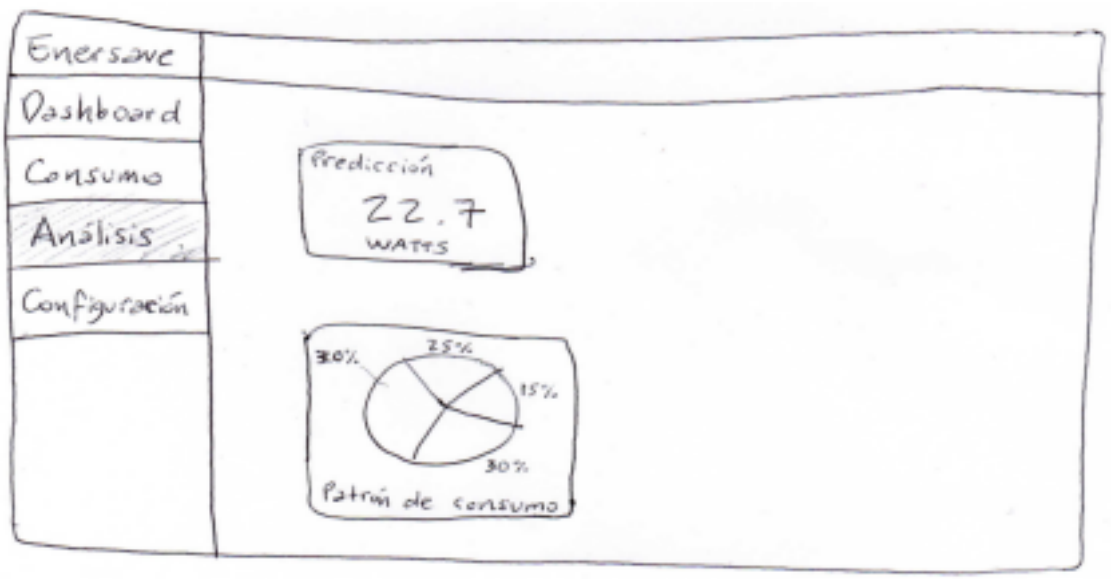


Figura 134: Diseño de intefaz. Consumo 2

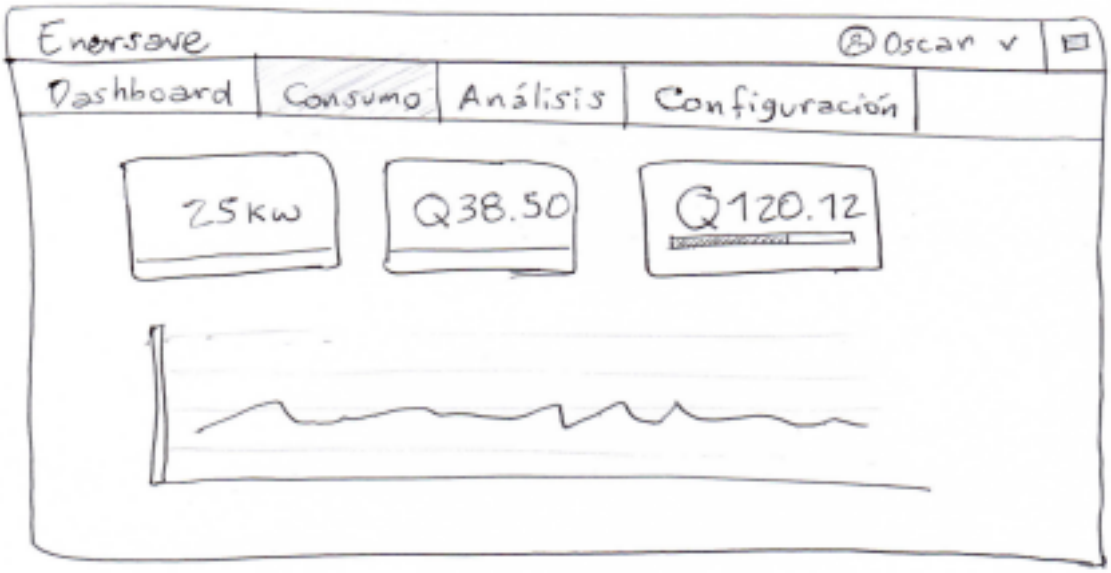
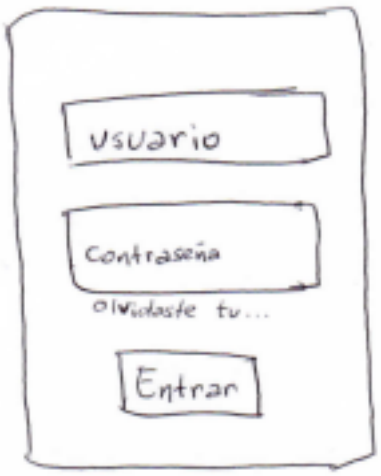


