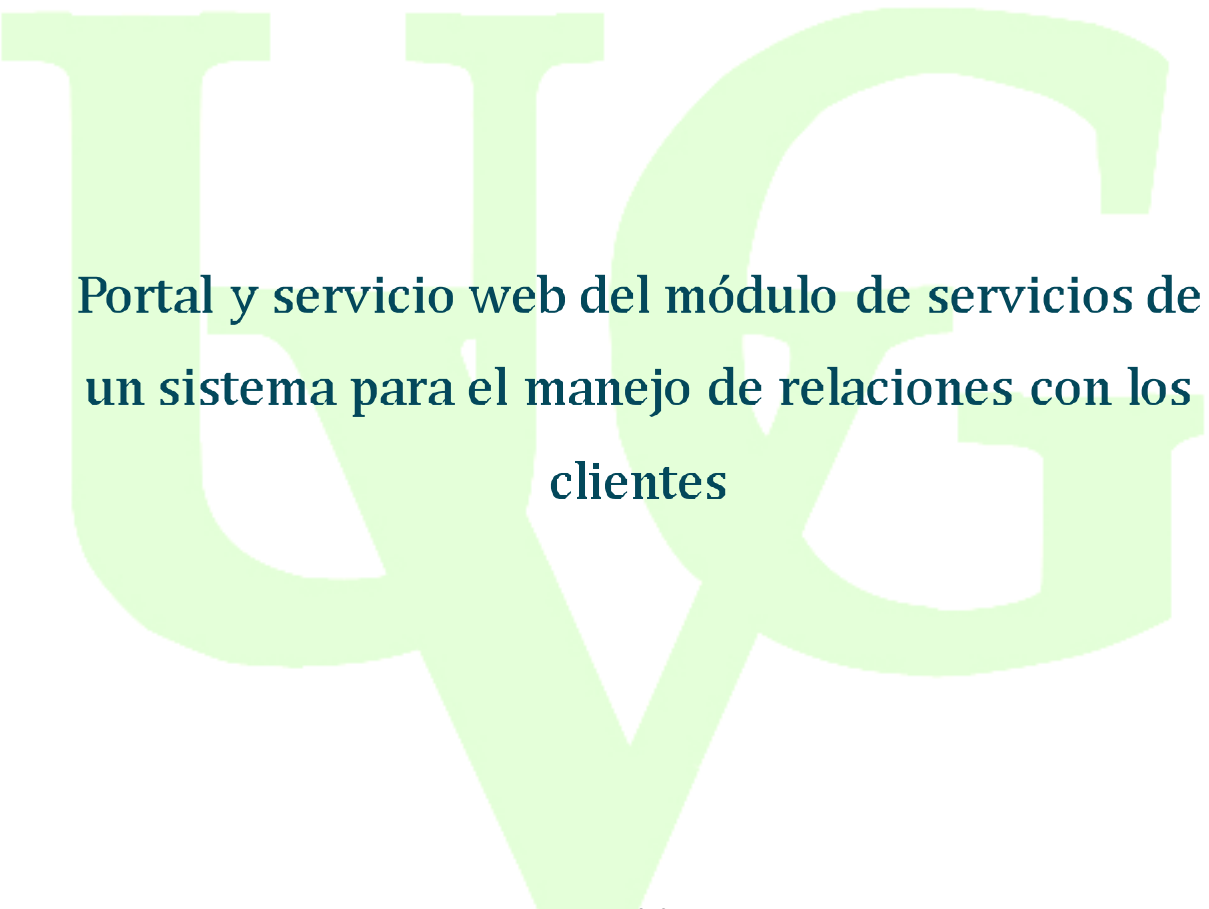



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Portal y servicio web del módulo de servicios de
un sistema para el manejo de relaciones con los
clientes**


Dennis Moritz Aldana Moscoso

Guatemala
2010



**Portal y servicio web del módulo de servicios de
un sistema para el manejo de relaciones con los
clientes**

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Portal y servicio web del módulo de servicios de
un sistema para el manejo de relaciones con los
clientes**

Trabajo de investigación presentado por
Dennis Moritz Aldana Moscoso
para optar al grado académico de
Licenciado en Ingeniería en Ciencias de la Computación

Guatemala
2010

Vo. Bo.:

f _____

(Ing. Heberth Campos)

Tribunal Examinador:

f _____

(Ing. Heberth Campos)

f _____

(Ing. Jenniffer Guzmán)

f _____

(Ing. Bidkar Pojoy)

Fecha de aprobación: Guatemala 9 de diciembre del 2010

Prefacio

Fue difícil encontrar una empresa especializada en el área de Ciencias de la Computación, que fuera flexible y permitiera implementar un proyecto con una idea nueva. Gracias al apoyo de María Mercedes Zaghi se pudo encontrar la empresa 1800 Software, que cumplía con esas características. Dentro del software que maneja esta empresa se encuentra Microsoft Dynamics, un sistema para la gestión de clientes. El precio de esta solución es alto y está fuera del alcance de muchas empresas en Guatemala.

Este proyecto se desarrolló con el objetivo de proveer una alternativa de bajo costo a los sistemas CRM existentes, que se ajuste a las necesidades específicas del cliente, y que sea fácilmente adoptable. Adicionalmente y para hacer más atractiva la solución, se desarrolló un proyecto paralelo que consiste en crear una interfaz móvil que se pueda conectar y sincronizar con este sistema, y que esté enfocada a la administración de las actividades de los técnicos que se encargan de prestar el servicio a los clientes fuera de las instalaciones de la empresa. Este proyecto paralelo se llevó a cabo por Helder Gramajo como trabajo de graduación titulado *Interfaz móvil del módulo de servicios de un sistema para el manejo de relaciones con los clientes*.

En general, las prácticas profesionales fueron una gran experiencia de aprendizaje que complementa en gran medida a la educación universitaria, y es al mismo tiempo una oportunidad de aportar algo para mejorar el país.

Tabla de contenido

Prefacio	V
Tabla de contenido	VI
Lista de cuadros	X
Lista de gráficos e ilustraciones	XI
Resumen.....	XII
I. Introducción	- 1 -
A. Descripción general	- 2 -
B. Justificación.....	- 3 -
C. Delimitación	- 5 -
1. Resultado	- 5 -
2. Pantallas de acceso al sistema (entregables)	- 5 -
3. Definición de data	- 6 -
4. Delimitación funcional	- 7 -
5. Límites.....	- 7 -
D. Metodología	- 7 -
E. Estructura de la tesis.....	- 8 -
II. Teoría	- 9 -
A. Ingeniería de software	- 9 -
1. Modelo clásico	- 10 -
2. Modelo incremental.....	- 14 -
3. Modelo DRA	- 15 -
B. Ingeniería de requisitos	- 17 -

1. Definición	- 17 -
2. KAOS.....	- 18 -
C. Software CRM	- 18 -
1. Características.....	- 19 -
2. Software CRM existente	- 22 -
D. Sistemas ERP.....	- 27 -
E. Computación en nube	- 28 -
1. SaaS.....	- 28 -
2. Computación de utilidades	- 28 -
3. Servicios web en nube	- 28 -
4. PaaS.....	- 29 -
5. MSP	- 29 -
6. Plataformas de comercio de servicios	- 29 -
F. Seguridad de la información	- 29 -
1. HTTPS	- 29 -
2. AES.....	- 30 -
3. Triple DES	- 30 -
G. Herramientas utilizadas.....	- 31 -
1. Windows Seven.....	- 31 -
2. VirtualBox.....	- 31 -
3. Windows Server 2008 R2	- 32 -
4. SQL Server 2008	- 32 -
5. IIS.....	- 33 -
6. Microsoft Office	- 33 -

7. Visual Studio.....	- 33 -
8. Active Directory.....	- 34 -
9. Internet Explorer.....	- 35 -
10. SharePoint.....	- 35 -
III. Planteamiento del proyecto	- 37 -
A. Metas	- 37 -
B. Objetivos.....	- 38 -
1. Objetivo general.....	- 38 -
2. Objetivos específicos.....	- 38 -
IV. Desarrollo.....	- 39 -
A. Método	- 39 -
1. Sujetos.....	- 39 -
2. Herramientas	- 40 -
3. Procedimiento.....	- 40 -
4. Discusión.....	- 41 -
B. Análisis	- 42 -
1. Modelo de requerimientos	- 42 -
2. Modelo de datos	- 45 -
3. Modelo de escenarios.....	- 47 -
4. Usuarios del sistema	- 49 -
5. Discusión.....	- 50 -
C. Diseño	- 51 -
1. Diseño arquitectónico.....	- 51 -
2. Diseño de componentes	- 54 -

3. Diseño de la interfaz de usuario	- 57 -
4. Diseño de seguridad de la información	- 67 -
5. Diseño de pruebas	- 68 -
6. Diseño de implementación	- 70 -
7. Discusión	- 71 -
V. Discusión.....	- 72 -
VI. Conclusiones	- 73 -
VII. Recomendaciones	- 74 -
VIII. Bibliografía.....	- 75 -
IX. Anexos	- 77 -
A. Revisión de términos	- 77 -
B. Revisión de literatura	- 81 -
C. Algoritmo para la codificación de la información de usuario	- 82 -
1. Componentes	- 82 -
2. Formato de la tabla de codificación.....	- 82 -
3. Algoritmo	- 82 -
D. Interfaz gráfica en SharePoint 2007	- 83 -

Lista de cuadros

TABLA 1: DEFINICIÓN DE DATOS	- 6 -
TABLA 2: CONCEPTOS EN LA METODOLOGÍA KAOS	- 18 -
TABLA 3: MÓDULOS DE UN CRM	- 19 -
TABLA 4: COMPARACIÓN ENTRE SOFTWARE CRM	- 22 -
TABLA 5: ENTIDADES PRINCIPALES DEL SISTEMA	- 45 -
TABLA 6: USUARIOS DEL SISTEMA	- 49 -
TABLA 7: ARQUITECTURA ESTRATIFICADA	- 51 -

Lista de gráficos e ilustraciones

ILUSTRACIÓN 1: COMPARACIÓN DE PRECIOS DE CRM'S	- 3 -
ILUSTRACIÓN 2: MODELO EN CASCADA	- 14 -
ILUSTRACIÓN 3: MODELO INCREMENTAL	- 15 -
ILUSTRACIÓN 4: DESARROLLO RÁPIDO DE APLICACIONES	- 16 -
ILUSTRACIÓN 5: SAP CRM	- 23 -
ILUSTRACIÓN 6: ORACLE SIEBEL	- 24 -
ILUSTRACIÓN 7: SALESFORCE 8 STA	- 25 -
ILUSTRACIÓN 8: MICROSOFT DYNAMICS	- 26 -
ILUSTRACIÓN 9: MÁQUINA DE DESARROLLO	- 32 -
ILUSTRACIÓN 10: VISUAL STUDIO 2010.....	- 34 -
ILUSTRACIÓN 11: SHAREPOINT 2007	- 36 -
ILUSTRACIÓN 12: SHAREPOINT 2010	- 36 -
ILUSTRACIÓN 13: RELACIONES ENTRE LAS ENTIDADES DEL SISTEMA	- 47 -
ILUSTRACIÓN 14: CASOS DE USO, TODOS LOS USUARIOS	- 48 -
ILUSTRACIÓN 15: CASOS DE USO, COORDINADORES Y ADMINISTRADORES	- 48 -
ILUSTRACIÓN 16: CASOS DE USO, ADMINISTRADORES	- 48 -
ILUSTRACIÓN 17: DIAGRAMA DE ESTADOS, ACTIVIDADES	- 49 -
ILUSTRACIÓN 18: DIAGRAMA DE CONTEXTO	- 52 -
ILUSTRACIÓN 19: COMPONENTES DEL SISTEMA	- 54 -
ILUSTRACIÓN 20: MAQUETA DE LA APARIENCIA	- 57 -
ILUSTRACIÓN 21: PANTALLA DE INICIO	- 58 -
ILUSTRACIÓN 22: PANTALLA DE CONFIGURACIÓN.....	- 59 -
ILUSTRACIÓN 23: PANTALLA PARA LISTAR USUARIOS.....	- 60 -
ILUSTRACIÓN 24: PÁGINA PERSONAL	- 61 -
ILUSTRACIÓN 25: NUEVO USUARIO	- 62 -
ILUSTRACIÓN 26: PANTALLA PARA BUSCAR Y LISTAR ELEMENTOS	- 63 -
ILUSTRACIÓN 27: FUNCIONES DE UNA LISTA	- 63 -
ILUSTRACIÓN 28: CALENDARIO DE ACTIVIDADES MENSUAL.....	- 64 -
ILUSTRACIÓN 29: CALENDARIO DE ACTIVIDADES SEMANAL	- 64 -
ILUSTRACIÓN 30: CALENDARIO DE ACTIVIDADES DIARIO	- 64 -
ILUSTRACIÓN 31: DETALLE DE ELEMENTO	- 65 -
ILUSTRACIÓN 32: DETALLE SIMPLE	- 65 -
ILUSTRACIÓN 33: NUEVO ELEMENTO	- 66 -
ILUSTRACIÓN 34: EDITAR ELEMENTO	- 66 -

Resumen

Este trabajo presenta el procedimiento llevado a cabo para el desarrollo de un sistema servidor para el manejo de relaciones entre una empresa de servicios y sus clientes.