Figura 135: Diseño de intefaz. Ingreso



F. Prototipos digitales

Figura 136: Prototipo de tablero

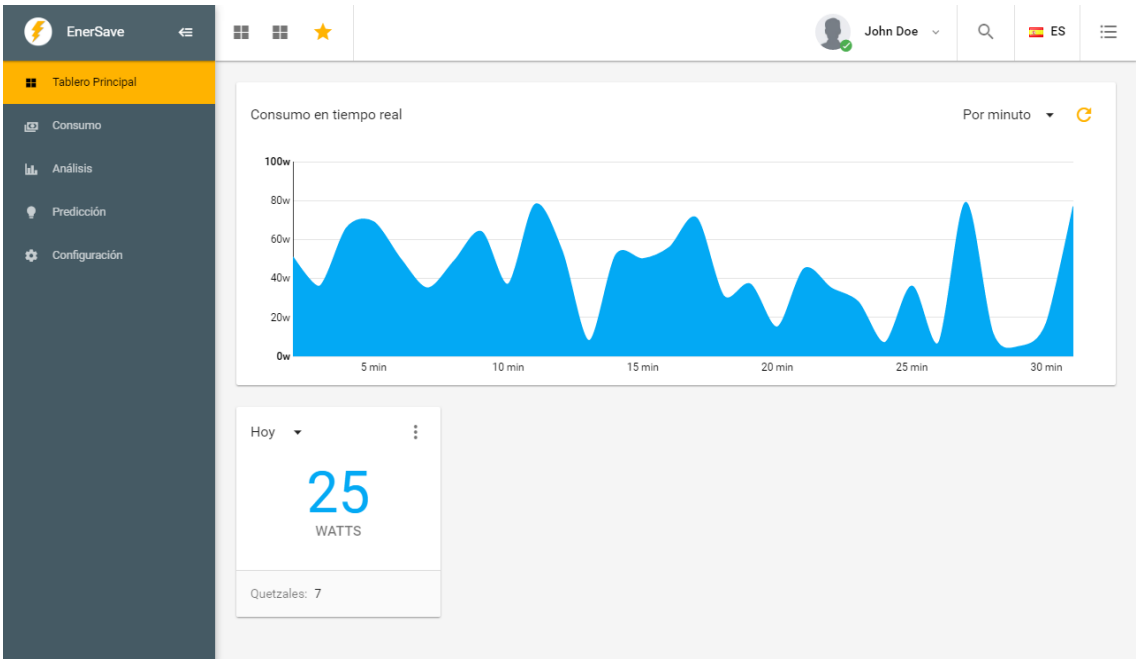


Figura 137: Prototipo de sección de consumo

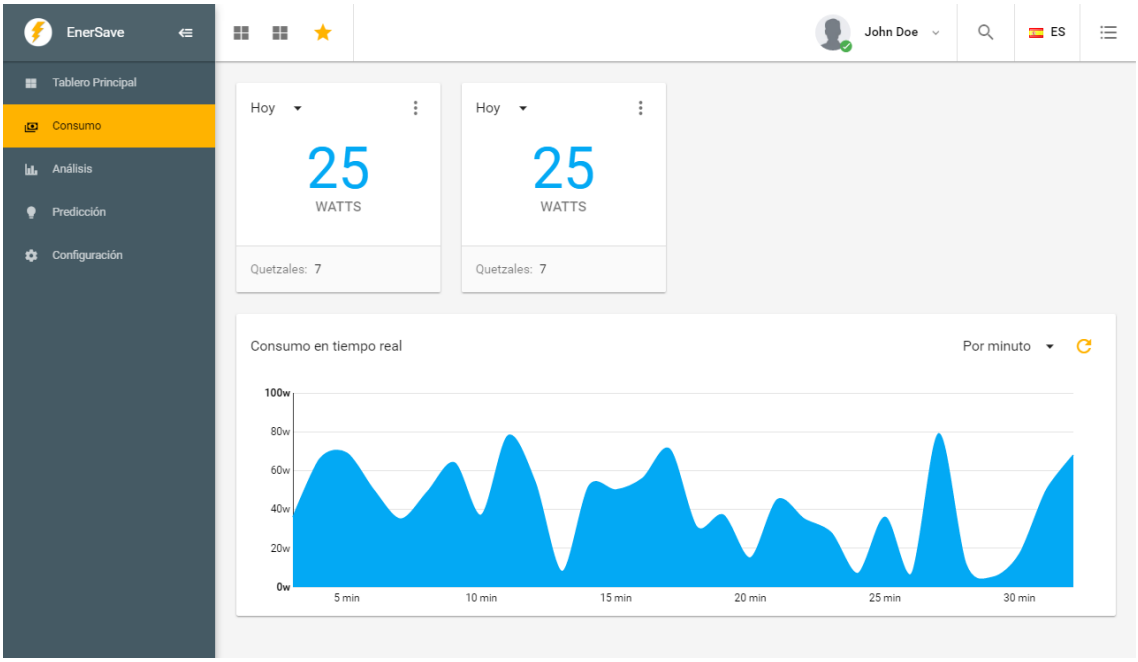


Figura 138: Prototipo de consumo. Barra colapsada