Esta herramienta permite la coordinación de los proveedores de servicio de la empresa además del manejo de la información de los clientes. Al utilizarla para estos propósitos, se espera que la empresa pueda mejorar la calidad de servicio que es capaz de prestar a sus clientes. De esta manera, la empresa será más competitiva en el mercado nacional, pues atraerá nuevos clientes y aumentará la lealtad de sus clientes existentes.

I. Introducción

El sector de servicios está en un constante desarrollo en la actualidad, y cada vez son más las empresas que compiten por satisfacer las necesidades de los clientes. Por lo tanto, es muy importante que sean atractivas para clientes nuevos, y que mantengan un alto nivel de lealtad con sus clientes existentes.

Una manera de lograr esta lealtad por parte de los clientes, es manteniendo una buena relación con ellos, mientras se provee un buen servicio. Regularmente las empresas buscan tener tantos clientes como sea posible, por lo que organizar esta información y asegurar un acceso rápido es un problema creciente. Es necesario de alguna manera centralizar la información y ponerla a fácil disposición de las personas que realizan el servicio. Además, estos proveedores de servicio deben ser monitoreados, coordinados y organizados para que la empresa pueda asegurar que las necesidades del cliente están siendo satisfechas.

Actualmente existen sistemas que una empresa puede implementar para administrar sus relaciones con los clientes, estos se conocen comúnmente como CRM's, por las siglas en inglés para *Customer Relationship Management*. Estos sistemas son capaces de administrar las interacciones entre una empresa, sus clientes actuales y sus clientes potenciales.

Un CRM hace uso de la tecnología para automatizar y sincronizar algunos procesos de negocios, en especial las actividades de ventas, mercadeo y servicio al cliente; buscando de esta manera reducir costos y conseguir la lealtad de los clientes. Su objetivo principal es atraer nuevos clientes, y mantener los clientes que la empresa tiene, por lo tanto, son una solución ideal para una empresa de servicio.

A. Descripción general

Como trabajo de graduación, se realizó un proyecto en la empresa 1800 Software que inició en el mes de febrero del año 2010 y se extendió por 400 horas. El proyecto se llevó a cabo como prácticas profesionales y fue supervisado por un asesor interno.

1800 Software es una empresa de tecnología que sigue el modelo llamado **computación en nube** (*Cloud Computing*). Dentro de sus actividades está la personalización de las aplicaciones de Microsoft Project, SharePoint y Dynamics. La aplicación propuesta fue desarrollada de manera acorde con este modelo.

La computación en nube consiste en un modelo en donde la computación se basa en el Internet. Los recursos de software e información se proveen a los dispositivos que los solicitan bajo demanda, pero nunca están directamente instalados en las máquinas cliente.

La propuesta consiste en un sistema para la gestión de relaciones con los clientes (CRM). Sin embargo, a causa de la restricción de recursos, es imposible crear un sistema CRM completo, por lo que el proyecto consiste en la construcción únicamente del módulo de servicios. La solución se implementará sobre la plataforma SharePoint en línea, para que el cliente pueda acceder a ella en cualquier momento y lugar, y para facilitar su alquiler.

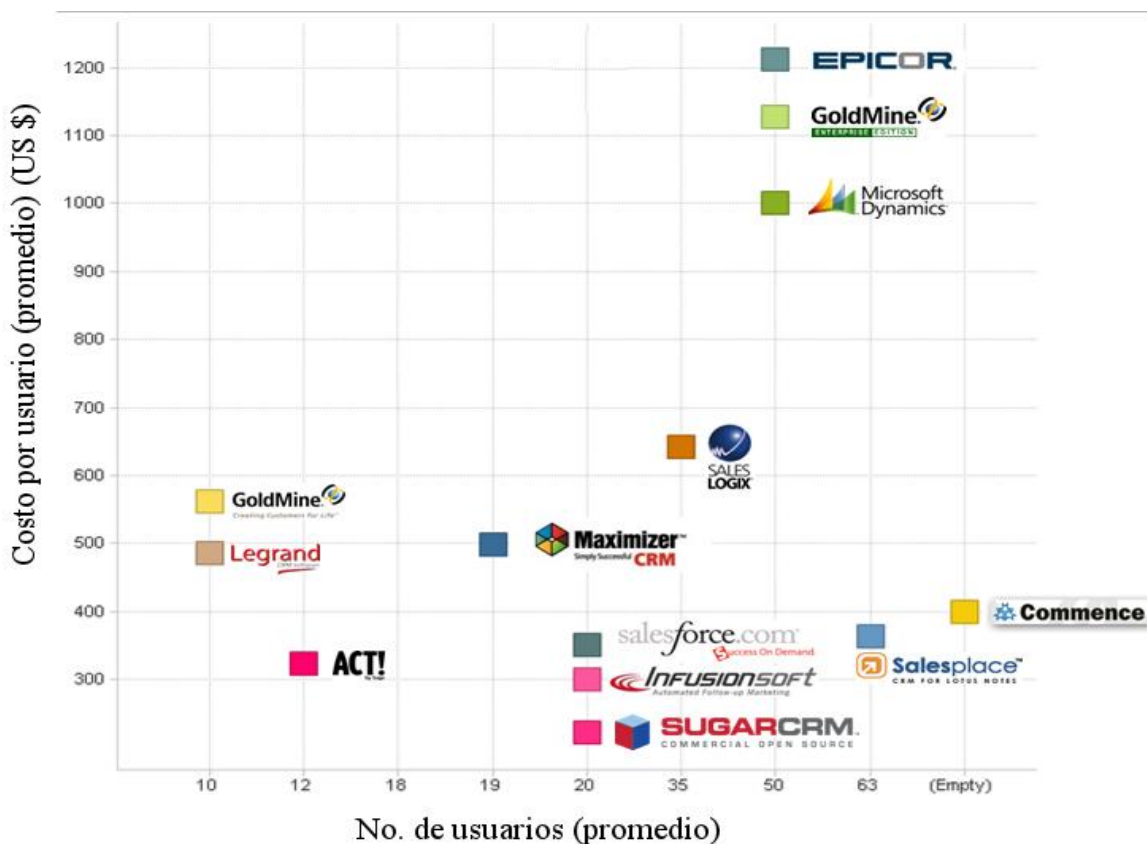
Paralelamente, se hizo el proyecto titulado: *Interfaz Móvil del Módulo de Servicios de un Sistema para el Manejo de Relaciones con los Clientes* para complementar las funcionalidades del módulo servidor. Este proyecto fue desarrollado por Helder Gramajo como trabajo de graduación.

B. Justificación

Manejar las relaciones con los clientes puede ser algo sencillo para empresas pequeñas. Normalmente hay un trabajador encargado de archivar la información de contacto con los clientes y de llevar control de los servicios que se le prestan a cada uno. Sin embargo, a medida que las empresas crecen, el número de clientes puede aumentar a cientos o miles, haciendo la administración manual prácticamente imposible.

Una empresa con un gran número de clientes necesita de un sistema CRM para administrarlos. Por su gran demanda existen varios sistemas CRM en la actualidad, pero su precio es considerablemente alto, como se puede observar en la **Error! Reference source not found.**

Ilustración 1: Comparación de precios de CRM's



(aiveo, 2009)

Dentro de los sistemas de CRM más populares hoy en día, resaltan: el de la empresa alemana SAP; el CRM que llegó a dominar el 45% mercado en 2002 y que fue adquirido por Oracle en 2005, Siebel; y Microsoft Dynamics, que es relativamente nuevo y gana mercado rápidamente. El precio de todas estas soluciones está lejos del alcance para empresas medianas en crecimiento por lo que se propone la creación de una solución especializada en el área de servicios que pueda ser vendida o alquilada a un precio menor.

Con la implementación de esta solución, las empresas medianas de servicios podrán adquirir a través de 1800 Software un CRM alternativo a un precio mucho menor que el de las grandes marcas, sin tener que pagar por funcionalidades que quizá no se utilicen. Además de un precio atractivo, la solución será capaz de adaptarse a las necesidades específicas de cada empresa interesada, lo cual facilitará la adaptación de los empleados y la integración a los sistemas existentes. Al mismo tiempo, los clientes de una empresa de servicio utilizando la solución propuesta se verán beneficiados, pues recibirán un mejor trato, de una manera más eficiente y veloz.

En el mercado de Guatemala, a pesar de la gran necesidad que hay de ellos, muy pocas empresas utilizan sistemas CRM. Introducir un nuevo software producido localmente impulsará la competencia de las empresas por brindar un mejor servicio y colaborará con la aceptación de sistemas CRM en el mercado.

Un estudiante que desarrolle el proyecto propuesto podrá adquirir conocimientos sobre la manera en que una empresa se relaciona con sus clientes y con sus técnicos. Estos conocimientos serán muy útiles al tratar tanto con clientes individuales como con empresas.

C. Delimitación

La delimitación del alcance de un proyecto consiste en desarrollar en detalle lo que está incluido o excluido del proyecto. Delimitar el proyecto de la manera correcta es importante para evitar malentendidos entre las partes interesadas. Existen varias maneras para delimitar el alcance, pero es conveniente primero llegar a un acuerdo acerca del resultado del proyecto. Una vez definido el resultado, se debe definir el alcance en sí, lo que se va a realizar y lo que no, las tres principales maneras para lograr esto, son: (Turbit, 2005)

- Definir los entregables
- Definir la funcionalidad
- Definir los datos

1. Resultado. Un sistema para la administración de las relaciones entre una empresa de servicios y sus clientes, enfocado en la administración de las actividades de servicio y la coordinación de los proveedores de servicio.

2. Pantallas de acceso al sistema (entregables). Desde el punto de vista de usuario, el sistema proveerá:

- Pantalla de inicio, con un acceso rápido a las funciones principales
- Pantalla de configuración del sitio
- Pantallas para listar usuarios, para mostrar el detalle de un usuario en particular y para crear uno nuevo
- Pantallas para buscar, crear, detallar, editar y listar:
 - Cuentas
 - Contactos
 - Contratistas
 - Contratos
 - Incidentes
- Pantallas para poder ver las actividades recientes
 - Detalle de una actividad
 - Lista de actividades en la semana
 - Lista de actividades en el día
 - Lista de actividades en el mes
- Pantallas para ver, crear, editar y eliminar circulares y anuncios

3. Definición de data.

El sistema guardará la información de:

Tabla 1: Definición de datos

Nombre	Descripción
Actividad	Una actividad es una tarea de servicio que un usuario del tipo “técnico” debe de llevar a cabo. La actividad tiene una fecha de inicio y finalización, y se debe llevar registro de su progreso en este tiempo.
Contacto	Un contacto es una entidad con un nombre y un apellido. Los clientes individuales que solicitan un servicio entran en esta categoría.
Cuenta	Una cuenta es una entidad financiera a la que se le presta un servicio. Los clientes que son empresas entran en esta categoría.
Contratista	Un contratista es una empresa que es subcontratada para realizar actividades en lugar de la empresa de servicios.
Contrato	Un contrato es un documento que sirve de prueba y delimitación de la relación entre la empresa y alguno de sus clientes.
Incidente	Un incidente es una solicitud de servicio por parte de un cliente, normalmente está asociado a actividades de servicio y debe ser resuelto.
Herramienta	Una herramienta es un objeto que es utilizado por uno o varios técnicos para ayudar a completar una actividad de servicio.
Material	Un material es un consumible que es utilizado en una sola actividad de servicio.
Mensaje	Un mensaje es un texto o un documento que es enviado desde un usuario a otros usuarios.

4. Delimitación funcional. El proyecto se limita a cumplir con las siguientes funciones:

- Administración de cada uno de los elementos en la Tabla 1:
 - Buscar
 - Mostrar
 - Listar
 - Crear
 - Modificar
 - Eliminar
- Administración de Técnicos/Coordinadores/Administradores:
 - Listar a los usuarios
 - Crear nuevos usuarios
- Interfaz de servicio web a la que se puedan acceder las demás funciones.