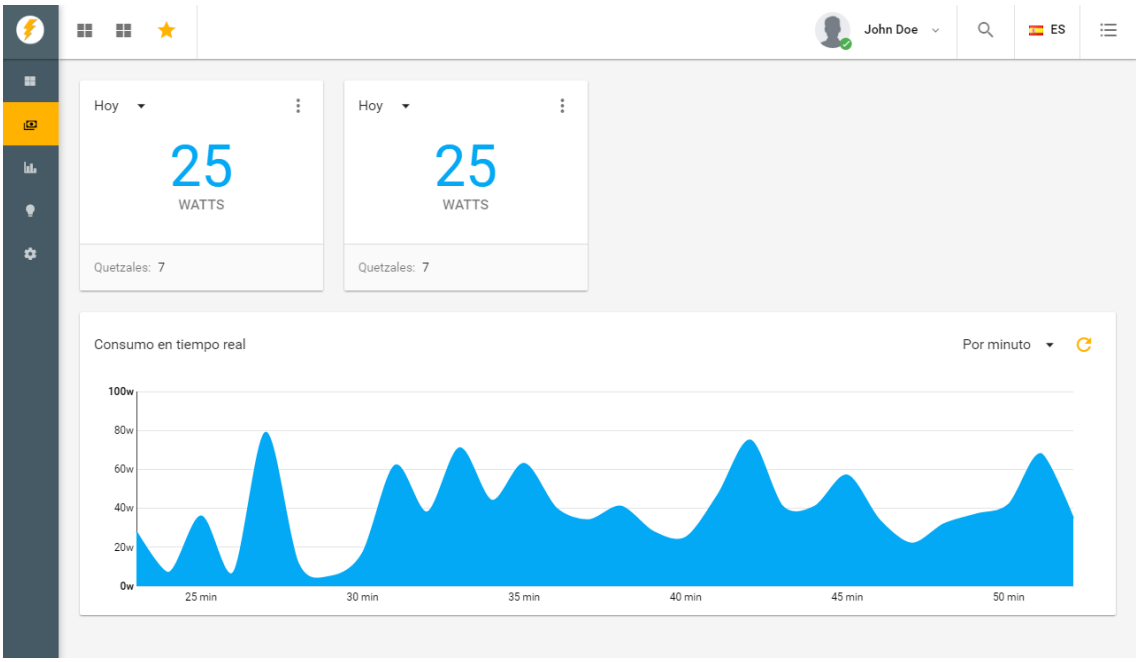


Figura 139: Prototipo de ingreso. *Splash screen.*

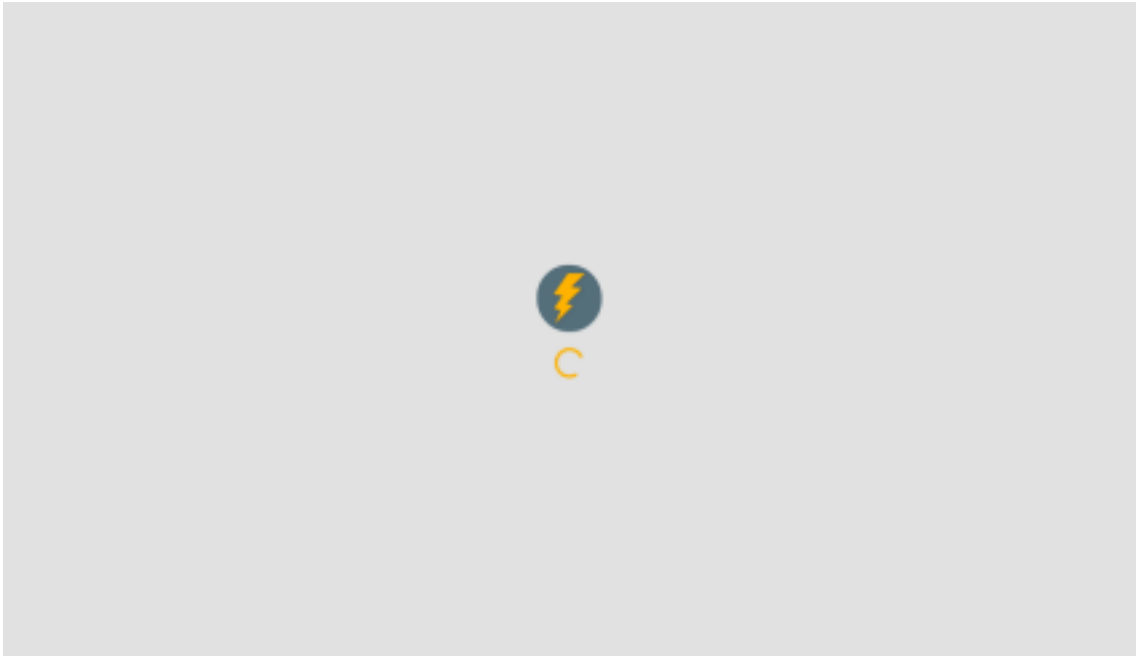
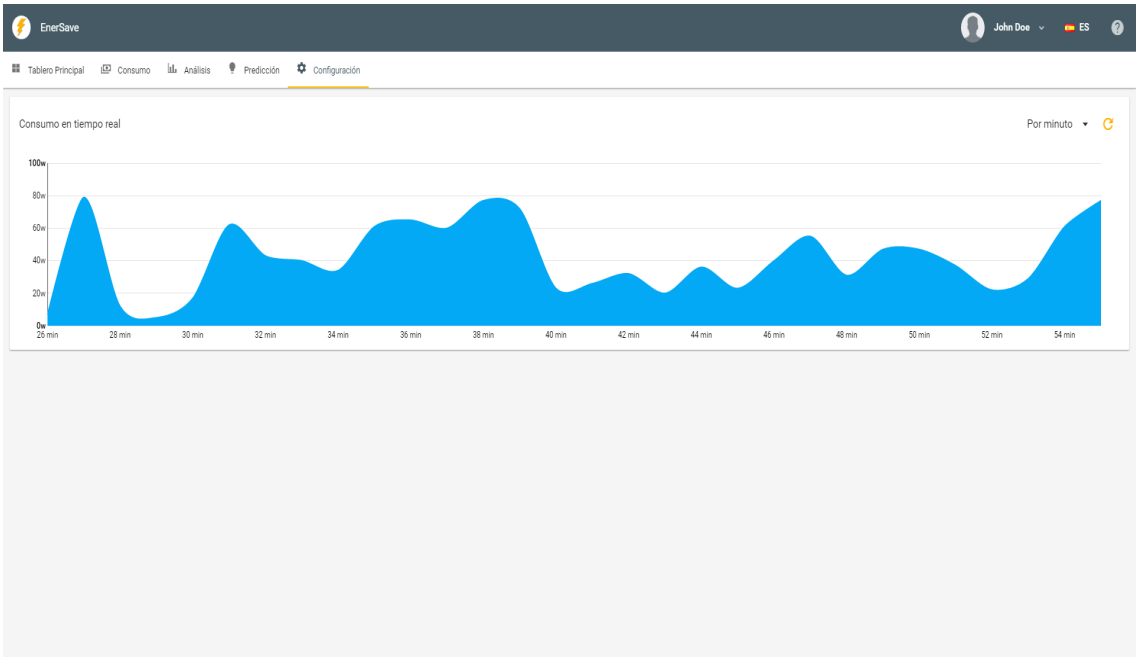


Figura 140: Prototipo de consumo. Barra horizontal.



G. Estudio de usabilidad I

Figura 141: Estudio de usabilidad I.