5. Límites. Las siguientes funcionalidades y módulos quedan fuera del alcance del proyecto:

- Interfaz móvil
- Módulo de ventas de un software CRM
- Módulo de mercadeo de un software CRM
- Funciones financieras del CRM
- Integración con otros sistemas de CRM
- Automatización de flujos de trabajo entre actividades
- Cálculo del costo total de la aplicación

D. Metodología

Tomando en consideración las limitaciones de tiempo y los requerimientos, se seleccionó la metodología conocida como “Desarrollo Rápido de Aplicaciones” para la implementación del módulo servidor. Éste es un modelo incremental en la que el ciclo de desarrollo es muy corto y con un enfoque orientado a componentes. Siguiendo este esquema, se decidió dividir el módulo servidor en varios módulos más pequeños y desarrollar cada uno de manera individual, sobre una plataforma previamente construida (SharePoint) que se encargue de la comunicación entre componentes.

E. Estructura de la tesis

El presente documento está organizado de la siguiente manera:

Introducción. En el capítulo introductorio, se expone la descripción más general del proyecto ejecutado, las razones que justifican su ejecución, su alcance y la metodología seguida. Pretende ser un punto de entrada para el documento y guiar a los lectores hacia los temas principales que serán tratados.

Teoría. El capítulo de teoría desarrolla los conceptos teóricos relacionados con el proyecto ejecutado, y describe las herramientas de software y técnicas utilizadas.

Planteamiento del proyecto. En este capítulo se describen los objetivos y metas que se buscaba alcanzar a través del proyecto.

Desarrollo. El capítulo de desarrollo cuenta los detalles del análisis y diseño de la solución propuesta, así como el procedimiento seguido para llevar a cabo la implementación.

Discusión. En el capítulo de discusión el estudiante analiza y comenta acerca del desarrollo de todo el proyecto. Se incluye una descripción general de la utilidad de las herramientas y tecnologías utilizadas.

Conclusiones y recomendaciones. Los capítulos finales son una recolección de los resultados e ideas obtenidas durante la elaboración de las prácticas supervisadas.

Apéndice. En el apéndice se puede encontrar un glosario de términos técnicos que deberían ayudar al entendimiento de los demás capítulos de este documento; una revisión de los trabajos más importantes que fueron citados; y algunas referencias adicionales que el lector puede encontrar de su interés.

II. Teoría

Este capítulo es una revisión de los principales conceptos teóricos involucrados en el desarrollo del sistema. La primera sección describe la ingeniería del software en general. Se incluyen algunos antecedentes al Desarrollo Rápido de Aplicaciones para facilitar el entendimiento del mismo. Posteriormente se describe la técnica para recolección de requerimientos, llamada KAOS, que fue utilizada. Se dedica una sección al estudio de lo que es un manejador de relaciones con los clientes, qué problemas soluciona y cuáles son los sistemas de este tipo más usados en la actualidad. La última sección está dedicada a describir el software y los lenguajes de programación que se utilizaron como herramientas a través del desarrollo del trabajo de graduación.

A. Ingeniería de software

El economista Howard Baetjer, Jr. (Baetjer, 1998) describe al software como “conocimiento materializado”. Explica que al inicio el software, como el conocimiento, está disperso y en gran medida incompleto. Para mejorar este conocimiento, es necesario un procedimiento de aprendizaje en el que el software interactúe con los diseñadores, con los usuarios y con las herramientas. Este procedimiento, en el que el “conocimiento materializado” es recolectado, depurado y organizado se conoce como el *desarrollo* de software y es la herramienta clave de la ingeniería de software.

Según (IEEE, 2004) la ingeniería de software es:

«La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software. »

Cada proyecto de ingeniería de software desarrollado tiene características únicas, sin embargo es posible definir algunas actividades que aplican a todos los proyectos de software, estas actividades se conocen como el marco de trabajo. (Pressman, 2005)

1. Comunicación
 - Esta es una actividad que se ejecuta durante la duración completa del proyecto. Se refiere a una comunicación y colaboración constante entre todos los interesados en el proyecto.
2. Planeación
 - La planeación es la actividad en la que se establece el plan de trabajo para el proyecto. Este es un plan técnico con las tareas que deben realizarse y los recursos que son requeridos.
3. Modelado
 - Durante el modelado se crean modelos que permiten tanto al desarrollador como a los clientes tener una mejor visión del software y sus requisitos.
4. Construcción
 - Es en esta actividad en la que se crea el código y se prueba para evaluar su calidad.
5. Despliegue (*deployment*)
 - Esta es la actividad de la ingeniería de software en la que el proyecto se entrega al cliente, quien lo evalúa y proporciona una retroalimentación a los desarrolladores.

Estas son las actividades más generales de la ingeniería de software, sin embargo carecen de una definición y una aplicación suficientemente detallada. Es tarea de los modelos de proceso definir este procedimiento para llevar a cabo un proyecto de ingeniería de software. (Pressman, 2005)

1. Modelo clásico. El modelo clásico es también conocido como modelo en cascada o como ciclo de vida del desarrollo de software. Este modelo fue propuesto por Winston Royce en 1970, siendo el modelo más antiguo de la ingeniería de software. Actualmente está en desuso debido a la aparición de técnicas más novedosas; sin embargo aún es utilizado en algunos proyectos. Propone un flujo secuencial de las siguientes actividades: (Kendall, 2005)

- Preparación previa
 - Identificación de problemas oportunidades y objetivos
 - Recolección de los requisitos del cliente
- Análisis
 - Estudio de las necesidades del sistema
- Diseño
 - Diseño lógico del sistema
- Desarrollo
 - Codificar y corregir los errores del programa
 - Desarrollar la documentación del software
- Pruebas
 - Búsqueda de problemas antes de la entrega al usuario
- Implementación
 - El sistema se pone en funcionamiento
 - Se capacita a los usuarios para su uso

a. Comunicación. La actividad de comunicación de la ingeniería de software en el modelo en cascada es más evidente en las primeras etapas al inicio del proyecto. La fase de preparación requiere que el ingeniero de software esté en constante comunicación con los clientes y los demás interesados.

En esta fase se identifican cuáles son los problemas que serán resueltos, delimitando el alcance de la solución de software. Por lo general, los problemas ya fueron identificados, y es por ello que es necesario desarrollar el proyecto. Otra parte muy importante de la fase de preparación es la identificación de objetivos, que es lo que se trata de conseguir con este proyecto. La fase de preparación requiere de herramientas de comunicación como entrevistas con los encargados y la coordinación de los usuarios.

En la recolección de requerimientos, la comunicación es también un factor clave. Para obtener los requisitos específicos se utilizan herramientas interactivas como entrevistas, muestreos, investigación, cuestionarios, etcétera. En esta fase se deben

encontrar los detalles funcionales del sistema actual: la gente involucrada, la actividad del negocio, el entorno en donde se desarrollan las actividades, el momento oportuno y la manera en que se realizan los procedimientos actuales. (Kendall, 2005)

b. Planeación. Aunque en el modelo clásico no hay una fase explícita para la planeación, en todo proyecto es necesaria una definición tentativa de un itinerario con tiempos estimados para las actividades técnicas. Por lo general este itinerario se prepara en la fase de preparación o en la fase de análisis. Esta definición debe ser mejorada en otras fases del proyecto.

Con un plan de proyecto, el cliente puede decir si vale la pena continuar, o si es necesario hacer cambios en la distribución de recursos o en la fecha límite del proyecto. (Kendall, 2005)

c. Modelado. Dentro del modelo clásico de desarrollo de software, la actividad de modelado del marco de trabajo de la ingeniería de software se hace notable en etapas de análisis y diseño. Varias implementaciones del modelo clásico combinan estas dos fases en una por esta razón.

En la ingeniería de software, la actividad de modelado se caracteriza por utilizar como herramientas varios tipos de diagramas, y en el modelo en cascada, las fases de análisis y diseño se apoyan fuertemente en el uso de diagramas de flujo y modelos de UML.

En la fase de análisis se busca realizar una investigación detallada de las necesidades del sistema. Un analista prepara una propuesta de posibles soluciones de software, que sintetiza los resultados del análisis, proporciona información de costo/beneficio de las alternativas, y ofrece recomendaciones de lo que se debe hacer. Con esta propuesta, el cliente puede evaluar de nuevo si considera factible alguna de las

soluciones y decidir si el analista debe seguir adelante. En la fase de diseño del ciclo de vida del desarrollo de sistemas, el diseñador utiliza la información recopilada en las fases previas para realizar un diseño teórico del sistema de información. (Kendall, 2005)

d. Construcción. La actividad de construcción del marco de trabajo está presente en las fases de desarrollo y pruebas.

La fase de desarrollo del modelo clásico consiste en la creación en sí de los programas necesarios. En conjunto con el desarrollo de la aplicación se realiza la documentación interna, pero es tarea del desarrollador trabajar con los usuarios en la documentación externa. Esta documentación externa consiste en manuales, sitios de ayuda, respuestas a preguntas frecuentes, archivos "léame", entre otros. Esta documentación debería ser suficiente para que los usuarios entiendan cómo utilizar el software y lo que deben hacer en caso de que surjan problemas mientras se usa.

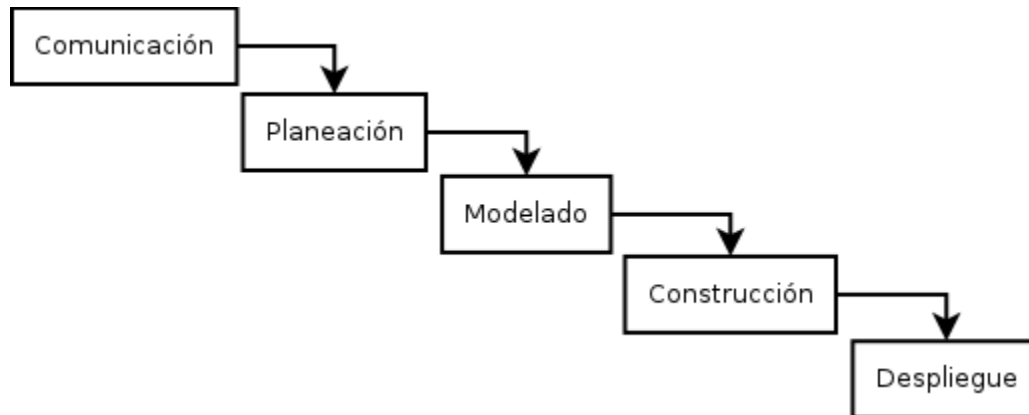
Una parte clave en la construcción de software es la de probarlo. Es mucho menos costoso encontrar los problemas en el sistema antes de entregarlo al usuario. Existen varios tipos de pruebas y en cada una los participantes son distintos.

Dentro de las actividades de construcción del modelo clásico se encuentra el mantenimiento del sistema y su documentación. En el desarrollo el software se crea de manera de que su mantenimiento sea sencillo, sin embargo se trabaja tratando de garantizar que el mantenimiento se mantendrá al mínimo. Es necesaria la construcción de un manual técnico con el que se facilite el mantenimiento en un futuro. (Kendall, 2005)

e. Despliegue. Finalmente, cuando se tiene una solución modelada y construida, el paso que falta es entregarla al cliente. Ésta es la última fase del modelo clásico. Es necesario capacitar a los usuarios en el uso de la herramienta. También es necesario desarrollar un plan para la adopción de la nueva herramienta en la empresa, y si es necesario para la migración de alguna herramienta antigua. En esta etapa se incluye

todo el movimiento de datos, conversión entre formatos, instalación de equipo nuevo y la puesta en funcionamiento del sistema. (Kendall, 2005)

Ilustración 2: Modelo en cascada



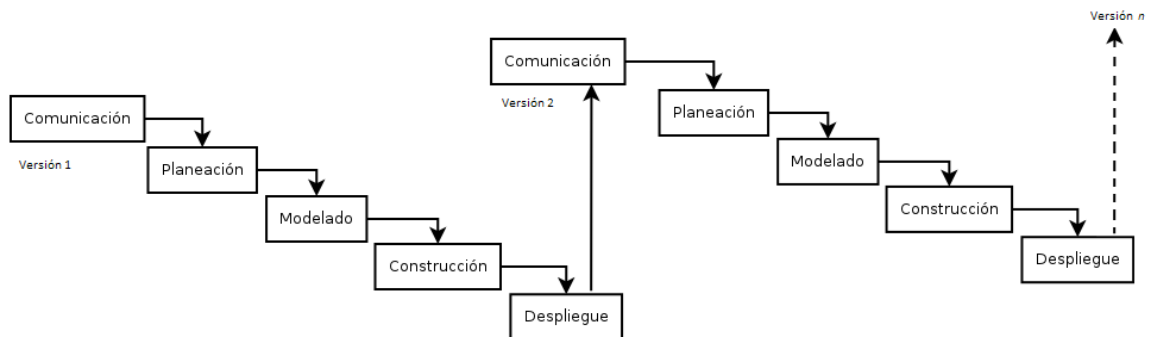
2. Modelo incremental. Se incluye en el documento una descripción del modelo incremental para poder utilizarlo como referencia y comparación con el Desarrollo Rápido de Aplicaciones. El proyecto de la interfaz móvil fue desarrollado utilizando esta técnica, y fue sincronizado con el desarrollo del servidor.

El modelo incremental tiene los mismos pasos que el modelo en cascada, pero aplicados en una manera iterativa. Las actividades de comunicación, planeación, modelado, construcción y despliegue ocurren secuencialmente y al terminar, se inician de nuevo, agregando funcionalidad y características al software que se está produciendo.

El modelo incremental provee en su primera iteración un producto muy básico, pero funcional. Este producto se evalúa o bien se le entrega al cliente, y considerando la retroalimentación que se tenga, se planea la siguiente iteración y se lleva a cabo. Este procedimiento debe repetirse hasta haber completado el software.

El desarrollo incremental es útil cuando el usuario exige una versión funcional en un tiempo corto, o cuando al inicio no se cuentan con los recursos para construir una solución completa. En este último caso, con el modelo incremental es posible iniciar el trabajo con los recursos limitados y luego de las primeras entregas aumentar los recursos que se necesiten. (Pressman, 2005)

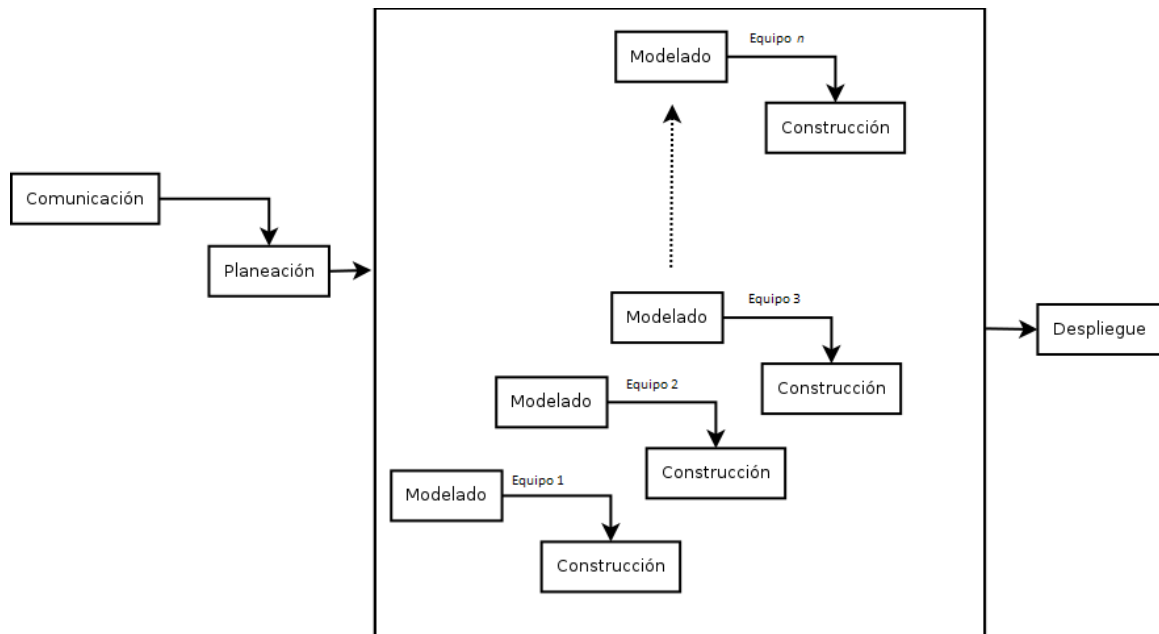
Ilustración 3: Modelo incremental



3. Modelo DRA. Se conoce como DRA al modelo de desarrollo rápido de aplicaciones. Este es un modelo incremental en la que el ciclo de desarrollo es muy corto y con un enfoque orientado a componentes.

En el DRA durante la actividad de comunicación del marco de trabajo se trata de entender el problema y las características que debe incluir el software. La actividad de planeación es vital en este modelo, pues es posible que varios equipos distintos trabajen sobre diferentes funciones del sistema. La actividad de modelado está dividida en tres fases, el de negocios, el de datos y el de proceso. En la construcción, se ensamblan los componentes. Y en el despliegue se crea una base para las iteraciones siguientes. (Pressman, 2005)

Ilustración 4: Desarrollo rápido de aplicaciones



Este modelo fue seleccionado para seguir durante la elaboración del proyecto. Regularmente, DRA es utilizado por equipos de desarrollo, en donde cada equipo puede trabajar en un componente distinto, y se apoya fuertemente en la colaboración entre los equipos para asegurar la adecuada interacción de los componentes. El presente proyecto fue desarrollado por una sola persona, así que la comunicación y la sincronización del desarrollo no fue un problema. Sin embargo, aplicar esta técnica cuando no se está en un equipo de trabajo agrega una dificultad a la coordinación de actividades ya que estas deben ser llevadas a cabo de manera secuencial aún cuando dependen entre sí.

El Desarrollo Rápido de Aplicaciones trata de hacer reutilización de código fabricado previamente, ya sea de componentes desarrollados en proyectos anteriores, o a través de componentes desarrollados por terceras personas que pueden ser adquiridos.

No se contaba con código de proyectos anteriores durante las prácticas profesionales, por lo que se optó por seleccionar herramientas previamente construidas que agregarán

al sistema la mayor funcionalidad posible, y trabajar en el desarrollo de las características faltantes. Una de estas herramientas fue Microsoft SharePoint, la cual provee funciones para abstraer el manejo de la base de datos, la seguridad del sistema y la apariencia gráfica de la aplicación. Además, contiene un sistema de “elementos web”, que pudo ser utilizado para la integración de los componentes desarrollados.

Se escogió DRA sobre el desarrollo incremental pues algunos componentes dependían en gran medida de la información entregada por otros componentes. Además, el proyecto de la aplicación móvil que se estaba trabajando de manera simultánea, necesitaba de la información provista por varios componentes. En un modelo incremental hubiera habido dificultades para mantener disponible esta información.

B. Ingeniería de requisitos

En el campo de las ciencias de la computación, se entiende un requerimiento o requisito como una necesidad que un software debe cumplir (Kendall, 2005). El análisis de requerimientos para el presente proyecto fue desarrollado utilizando la metodología orientada a metas KAOS. En las siguientes secciones, se describe a detalle en qué consiste un análisis formal de requerimientos, cómo funciona la metodología KAOS y por qué fue seleccionada.

1. Definición. La mejor medida del éxito de un proyecto es que tan bien se cumple su propósito. Por lo tanto, la identificación de este propósito debería ser una de las actividades principales en el desarrollo de sistemas. Con el objetivo de enumerar, refinar y analizar estos requerimientos existe la rama de la ingeniería de software llamada ingeniería de requisitos o ingeniería de requerimientos. (LapouChnian, 2005)

Según la IEEE (D. Ross, 1977) la ingeniería de requisitos es:

«Una cuidadosa definición de la necesidad que el sistema debe llenar.»

2. KAOS. La metodología KAOS es un tipo de ingeniería de requisitos orientada a metas. Su nombre proviene de las siglas en inglés para adquisición de conocimiento en especificación automatizada (*Knowledge Acquisition in autOated Specification*). (LapouChnian, 2005)

En esta metodología, se incluyen los siguientes conceptos:

Tabla 2: Conceptos en la metodología KAOS

Concepto	Descripción
Objeto	Cosas de interés en el sistema, que pueden cambiar de forma al pasar de un estado a otro. Los objetos pueden ser entidades, relaciones o eventos.
Operación	Una relación de entrada-salida entre objetos. Con las operaciones también se definen los cambios de estado.
Agente	Un agente es un objeto especial, que actúa como procesador de operaciones.
Meta	Una declaración prescriptiva que el sistema tiene la intención de satisfacer.

C. Software CRM

El nombre de CRM viene de las siglas en inglés para gestión de la relación con clientes, (*Customer Relationship Management*) y se utiliza para referirse al software que desempeña esta tarea. (Goldensoft, 2010)

La comunicación con los clientes es una necesidad de cualquier empresa, y si está bien gestionada puede convertirse en una herramienta, y en una importante ventaja competitiva.

Para gestionar bien todos los eventos de interacción entre la empresa y sus clientes se utilizan comúnmente los programas de CRM. Estos ayudan a tener una visión completa de la relación con cada cliente, facilitando el almacenamiento y seguimiento de los eventos y acciones que giran en torno a ellos. Aumenta la importancia del uso de estas herramientas de gestión cuando existe un número considerable de clientes, mucha rotación del propio personal, o si la comunicación debe recorrer distintas personas o departamentos.

Es posible dividir un CRM en tres módulos generales:

Tabla 3: Módulos de un CRM

Mercadeo	Ventas	Servicios
<ul style="list-style-type: none">• Enfocado a la planificación y manejo de las actividades de mercadeo• Facilita la comunicación con clientes potenciales	<ul style="list-style-type: none">• Enfocado en el equipo de ventas de la empresa• Ayuda a organizar los procesos previos a una venta	<ul style="list-style-type: none">• Enfoca a hacer más efectivo el servicio al cliente• Administra las actividades de servicio enfocándose en el cliente

1. Características. En general, un software CRM debe cumplir con ciertas características generales. El presente proyecto implementa únicamente las características necesarias el módulo de servicios. (Goldensoft, 2010)

a. Registro de los eventos de clientes. Toda la información de contacto de un cliente debe estar registrada y los incidentes que sean reportados, deben ser anotados para ser resueltos por una persona o un grupo. Un sistema bien organizado proporciona

una respuesta satisfactoria al cliente. El presente proyecto implementa esta característica en el componente de incidentes.

b. Gestión de correo electrónico y faxes. Cada fax o correo electrónico entrante o saliente debe quedar registrado junto con el nombre del cliente que lo envió. De esta forma puede ser consultado por cualquier otro usuario. Esta característica es implementada en el proyecto por el componente de incidentes.

c. Control de usuarios integrado. Al acceder al programa cada usuario debe ser identificado mediante un nombre y clave, mostrando el sistema de forma automática sus eventos y acciones pendientes, así como los de su grupo. SharePoint se encarga de la autenticación de usuarios en el proyecto, las actividades pendientes se muestran en la página principal.

d. Usuarios alternativos. En los casos de ausencias temporales de un usuario, debe ser fácil trasladar todos sus asuntos pendientes a otro usuario, consiguiéndose una atención de calidad hacia el cliente. En el sistema implementado, un usuario coordinador o administrador puede reasignar las tareas de los usuarios técnicos en caso de que alguno de estos se ausente. Si el usuario está consiente que va a ausentarse, puede reasignar la tarea personalmente de antemano.

e. Actividades con clientes. Las actividades comerciales y técnicas efectuadas por los respectivos responsables deben quedar almacenadas, para así obtener un control de las acciones realizadas por los usuarios. El sistema implementa todo lo relacionado a las actividades técnicas en el componente de actividades. Lo relacionado a actividades comerciales queda fuera del alcance del proyecto.

f. Información centralizada. Todos los integrantes de la empresa deben tener acceso a la información e historial de cada cliente, las acciones pasadas, incidentes

resueltos o pendientes, datos de contacto, etcétera. Esta característica es implementada en los componentes de cuentas y contactos.

g. Oportunidades de venta. Debe registrar las posibles ventas, aumentando las expectativas de éxito de clientes actuales o potenciales. Todo lo relacionado con ventas queda fuera del alcance del proyecto.

h. Gestión de incidencias. No debe perderse ningún incidente reportado por un cliente; debe ser registrado y clasificado. También podrá documentarse la solución y verificación, incluyendo archivos relacionados. El componente de incidentes se encarga de esta tarea en la solución.

i. Planificación de tareas. Cada usuario o grupo de usuarios debe poder ser asignado a diversas tareas o actividades a desarrollar. El componente de actividades es responsable de esta característica.

j. Encuestas telefónicas. Se debe poder crear un procedimiento a seguir por los agentes de telemarketing para la ejecución de una encuesta y así obtener un resultado estadístico que permita elaborar conclusiones. Todo lo relacionado con el mercadeo queda fuera del alcance del proyecto.

k. Obtención y estudio de resultados. Si la información está registrada se puede elaborar un estudio estadístico que ayude a obtener conclusiones que apoyen la toma de decisiones. Esta se consideró también como una actividad de mercadeo, por lo que queda fuera del alcance del proyecto.

2. Software CRM existente. La empresa estadounidense de investigación Gartner (Gartner, 2009) publica cada año un estudio sobre las tendencias de los software para CRM. A continuación se muestra una tabla en la que figuran las compañías que proveen los software que cubren la mayor parte del mercado (los datos están en millones de dólares).