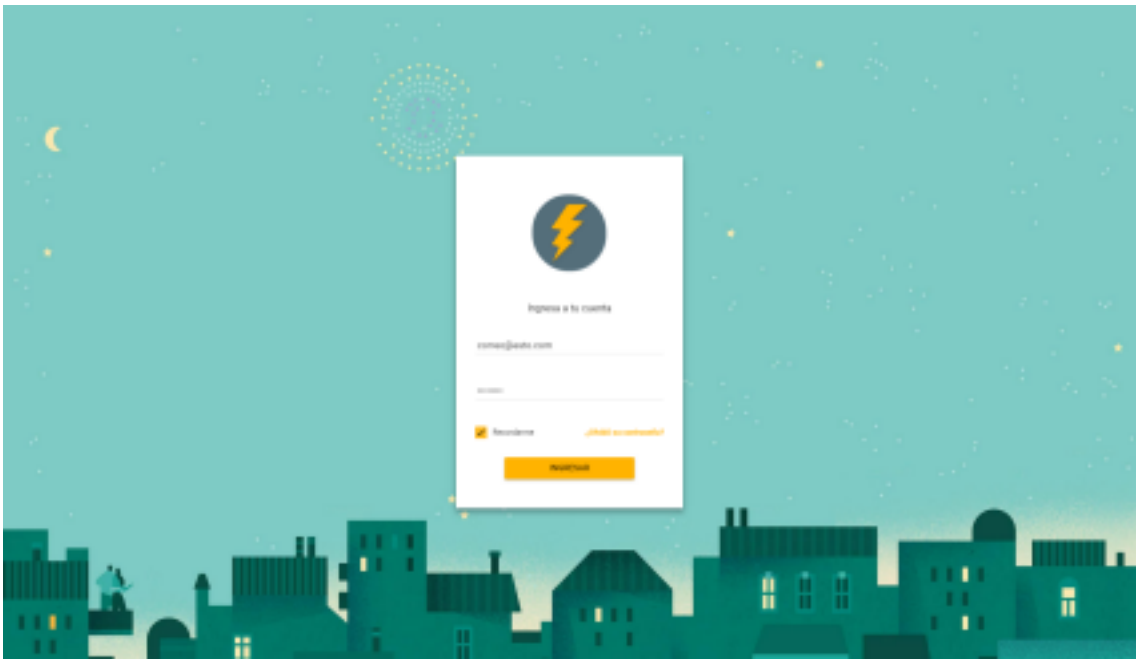


Figura 142: Estudio de usabilidad I.

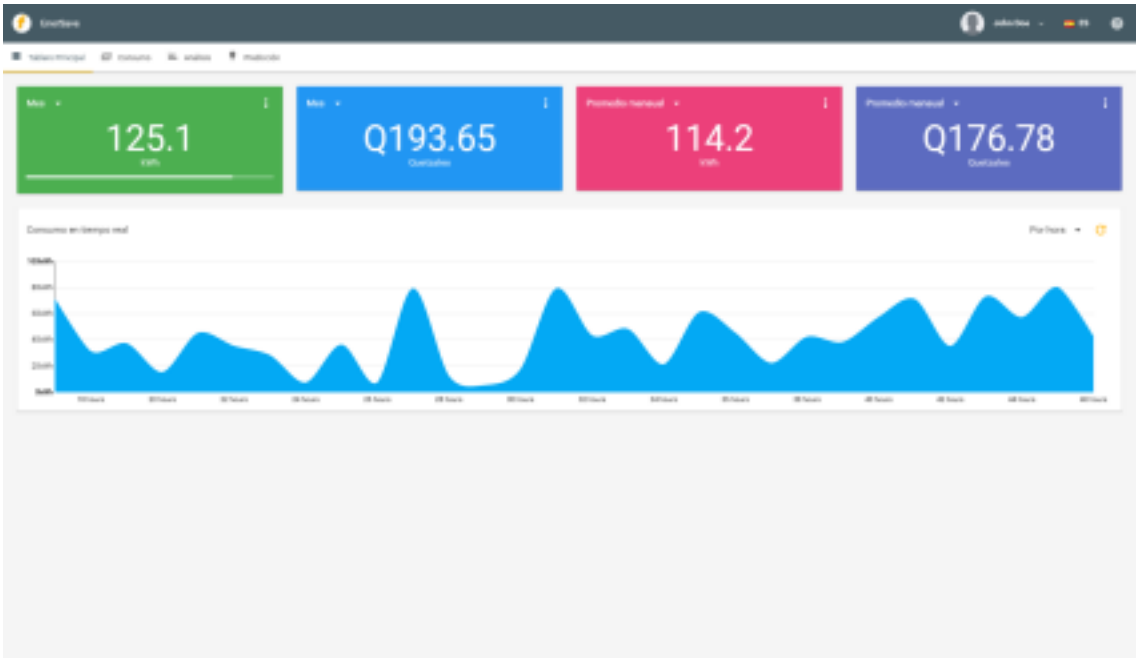


Figura 143: Estudio de usabilidad I.

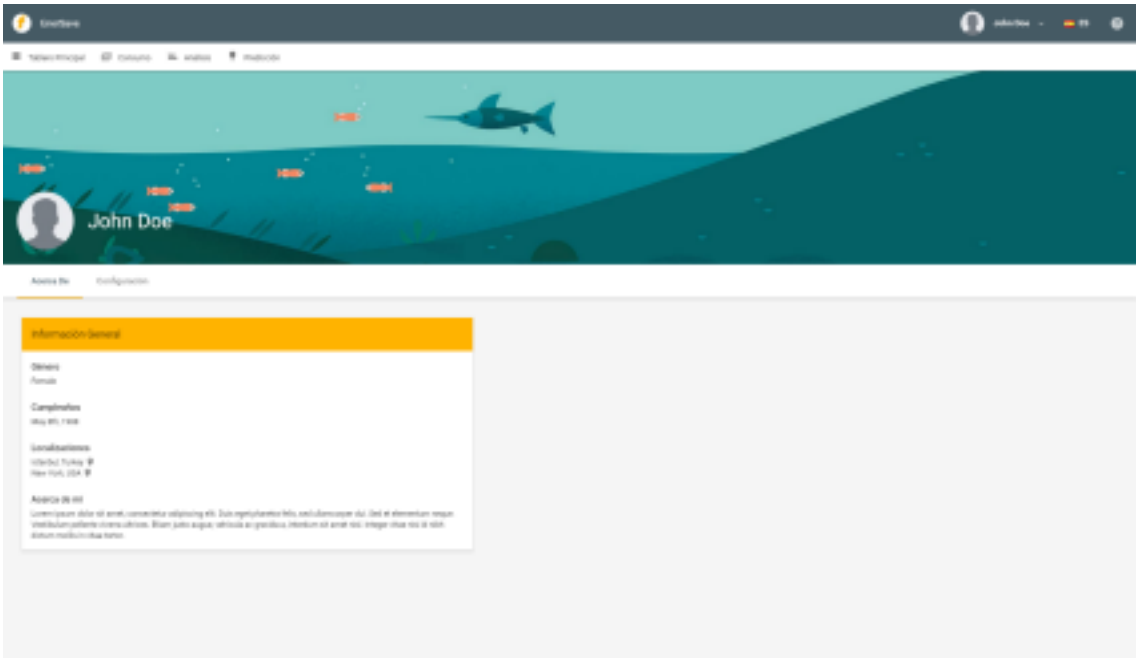


Figura 144: Estudio de usabilidad I.