Tabla 4: Comparación entre software CRM

Empresa	Ganancias 2008	% del mercado	Ganancias 2007	% del mercado	Ganancias 2006	% del mercado
SAP	2,055	22.5%	2,050.8	25.3%	1,681.7	26.6%
Oracle	1,475	16.1%	1,319.8	16.3%	1,016.8	15.5%
Salesforce	965	10.6%	676.5	8.3%	451.7	6.9%
Microsoft	581	6.4%	332.1	4.1%	176.1	2.7%
Amdocs	461	4.9%	421.0	5.2%	365.9	5.6%
Otros	3,620	39.6%	3,289.1	40.6%	2,881.6	43.7%
Total	9,147	100%	8,089.3	100%	6,573	100%

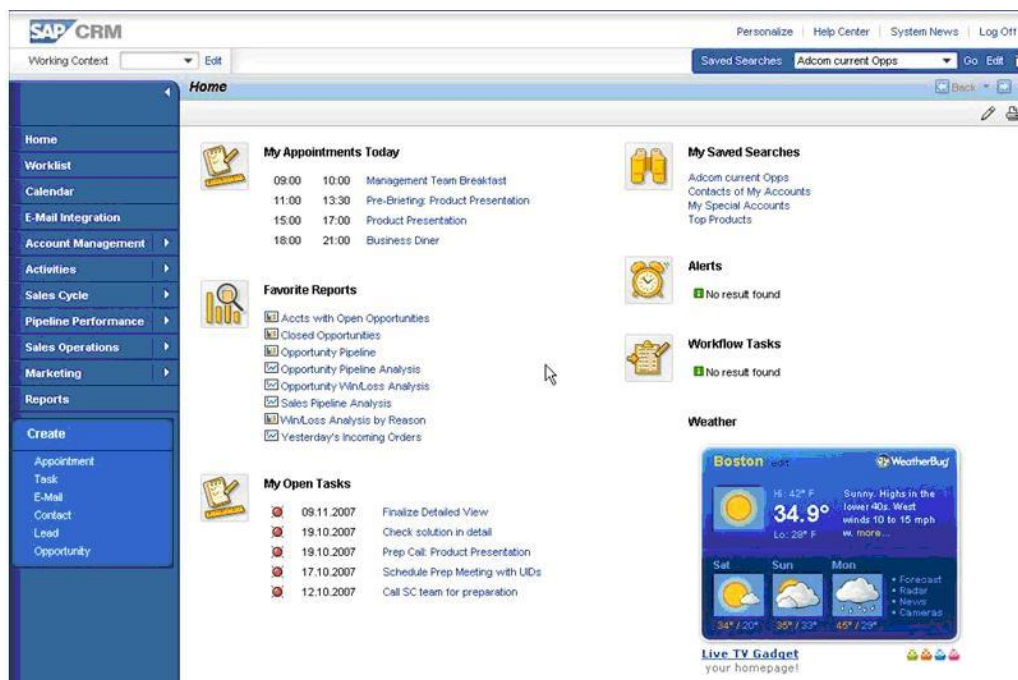
En la tabla se puede observar que el líder del mercado para el año 2008 fue SAP, con casi un cuarto del mercado, a pesar de que en comparación con los años 2006 y 2007 su porcentaje ha disminuido. SAP es seguido de cerca por Oracle y Salesforce, quienes han mantenido un porcentaje constante en los últimos años. Microsoft Dynamics aún no había entrado por completo al mercado y esto se puede notar en su porcentaje de ventas, sin embargo su crecimiento es más alto que sus competidores.

a. **SAP:** El nombre de la empresa viene de las siglas en alemán para Sistemas, Aplicaciones y Productos (*Systeme, Anwendungen und Produkte*). Está catalogada como la tercera empresa independiente más grande proveedora de software del mundo, después de Microsoft y Oracle. Cuenta con 12 millones de usuarios, 100.700 instalaciones y más de 1.500 socios comerciales. A finales de 2005, SAP empleaba a 35.873 personas en más de 50 países y sus ingresos anuales fueron de 8.513 millones de euros. (TEC, 2010)



La solución CRM de SAP se conoce simplemente como SAP CRM. Ha sido utilizada en más de 25 industrias distintas. Entre sus características resaltan el manejo de empresas asociadas y canales de comunicación, mercadeo, ventas, servicios profesionales, un centro de administración interactivo, manejo de ofertas en tiempo real, manejo de canales web, publicidad, y administración de la comunicación del negocio. (TEC, 2010)

Ilustración 5: SAP CRM

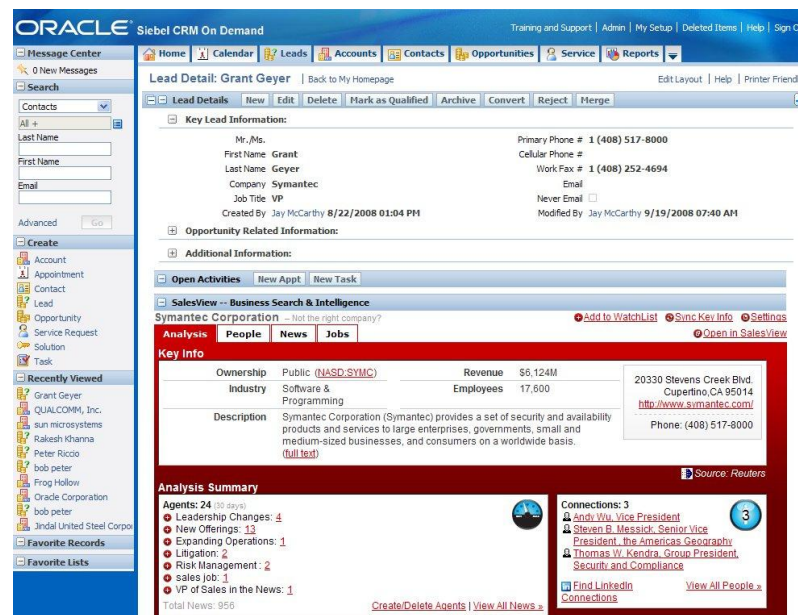


b. Oracle: Oracle Corporation es una de las más grandes compañías de software de todo el mundo, rivalizada solo por Microsoft. Su producto más representativo es su manejador de bases de datos que domina el 33.8% del mercado (Gartner, 2009). La compañía fue fundada en 1977 y actualmente tiene ganancias de casi 11 billones de dólares al año. Es relativamente nueva en el área de CRM, pero ha adquirido uno de los CRM's más populares, Siebel.



La solución CRM de Oracle se conoce como *Oracle Siebel On Demand*. **Oracle Siebel On Demand** fue lanzado recientemente con un enfoque hacia la Web 2.0, siendo una mejora de Siebel. Se integraron características de colaboración y otras características de análisis, así como un centro de contacto, y notas personales y un centro de mensajes. Es posible extender la aplicación a través de applets y widgets, e incluye funcionalidad para la integración con Google, Yahoo y Microsoft SharePoint. Siebel fue adquirido por Oracle en 2005. Es reconocido por haber llegado a dominar el mercado con un 45% de clientes en 2002. Se enfoca en mejorar la toma de decisiones “haciendo llegar la información a la persona correcta en el momento correcto”. Provee análisis predictivo, inteligencia de tiempo real, y flexibilidad, pues soporta extensiones en J2EE y .NET. (TEC, 2010)

Ilustración 6: Oracle Siebel



c. **Salesforce.com**. Es una empresa dedicada a la venta de su software de CRM. Se enfoca en la automatización del mercadeo, del servicio al cliente y de soporte técnico. Su solución tiene características como una arquitectura bajo demanda y herramientas de personalización de un solo clic. Los productos de salesforce.com han ganado varios premios. (TEC, 2010)

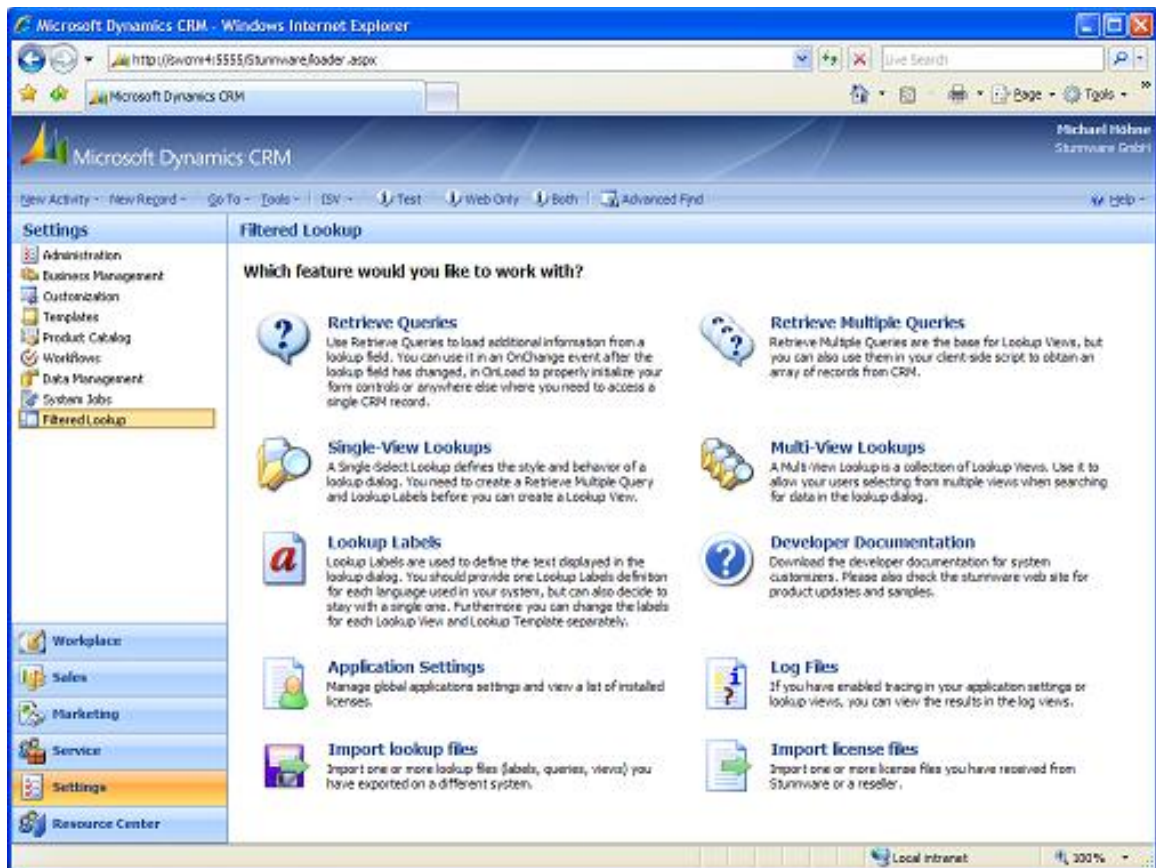


Ilustración 7: Salesforce 8 STA

The screenshot displays the Salesforce 8 STA interface. At the top, there is a navigation bar with the Salesforce logo and the text "Setup System Log Help Logout". Below this is a search bar and a list of tabs: Home, eTriage, Campaigns, Leads, Accounts, Contacts, Opportunities, Reports, Dashboards, New Prospects Snapshots, Documents, and Products. The main content area is titled "(1) Most Active - Last 24 hours" and contains a table of leads. The table has columns for Action, Name, Company, Title, City, State, Email, Phone, Last Visit Date, Visit Count, Page Count, and Profile/Source. The table lists various leads, including Adam Marsh, Alan Ho, Alberto Bacci, and others. On the left side of the interface, there is a sidebar with a search bar, a "Limit to items I own" option, and a "Recent Items" list. At the bottom of the interface, there is a pagination bar showing "1-50 of 232" and "Page 1 of 5".

d. Microsoft: Esta es la compañía de software más grande del mundo. Produce software de casi todos los tipos y los software de CRM no son la excepción. Su CRM se llama **Microsoft Dynamics** y aunque es relativamente nuevo, gana mercado rápidamente, como se puede observar en la Tabla 3. Dynamics es una solución integrada de CRM, y algunas de sus características son una suite de ventas, mercadeo y servicio al cliente. Se ofrece una integración nativa con Office Outlook y otros programas de Microsoft.

Ilustración 8: Microsoft Dynamics



D. Sistemas ERP

Una solución al manejo de relaciones con los clientes más general que un CRM es un ERP. ERP son las siglas en inglés para planeación de recursos empresariales (*Enterprise Resource Planning*). Se refiere a un software que se implementa a través de toda la compañía, y que es usado para gestionar y coordinar todos los recursos de información y funciones de un negocio a través de almacenamiento de datos. Un sistema ERP es toda la infraestructura de software que sostiene los procesos de la compañía.

Regularmente un sistema de ERP maneja en una empresa, la contabilidad, la logística, los mecanismos de distribución, la manufactura, y las ordenes de ventas. Puede ayudar a controlar actividades de negocio como ventas, mercadeo, finanzas, inventario, entrega de productos, producción, control de calidad, entre otras. Adicionalmente, muchos ERP incluyen actualmente una función de inteligencia de negocios, a través de la cual se puede proporcionar la información del sistema a un sistema de apoyo en la toma de decisiones. (wathiserp, 2010)

ERP es un concepto tan general que algunos autores consideran a un sistema CRM como una parte de un ERP, la parte responsable de los clientes. A nivel empresarial, comúnmente se considera a un CRM y a un ERP como software completamente distinto, pues las empresas tienden a separar los procesos internos de la compañía a los procesos externos que tratan con los clientes. Desde este punto de vista, un CRM debe dar una visión completa de los clientes, mientras un ERP debe dar una visión completa de las capacidades del negocio. (Bucholtz, 2010)

Se consideró que implementar un ERP en el tiempo estipulado para las prácticas supervisadas era complicado. Además, muchas de las funcionalidades que estos sistemas proveen no era necesaria para alcanzar satisfactoriamente los objetivos y hubiera aumentado el costo total de la solución. Por lo tanto se decidió construir un CRM, y posteriormente se decidió realizar únicamente el módulo de servicios.

E. Computación en nube

El término computación en nube se refiere a cualquier tecnología que entregue servicios desde un servidor, a clientes en la red. Un servicio en nube tiene características como:

- Se entrega bajo demanda, cobro por tiempo.
- Es flexible, un usuario puede limitar el servicio según quiera.
- Es completamente manejado por el proveedor.

Las empresas buscan soluciones de computación en nube generalmente cuando necesitan aumentar las características de tecnología o agregar nuevas funciones, sin invertir en infraestructura, entrenamiento o licenciamiento de nuevo software. Existen en la actualidad varias aplicaciones de la computación en nube, algunas de ellas se detallan a continuación. (Knorr & Gruman)

1. SaaS. Siglas en inglés para software como un servicio (Software As A Service). Este es un tipo de computación en nube en la que muchos clientes reciben una única aplicación a través del navegador de internet. Un ejemplo de SaaS es la plataforma Google Docs.

2. Computación de utilidades. En este tipo de computación en nube, se proveen recursos como servidores web, espacio de disco, memoria, o procesamiento, de manera virtual a través de la red. Algunas compañías que proveen este servicio son Sun, IBM y Amazon.

3. Servicios web en nube. En este modelo, existen en los servidores proveedores de servicios web, a los que desarrolladores de software pueden acceder a través de internet. Un ejemplo de esto es Google Maps API.

4. PaaS. Siglas en inglés para plataforma como servicio (*Platform As A Service*). Este tipo de computación en nube es una variación de SaaS. Los clientes reciben ambientes de desarrollo como un servicio. Es posible construir aplicaciones que corren en la infraestructura en el servidor y son entregadas a los usuarios por medio del Internet. Un ejemplo de PaaS es Yahoo Pipes.

5. MSP. Siglas en inglés para proveedores gestionados de servicio (*Managed Service Providers*). Se refiere a proveedores de tecnología que asumen la responsabilidad de entregar un servicio al cliente. Un ejemplo de MSP es Karsperky labs que permite hacer un escaneo de antivirus desde el Internet.

6. Plataformas de comercio de servicios. Son plataformas que ofrecen servicios con los que los usuarios pueden interactuar. Un ejemplo es la plataforma Aribaweb provista por la compañía Ariba.

F. Seguridad de la información

Proteger la información enviada entre los componentes fue una tarea delegada a SharePoint, el cual utiliza la seguridad nativa de Windows para ello.

Para proteger la información que se envía a través de la red, se debió escoger alguna plataforma que asegurara que un usuario ajeno a la empresa de servicios no pudiera tener acceso a la información enviada a través del sistema.

1. HTTPS. El nombre HTTPS proviene de las siglas en inglés para protocolo seguro para la transferencia de hipertexto (*Hypertext Transfer Protocol Secure*). Combina las tecnologías de HTTP y TLS (*Transport Layer Security*, se refiere a protocolos de cifrado que proveen seguridad para las comunicaciones sobre internet) para evitar que un intermediario acceda a información sensible enviada entre dos nodos de una red.

Utilizar HTTPS fue la primera opción para proteger la información enviada por el servicio web a los clientes móviles. Sin embargo, para que la seguridad de HTTPS esté completa el servidor web debe estar respaldado por una autoridad de certificación, a la cual los clientes puedan solicitar un certificado que asegure la identidad del mismo. Este es un servicio pagado con el que no se contaba durante el desarrollo del proyecto. Se implementó la comunicación a través de HTTPS sin el respaldo de la autoridad de certificación, pero las pruebas probaron que la seguridad era fácilmente quebrantable.

2. AES. “Advanced Encryption Standard” o estándar de cifrado avanzado es el algoritmo para la codificación de la información aprobado por el departamento de defensa de Estados Unidos. Es ampliamente utilizado y hasta la fecha no hay evidencia de que la seguridad provista por éste pueda ser traspasada.

Cifrar la información que se envía a través de internet con un algoritmo como AES, garantiza que si alguien logra tener acceso a ella, no pueda comprender ni hacer uso de su contenido. Se consideró esta opción luego de que HTTPS se identificara como una opción no viable pero cuando se estudiaron las capacidades de la plataforma móvil donde se desarrollaba el proyecto paralelo, se encontró que no implementaba una manera de utilizar AES. Desarrollar AES manualmente también se consideró, pero esto resultaría en un algoritmo con una mayor exigencia de recursos que utilizar uno previamente implementado.

3. Triple DES. “Triple Data Encryption Algorithm” o algoritmo triple para el cifrado de datos es un algoritmo en el que se aplica tres veces el algoritmo de cifrado conocido como DES. De esta manera se aumenta considerablemente la seguridad.

La plataforma móvil provee acceso a funciones para la codificación de datos con este algoritmo, por lo que se decidió proteger de esta manera la información enviada a través de internet. Adicionalmente se implementó un algoritmo de seguridad para

proteger el nombre de usuario y la contraseña, los detalles pueden encontrarse en el apéndice.

G. Herramientas utilizadas

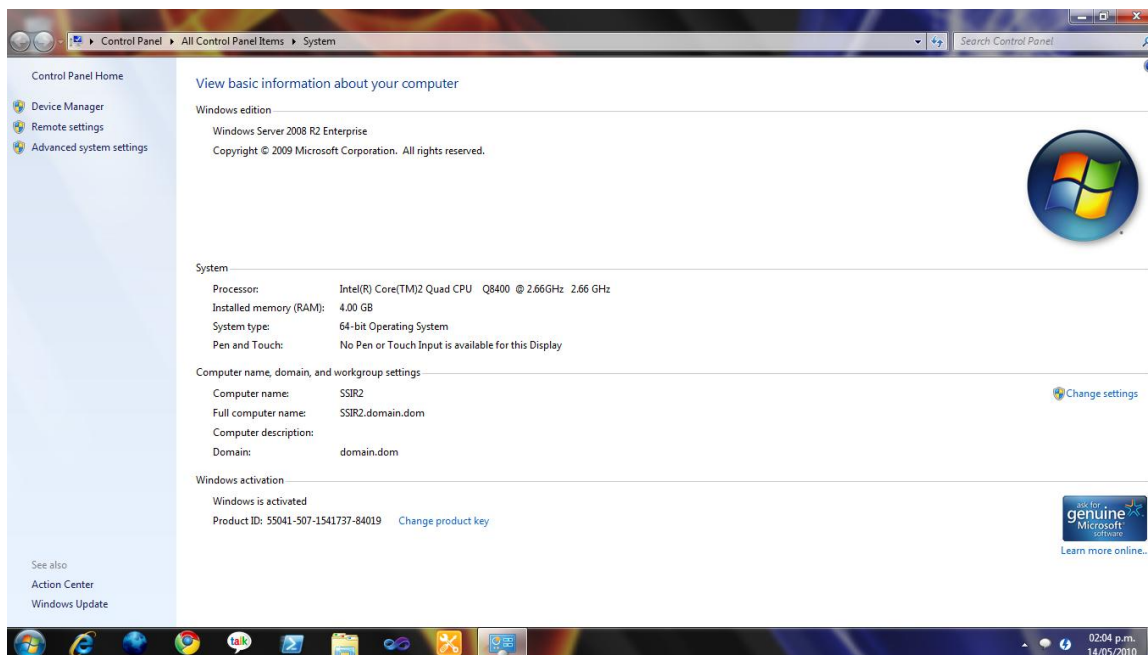
A continuación se describen las herramientas de software más utilizadas para el desarrollo del proyecto. Se incluyen las razones por las que se seleccionaron y el uso que se les dio. Se seleccionó para el desarrollo principalmente software de Microsoft, pues 1800 Software, la empresa en donde se trabajó, cuenta con la calificación de “Gold Partner” por lo que tiene acceso a una gran variedad de licencias, incluyendo productos que aún no están disponibles al público general.

1. Windows Seven. Esta es hasta la fecha la última versión del sistema operativo de Microsoft, Windows, para la línea de computadoras personales. Salió a la venta en Julio de 2009 para reemplazar a su predecesor, Windows Vista. Al inicio del proyecto, se trabajó sobre Windows 7 por requerimiento de 1800 Software y para maximizar la compatibilidad con el resto del software.

2. VirtualBox. VirtualBox es un paquete para la virtualización de sistemas operativos. Fue creado por la compañía alemana Innotek, y luego comprado por Sun Microsystems. Actualmente pertenece a Oracle Corporation, quien adquirió Sun hace algunos años. VirtualBox está disponible de manera gratuita y es una de las mejores plataformas de virtualización en el mercado. Es muy estable y se encuentra en constante mejora gracias al apoyo de la comunidad en el proyecto, por lo que se prefirió este paquete al ofrecido por Microsoft, Virtual PC. Durante el inicio de las prácticas profesionales, se instaló una máquina virtual con Windows Server 2008, que actuó como servidor, mientras se desarrolló en la máquina real de manera “remota”.

3. Windows Server 2008 R2. En la línea de sistemas operativos para servidores, éste es el más reciente lanzado por Microsoft a la fecha. Salió a la venta en Octubre de 2009. Utiliza el mismo núcleo que Windows 7 y fue lanzado únicamente para plataformas de 64 bits. A la mitad de las prácticas profesionales, se cambió el sistema operativo plataforma de Windows 7 a Windows Server 2008 R2. Esto se debió al cambio de la versión de SharePoint, que introdujo nuevos requisitos de hardware que no permitieron emular el servidor. De esta manera, se instaló el servidor como sistema operativo nativo y se desarrolló en él.

Ilustración 9: Máquina de desarrollo



4. SQL Server 2008. SQL Server es la solución de Microsoft para el manejo de bases de datos. Se seleccionó pues era requisito de software para SharePoint, quien no soporta integración con otros servidores de bases de datos. Normalmente, se dedica un servidor distinto para la base de datos y para la aplicación, pero gracias a las características de hardware de la máquina de desarrollo, se pudo utilizar para ambas tareas.