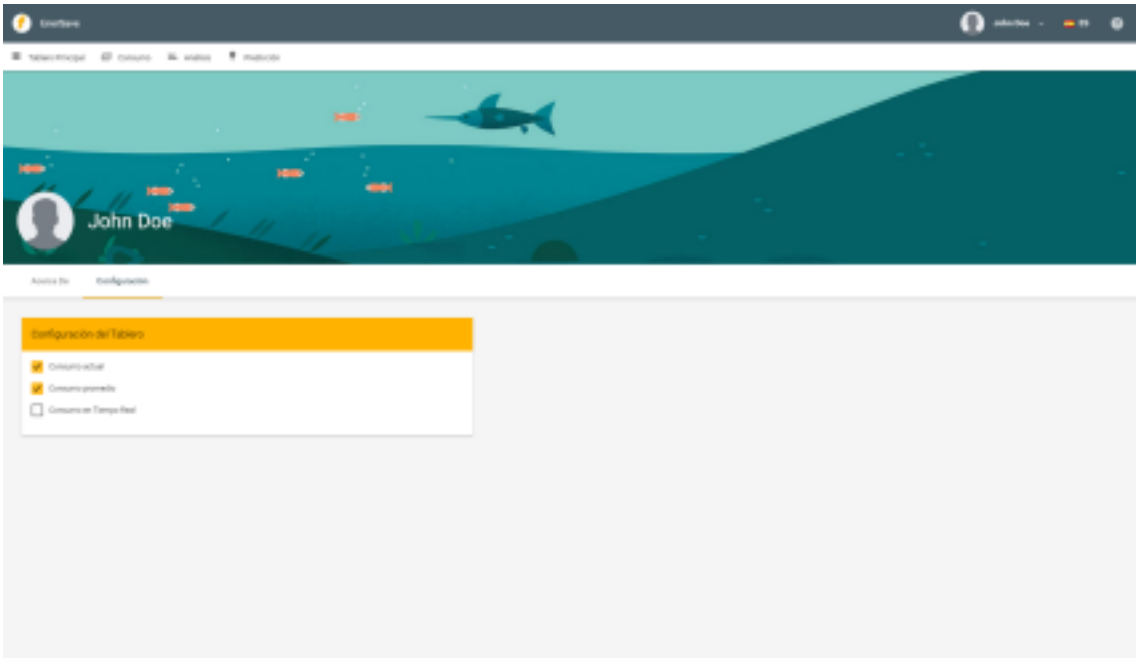
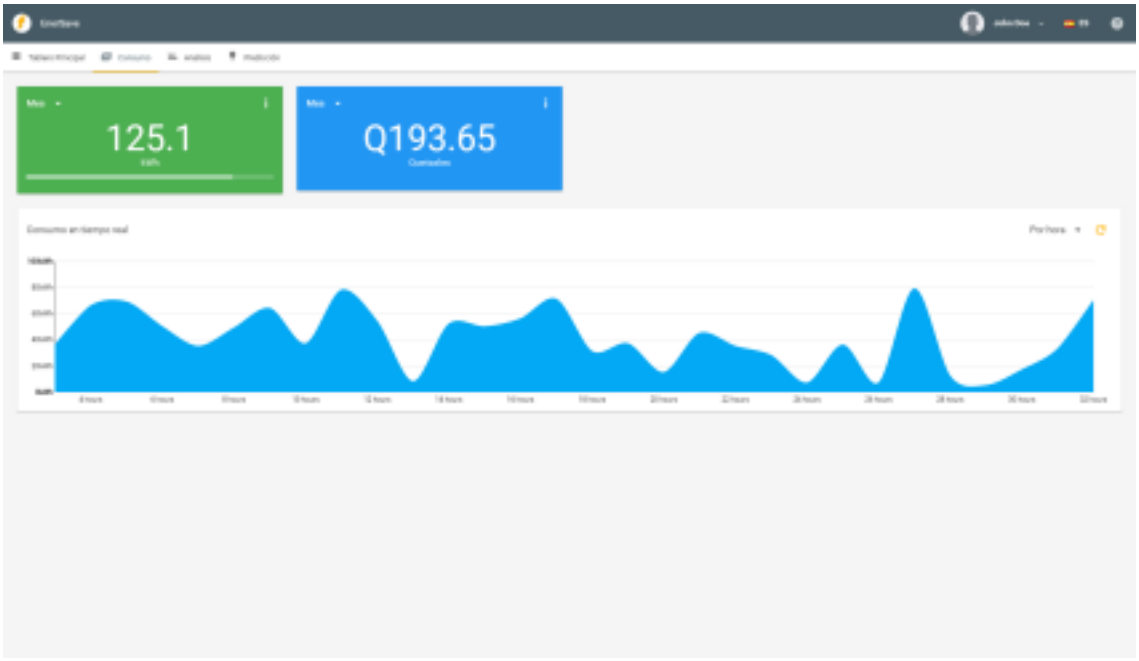
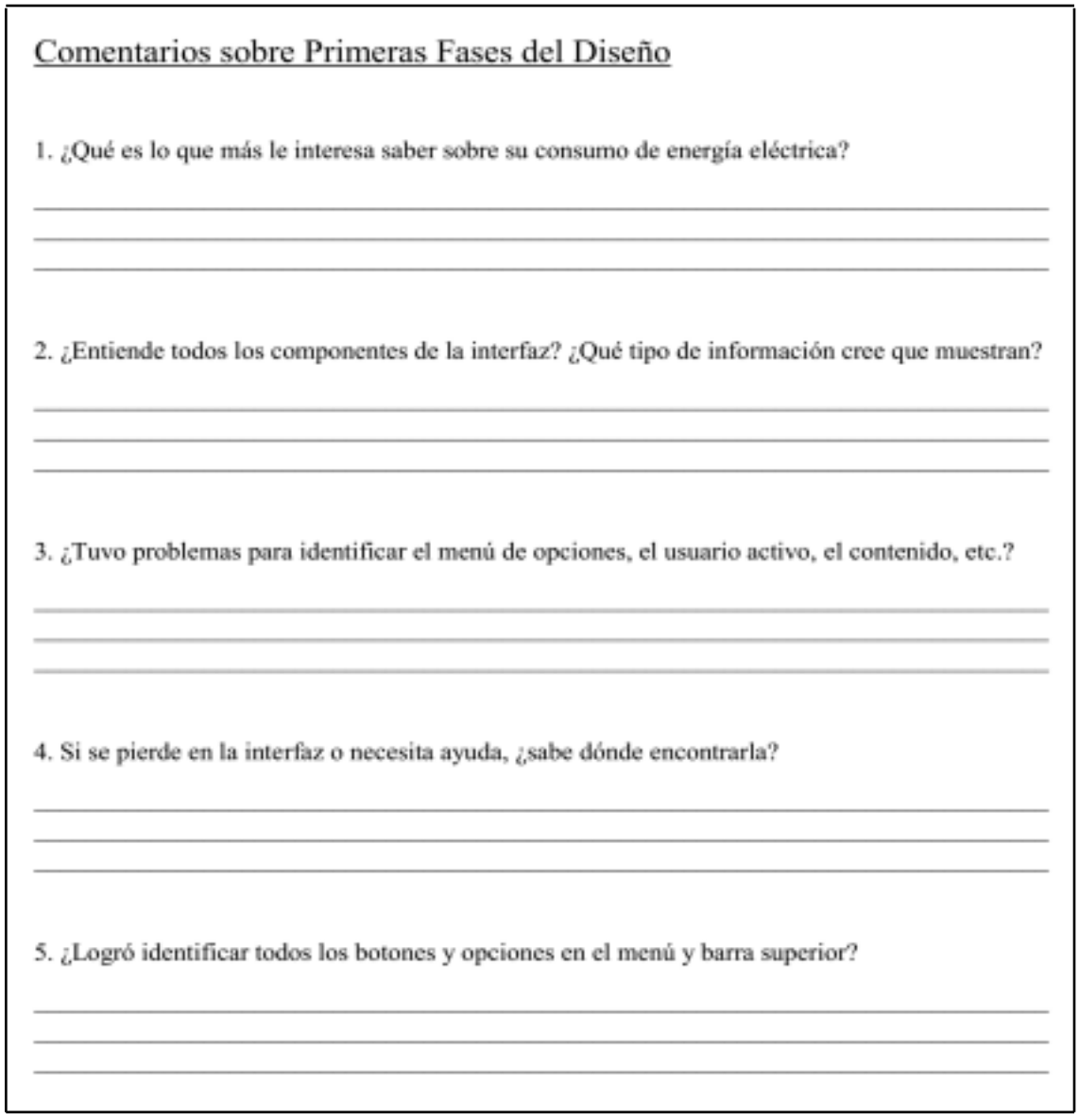