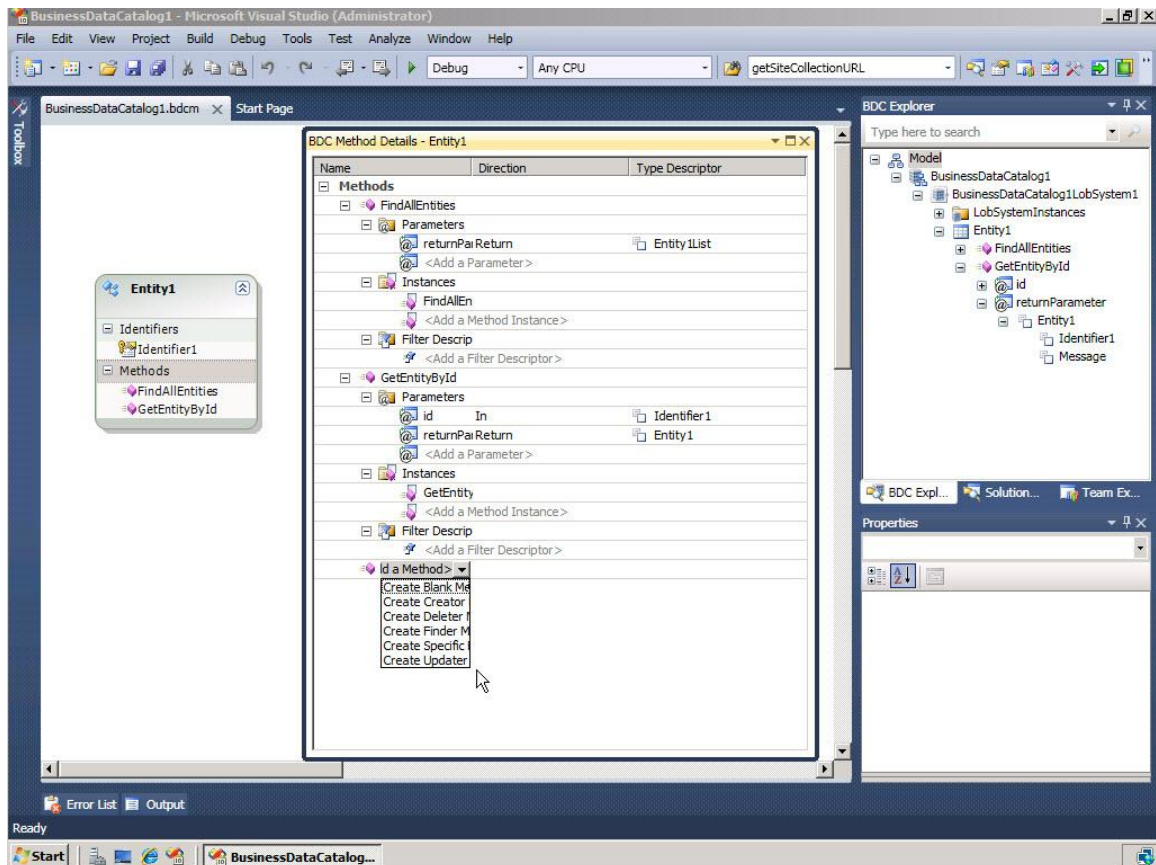
5. IIS. Llamado así por las siglas en inglés para Servicios de Información de Internet, (*Internet Information Services*). Es un servidor web creado por Microsoft y el segundo más utilizado (después de Apache) con el 24.47% de todos los sitios en internet según (Netcraft, 2010). IIS se utilizó para el despliegue y publicación de ambas interfaces web, así como del sitio de administración. Este es también un requisito de software de SharePoint.

6. Microsoft Office. Es una suite de herramientas de procesamiento de texto y datos interrelacionadas. Es desarrollada por Microsoft y la líder en el mercado de su tipo. Se utilizó durante el desarrollo para la creación de la documentación y para la visualización de documentos. Al inicio se trabajó con Office 2007, pero a la mitad del desarrollo se cambió a Office 2010 al migrar el resto de la plataforma. Es capaz de interconectarse con SharePoint para agregarle funcionalidades y facilidades para los usuarios.

a. SharePoint Designer. Este es un editor especializado de HTML para el diseño web. Se caracteriza porque permite editar los sitios y páginas de SharePoint. Al inicio de las prácticas, se trabajó con Microsoft Office SharePoint Designer (el cual era parte de Microsoft Office 2007), y luego se migró a la nueva versión, Microsoft SharePoint Designer, la cual tenía una mejor compatibilidad al cambiar a SharePoint 2010.

7. Visual Studio. Es el IDE para el desarrollo creado por Microsoft. Con Visual Studio es posible desarrollar aplicaciones de consola, interfaz gráfica, sitios web, aplicaciones web y servicios web que pueden ejecutarse sobre el Framework de .NET. Se caracteriza por ser un editor de código con un depurador perfectamente integrado, y por su amplia gama de herramientas. Soporta una gran cantidad de lenguajes de programación. Entre los que se soportan por defecto están: C/C++ (Visual C++), VB.NET (Visual Basic .NET), C# (Visual C#), and F#. Visual Studio 2010 agrega soporte para el desarrollo de aplicaciones para SharePoint, por lo que esta herramienta fue una selección ideal como el IDE para el desarrollo.

Ilustración 10: Visual Studio 2010



8. Active Directory. Es una tecnología creada por Microsoft para la administración de usuarios. Contiene una variedad de servicios de red, incluyendo:

- LDAP (Lightweight Directory Access Protocol): Es un protocolo de internet utilizado para solicitar y modificar data desde un servidor de nombres.
- Kerberos: Un protocolo para la autenticación en red, que permite a nodos comunicándose sobre un canal no seguro, intercambiar información de manera segura utilizando autenticación mutua entre cliente y servidor.
- DNS (Domain Name System): Sistema que asocia “nombres de dominio” a nodos en una red, y provee mecanismos para traducir estos nombres a identificadores numéricos.
- Administración de red y delegación de autoridad
- Almacenamiento central de datos
- Sincronización entre servidores

El manejo de usuarios del proyecto se llevó a cabo utilizando los servicios de Active Directory. Varios de los componentes del sistema se conectan directamente a estos servicios y agregan funcionalidades al sitio.

9. Internet Explorer. Se refiere a una serie de navegadores gráficos para internet. Durante las prácticas, se realizaron las pruebas del sitio necesarias utilizando Internet Explorer 8, por lo que es posible garantizar que este navegador es compatible con todas las funciones. También se realizaron pruebas con los navegadores Mozilla Firefox y Google Chrome, pero algunas funcionalidades menores presentan problemas de despliegue. Estos problemas de despliegue son causados por la falta de compatibilidad de SharePoint, por lo que resolverlos queda fuera del alcance.

10. SharePoint. SharePoint es un servidor web que agrega una gran cantidad de herramientas a IIS para la generación de páginas web dinámicas. Se seleccionó para proveer tres abstracciones al sistema:

- Interfaz web y procesar las solicitudes de HTTP y HTTPS.
- Una capa de aplicación, que provee funcionalidades como búsqueda y servicios de conexión con otras aplicaciones como Excel.
- Servicio de identificación de usuarios seguro.
- Manejo de base de datos.

Los sitios creados con SharePoint son esencialmente aplicaciones web de ASP.NET, que son servidos por IIS y que utilizan una base de datos SQL Server. Todo el contenido del sitio está almacenado en la base de datos, inclusive las páginas y archivos.

Gracias a las abstracciones provistas por SharePoint, fue posible concentrar el diseño y desarrollo en la funcionalidad, pues no fue necesario diseñar una base de datos desde cero, e implementar módulos para el manejo de usuarios. También gracias a la plataforma de integración que provee, se pudo abstraer el manejo de seguridad al nivel interno de la aplicación.

Al inicio de las prácticas, se desarrolló la solución sobre SharePoint 2007, pero a la mitad, salió al mercado SharePoint 2010, y fue necesario migrar todo lo desarrollado a una nueva plataforma. En el proceso, la apariencia y estructura del proyecto cambió

considerablemente, por lo que se le dio un nuevo nombre y se consideró como una nueva versión.

Ilustración 11: SharePoint 2007

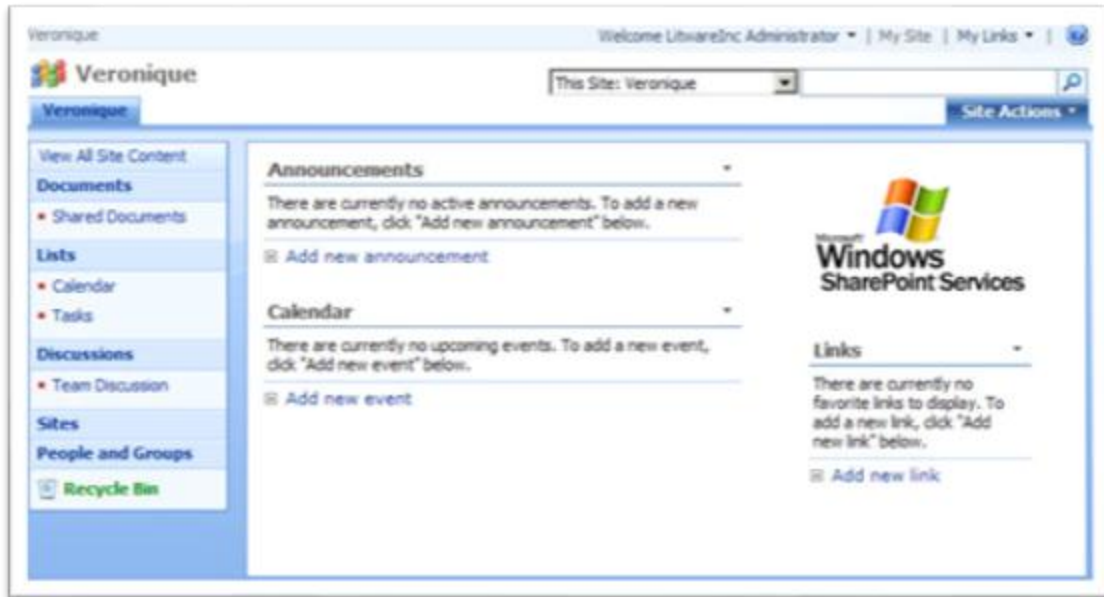
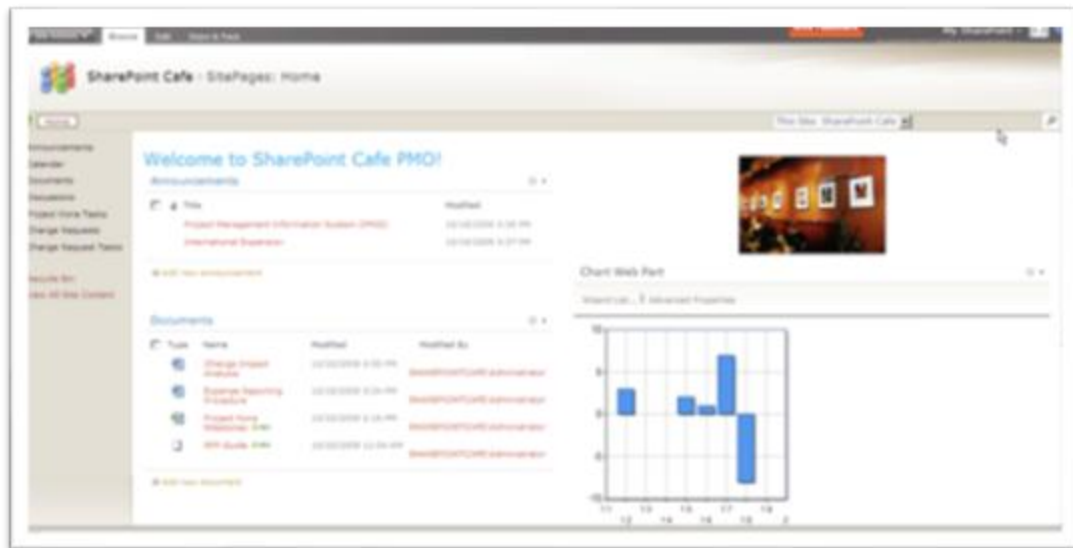


Ilustración 12: SharePoint 2010



III. Planteamiento del proyecto

Con el objetivo de ayudar a las empresas de servicio a proveer un mejor servicio al cliente, se propone la creación de un sistema en el que se pueda centralizar la información de los clientes, y desde donde se pueda coordinar las actividades de los proveedores de servicio. Este sistema debe ser adaptable a las necesidades específicas de la empresa que lo implemente.

Ya que la elaboración de un sistema de CRM completo resultaría en un producto más costoso para las empresas de servicios de tamaño mediano, esta solución consiste en la construcción de únicamente el módulo de servicios.

Para complementar la funcionalidad de este proyecto se desarrolla de manera simultánea, pero como un proyecto separado, una aplicación móvil que facilita el acceso y actualización de la información almacenada en el sistema principal a los técnicos. Los técnicos son las personas que están a cargo de realizar las tareas de servicio.

A. Metas

A través de este proyecto se espera que a largo plazo se puedan alcanzar las siguientes metas:

- Haber creado un software:
 - que ayude a una empresa a proveer un mejor servicio al cliente.
 - que pueda recolectar, ordenar e integrar la información de los clientes de una empresa.
 - que ayude a una empresa a entender mejor las necesidades de sus clientes, y a través de ello, a mantener relaciones por periodos largos.

- que facilite la administración del servicio al cliente, llevando control de las necesidades de éstos, y las actividades que se lleven a cabo para cumplirlas.
- Haber creado una alternativa a los software CRM actuales que las empresas medianas puedan obtener a un precio bajo.
- Haber creado una solución muy flexible y expandible, que pueda adaptarse a las necesidades de muchas empresas en el área de servicios.

B. Objetivos

1. Objetivo general

- Diseñar y construir el módulo de servicios de un software para el manejo de las relaciones entre una empresa y sus clientes (CRM).

2. Objetivos específicos

- Identificar las características de un software CRM que forman parte del módulo de servicios.
- Determinar los componentes generales del módulo de servicios de un software CRM.
- Diseñar cada uno de los componentes que forman parte del módulo de servicios de un software CRM.
- Diseñar una interfaz para la comunicación entre la solución y aplicaciones externas.