Figura 145: Estudio de usabilidad I.



H. Documentos para estudios de usabilidad

Figura 146: Documentos. Comentarios de usuarios



Comentarios sobre Primeras Fases del Diseño

1. ¿Qué es lo que más le interesa saber sobre su consumo de energía eléctrica?

2. ¿Entiende todos los componentes de la interfaz? ¿Qué tipo de información cree que muestran?

3. ¿Tuvo problemas para identificar el menú de opciones, el usuario activo, el contenido, etc.?

4. Si se pierde en la interfaz o necesita ayuda, ¿sabe dónde encontrarla?

5. ¿Logró identificar todos los botones y opciones en el menú y barra superior?

Figura 147: Documentos. Consentimiento informado

Consentimiento Informado

Se le agradece su participación en esta entrevista perteneciente al Megaproyecto "*EnerSave: Monitoreo del Consumo de Energía Eléctrica en Tiempo Real*" desarrollado por estudiantes de la Universidad del Valle de Guatemala pertenecientes al último año de las carreras de Ingeniería en Ciencias de la Computación y Tecnologías de la Información e Ingeniería Mecatrónica. Dicho proyecto es asesorado por el Ingeniero Sergio Izquierdo y la Coordinadora del Megaproyecto, Ing. Lynette García.

El objetivo general del proyecto es desarrollar una plataforma que permita el monitoreo del consumo de energía eléctrica en tiempo real en los hogares. Para utilizar esta plataforma, se necesita el diseño y desarrollo de una interfaz de usuario que sea útil y fácil de usar. Dado lo anterior, su participación en esta prueba ayudará a determinar la mejor forma de diseñar la interfaz gráfica para EnerSave. Se mantendrá completa confidencialidad en todos los resultados obtenidos durante este estudio, y su anonimato está asegurado. Los datos que se pretenden recaudar serán utilizados expresa y únicamente para fines de esta investigación, por lo tanto, no se extraerán conclusiones de forma individual. Finalmente hacemos de su conocimiento que usted está en su derecho de retirarse de la realización de la entrevista en cualquier momento. Por último, se le agradece nuevamente por su participación en este proyecto.

Yo: _____

Fecha de nacimiento: _____

El día de hoy:

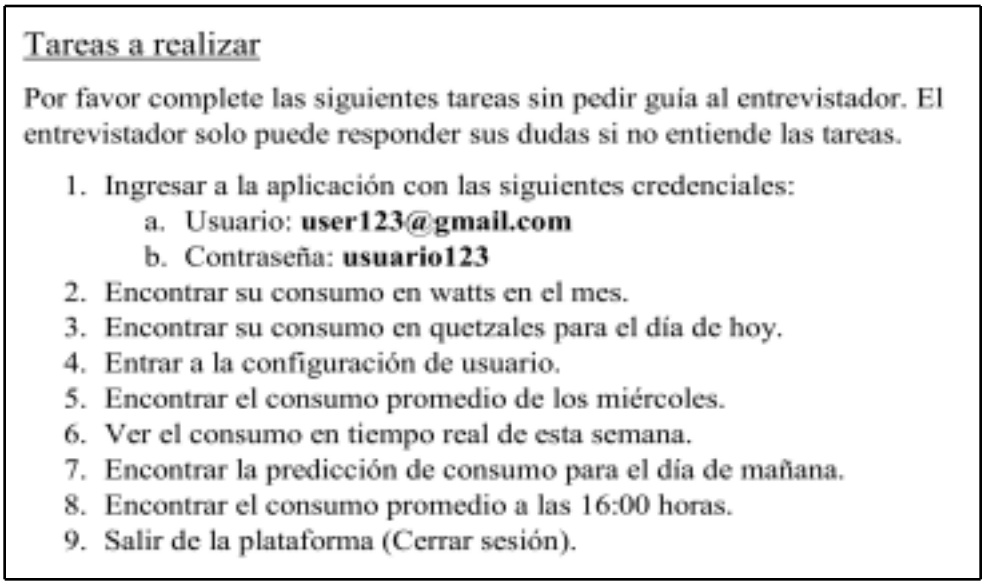
1. **Acepto participar en el proyecto "Enersave" para fines educativos en la Universidad del Valle de Guatemala.**
2. **Comprendo la naturaleza y propósito del procedimiento.**
3. **Comprendo que la información que proporcione será anónima.**
4. **He tomado la oportunidad de aclarar todas mis dudas.**
5. **Estoy consciente de los posibles riesgos y beneficios que este proyecto representa para mí.**
6. **Entiendo que puedo retirarme en cualquier momento.**
7. **Estoy satisfecho(a) con la información que se proporcionó en este consentimiento informado.**

Firma: _____

Fecha de hoy: _____

Cualquier duda relacionada a la entrevista o al resultado de la misma, se puede comunicar con Oscar Gil, correo: gil12358@uvg.edu.gt.

Figura 148: Documentos. Tareas a realizar durante entrevista

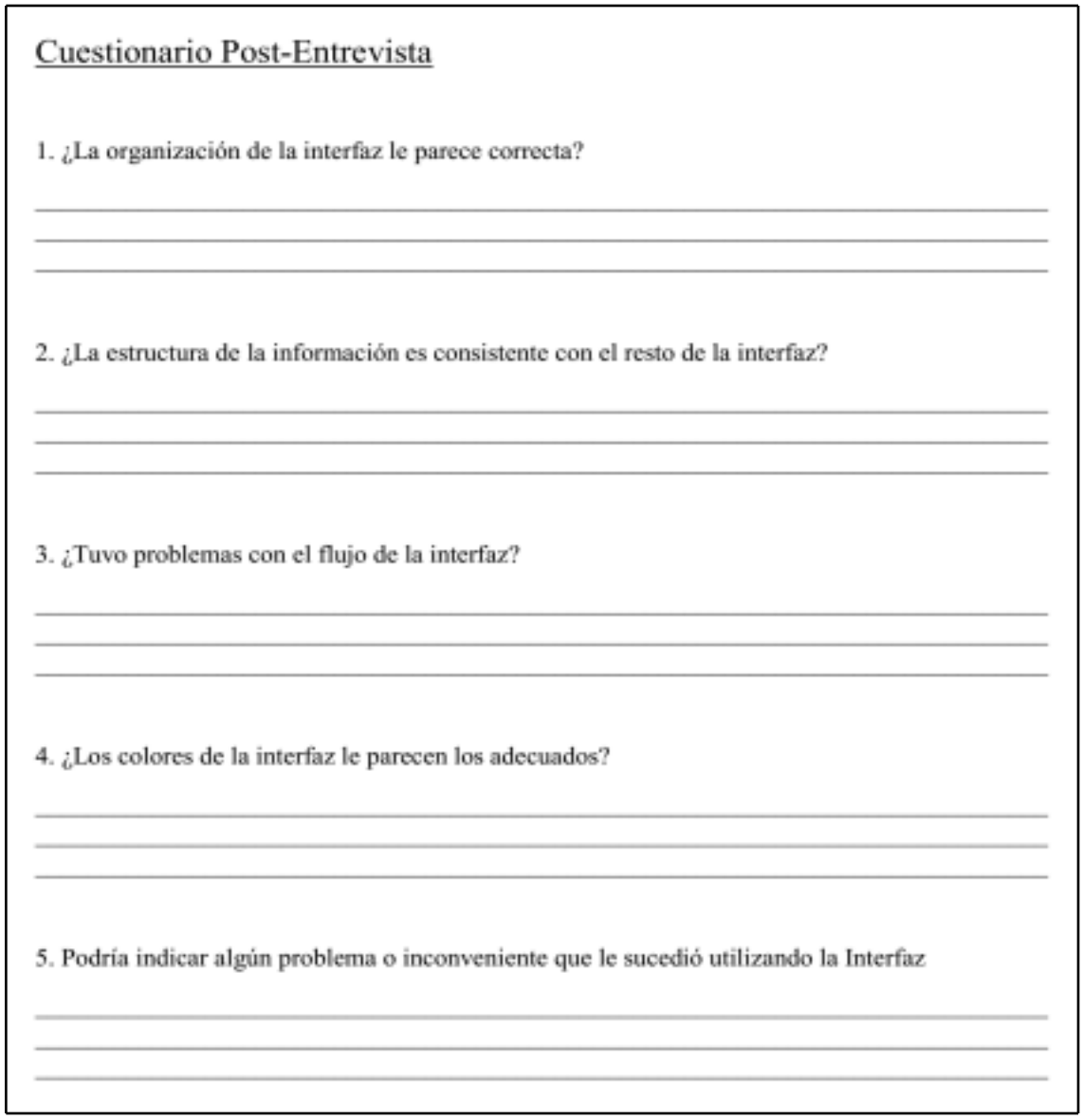


Tareas a realizar

Por favor complete las siguientes tareas sin pedir guía al entrevistador. El entrevistador solo puede responder sus dudas si no entiende las tareas.

1. Ingresar a la aplicación con las siguientes credenciales:
    a. Usuario: **user123@gmail.com**
    b. Contraseña: **usuario123**
2. Encontrar su consumo en watts en el mes.
3. Encontrar su consumo en quetzales para el día de hoy.
4. Entrar a la configuración de usuario.
5. Encontrar el consumo promedio de los miércoles.
6. Ver el consumo en tiempo real de esta semana.
7. Encontrar la predicción de consumo para el día de mañana.
8. Encontrar el consumo promedio a las 16:00 horas.
9. Salir de la plataforma (Cerrar sesión).

Figura 149: Documentos. Cuestionario



Cuestionario Post-Entrevista

1. ¿La organización de la interfaz le parece correcta?

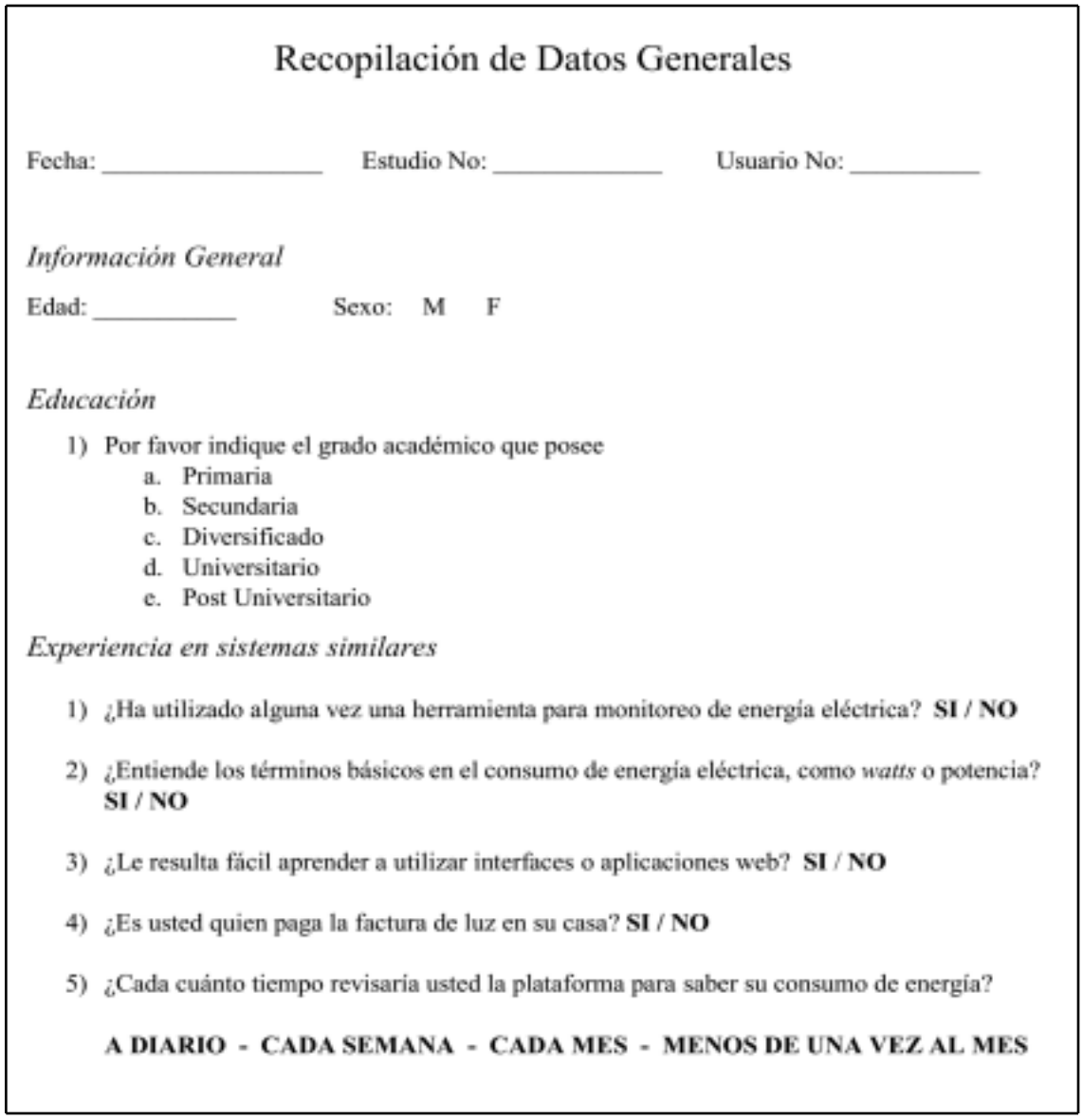
2. ¿La estructura de la información es consistente con el resto de la interfaz?

3. ¿Tuvo problemas con el flujo de la interfaz?

4. ¿Los colores de la interfaz le parecen los adecuados?

5. Podría indicar algún problema o inconveniente que le sucedió utilizando la Interfaz

Figura 150: Documentos. Datos generales



Recopilación de Datos Generales

Fecha: _____ Estudio No: _____ Usuario No: _____

Información General

Edad: _____ Sexo: M F

Educación

1) Por favor indique el grado académico que posee
   a. Primaria
   b. Secundaria
   c. Diversificado
   d. Universitario
   e. Post Universitario

Experiencia en sistemas similares

1) ¿Ha utilizado alguna vez una herramienta para monitoreo de energía eléctrica? **SI / NO**
2) ¿Entiende los términos básicos en el consumo de energía eléctrica, como *watts* o potencia? **SI / NO**
3) ¿Le resulta fácil aprender a utilizar interfaces o aplicaciones web? **SI / NO**
4) ¿Es usted quien paga la factura de luz en su casa? **SI / NO**
5) ¿Cada cuánto tiempo revisaría usted la plataforma para saber su consumo de energía?

   A DIARIO - CADA SEMANA - CADA MES - MENOS DE UNA VEZ AL MES