IV. Desarrollo

A continuación se presenta un capítulo dedicado a mostrar el proceso seguido para la implementación del sistema para el manejo de relaciones con los clientes de una empresa de servicio. La metodología de software utilizada fue Desarrollo Rápido de Aplicaciones. En la primera sección se detalla el método seguido para la elaboración del sistema. La segunda sección sigue el procedimiento de análisis general que se llevó a cabo, éste es un análisis de alto nivel del sistema y sólo muestra las características más generales. Posteriormente se encuentra una sección que trata el diseño del sistema. El siguiente paso en la metodología DRA es la implementación, la cual implica un pequeño análisis y diseño de cada uno de los componentes del sistema, se dedica una breve sección para cada uno de estos componentes. Finalmente se incluye una sección que explica a grandes rasgos las metodologías seguidas para probar el software. Cada una de las secciones anteriormente mencionadas incluye una discusión acerca de los resultados obtenidos y la eficacia de las herramientas utilizadas.

A. Método

El método se refiere al procedimiento seguido para ejecutar el proyecto; los sujetos involucrados y las herramientas utilizadas.

1. Sujetos. Para la elaboración del trabajo, fue necesario mantener comunicación con una persona con experiencia previa en el tema de CRM.

a. Gerente de ventas. Conocedor del área de desarrollo de proyectos, y con una gran experiencia y habilidad en cuanto al trato con los clientes, tanto individuales como empresariales. Su aporte fue clave en la estructuración del proyecto, y como guía en la selección de herramientas. La comunicación se llevó a cabo principalmente a través de reuniones y entrevistas periódicas.

2. Herramientas. Para el desarrollo del trabajo de graduación fueron necesarias las siguientes herramientas.

a. Herramientas para el diseño de modelos. Estas herramientas tienen como objetivo apoyar en el diseño de cada uno de los componentes. Se utilizaron para la representación informal y posteriormente para la representación formal del sistema. No solamente se utilizaron programas de software como herramientas.

b. Herramientas para la programación. Para poder llevar a cabo el objetivo de construir e integrar los componentes, fueron necesarias herramientas de desarrollo y lenguajes de programación. Para más información, referirse a la sección de **Herramientas utilizadas**, en el capítulo de **Teoría**.

c. Hardware. Fue necesaria una máquina para actuar como servidor y como máquina de desarrollo. También se hizo uso de la infraestructura de red existente en la empresa.

3. Procedimiento

- Se realizó una investigación bibliográfica acerca del software CRM.
- Se realizó una exploración práctica de **Microsoft Dynamics**.
- Se realizaron reuniones con el gerente de ventas de la empresa.
- Se recolectó la lista de características generales con las que cumple un software CRM.
- Se definieron las características necesarias para construir el módulo de servicios.
- Se llevó a cabo un diseño arquitectónico de la solución, incluyendo el componente proveedor del servicio web y la integración con el proyecto de la interfaz móvil.
- Se construyeron de manera simultánea cada uno de los componentes.
- Se crearon los manuales técnicos y de usuario.
- Se analizaron y discutieron los resultados.
- Se redactó el documento final.

4. Discusión. El proyecto fue propuesto y llevado a cabo en una empresa de desarrollo de software, por lo tanto en realidad no se contaba con un cliente específico, y se quería obtener una solución que pudiera adaptarse a muchos tipos distintos de clientes. Al no tener un cliente, fue muy útil mantener una estrecha conversación con el gerente de mercadeo, quien en realidad tenía mucha experiencia con el trato de clientes y estaba consiente de las necesidades que estos tenían.

Las herramientas de software que se utilizaron fueron en su mayoría producidas por la compañía Microsoft. Esto facilitó la integración de los componentes del sistema, pero introdujo algunos problemas. Por ejemplo, todas las soluciones son costosas en cuanto a recursos, y a pesar de que la máquina de desarrollo era de “última generación” no contaba con las capacidades de un servidor. Esto dificultó el desarrollo en algunas ocasiones pues el tiempo de respuesta era algo lento. En particular, la plataforma SharePoint exige más de 4 gigabytes de memoria en el sistema, y hace uso de varias instancias de SQL Server que llegaban a consumir la capacidad total del procesador (Intel Core 2 Quad Q8400 @ 2.66 GHz). Se dieron muchos otros problemas en torno a SharePoint, que, aunque era aparentemente amigable para los desarrolladores, muchas de sus funciones no cumplían la tarea que se esperaba. Así mismo, SharePoint Designer (en particular la versión 2003) era una herramienta inestable y difícil de manejar. La documentación provista por Microsoft para el uso de estas herramientas es amplia pero resultó complicado encontrar información puntual. Esto se debe a que se concentra en explicar la funcionalidad y características de las herramientas, pero no su modo de uso. El proyecto sufrió un cambio de plataforma completo a la mitad del trabajo, y SharePoint en realidad no proveía servicios para la migración por lo que fue trabajoso hacerla manualmente. Luego de cambiar el sistema de SharePoint 2007 a SharePoint 2010, encontrar documentación en el tema fue mucho más difícil, pues el software acababa de ser lanzado y aún ni siquiera estaba disponible al público general.

B. Análisis

El análisis general es la primera representación técnica del sistema. A continuación se definen los requerimientos del sistema, las características de los datos, y un modelado de escenarios basado en casos de uso. (Pressman, 2005)

1. Modelo de requerimientos. La metodología utilizada para la creación del modelo de requerimientos del sistema servidor se conoce como KAOS. Ésta es una metodología para la recolección de requerimientos orientada a metas. Se seleccionó esta técnica por que se tenía una gran libertad en el diseño por la falta de un cliente específico.

En esta técnica se crean metas funcionales basadas en los objetivos, y se conectan entre sí, de manera que cada meta general sea justificada por metas más específicas. Siguiendo esta técnica, se definieron, las siguientes metas funcionales de nivel alto:

- Crear un sistema que mejore el servicio al cliente en una empresa
- Crear un sistema que satisfaga los requerimientos
- Crear un sistema que se pueda acceder desde una plataforma web
- Crear un sistema que provea un servicio web
- Crear un sistema transaccional
- Crear un sistema usable
- Crear un sistema multiplataforma

También se definieron las siguientes entidades y agentes tras las investigaciones bibliográfica y práctica, y tras las entrevistas:

- Entidades (Se incorporan al sistema de manera indirecta a través de los agentes):
 - Cuenta
 - Una cuenta es una empresa a la que se le presta un servicio.
 - Contratista
 - Un contratista es una empresa subcontratada para prestar un servicio.

- Contacto
 - Un contacto es una persona individual, con nombre y apellido, que puede estar relacionado a una cuenta o a un contratista, o ser un cliente independiente.
- Contrato
 - Un contrato es un documento con el que se asocia la empresa de servicios con una cuenta, un contratista, un contacto y un grupo de actividades de servicio.
- Material
 - Es un instrumento necesario para realizar una actividad, que se gasta, por ejemplo, cable.
- Herramienta
 - Es un instrumento necesario para realizar una actividad que puede ser reutilizado.
- Actividades
 - Una actividad es una tarea de servicio que debe realizarse. Tareas complicadas pueden ser desglosadas en sub actividades. Las actividades pueden ser periódicas, y pueden necesitar materiales y herramientas.
- Incidentes
 - Un incidente es una solicitud de un servicio por parte de un cliente. Puede ser parte de un contrato y está compuesto por actividades.

- Agentes (Tienen acceso directo al sistema):
 - Usuarios técnicos
 - Un técnico es una persona interna a la empresa que presta un servicio.
 - Usuarios coordinadores
 - Los usuarios coordinadores están en comunicación directa con los clientes, y tienen la capacidad de ordenar a los usuarios técnicos a realizar tareas. También pueden realizar tareas ellos mismos.
 - Usuarios administradores
 - Un administrador tiene acceso total al sistema.

Se definieron también las siguientes operaciones :

1. Los tres agentes pueden buscar, listar, detallar, crear, editar y eliminar:
 - a. Cuentas
 - b. Contactos
 - c. Contratistas
 - d. Contratos
 - e. Incidentes
 - f. Actividades
2. Los Administradores pueden gestionar a otros usuarios.
3. Los Administradores pueden cambiar la apariencia global del sitio.
4. Los tres tipos de usuario pueden cambiar la apariencia de su página principal personal.

a. Requerimientos. Siguiendo los pasos del análisis, se llegó a los siguientes requerimientos:

- El sistema debe proveer las funcionalidades necesarias para dar mantenimiento a las tareas de los proveedores de servicio.
- El sistema debe ser capaz de administrar los siguientes grupos de usuarios:
 - Usuarios técnicos
 - Usuarios coordinadores
 - Usuarios administradores
- El sistema debe proveer las funciones necesarias para el mantenimiento de las cuentas asociadas a la empresa.
- El sistema debe proveer las funciones necesarias para el mantenimiento de los contratistas asociados a la empresa.
- El sistema debe proveer las funciones necesarias para el mantenimiento de los contratos que pueden estar asociados a una cuenta, a un contratista o a un cliente.
- El sistema debe proveer las funciones necesarias para el mantenimiento de los materiales y herramientas que pueden estar asociados a una actividad.
- El sistema debe proveer las funciones necesarias para el mantenimiento de los incidentes, que están asociados a una cuenta, un contacto, un contrato o un contratista.
- Un usuario debe ser capaz de crear y asignar nuevas tareas.
- Un usuario debe ser capaz de reportar incidentes.

2. Modelo de datos. En el modelado de datos se definen todos los objetos de datos que se procesan dentro del sistema, y las relaciones entre los objetos.

Un *objeto de datos* es una representación de cualquier información que el sistema debe entender. Esta información tiene muchas propiedades, que se llaman *atributos*. Los objetos se relacionan entre sí a través de *relaciones*.

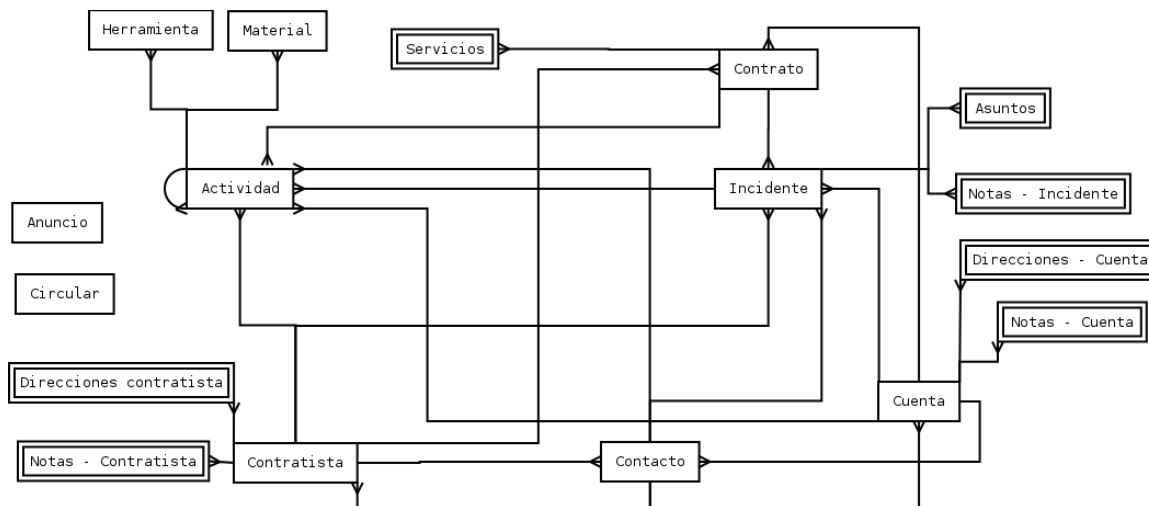
Tabla 5: Entidades principales del sistema

Objeto de datos	Atributos
<p>Actividad</p> <p><i>Una actividad es una tarea de servicio que un usuario del tipo "técnico" debe llevar a cabo.</i></p>	<ul style="list-style-type: none"> ▪ Título ▪ Ubicación ▪ Hora de inicio ▪ Hora de finalización ▪ Detalle ▪ Categoría ▪ Periodicidad ▪ Tipo de actividad ▪ Cuenta ▪ Contratista ▪ Contacto ▪ Responsable(s) ▪ Prioridad ▪ Número de teléfono ▪ Correo electrónico ▪ Entrante/saliente ▪ Porcentaje completado ▪ Actividad principal ▪ Contrato ▪ Incidente ▪ Estado
<p>Contacto</p> <p><i>Un contacto es una entidad con un nombre y un apellido.</i></p>	<ul style="list-style-type: none"> ▪ Nombre de contacto ▪ Apellido ▪ Puesto ▪ Cuenta ▪ Contratista ▪ Teléfono de trabajo ▪ Teléfono personal ▪ Fax ▪ Localizador ▪ Correo electrónico ▪ Nombre dirección ▪ Calle 1 ▪ Calle 2 ▪ Calle 3 ▪ País/región ▪ Estado/provincia ▪ Ciudad ▪ Código postal ▪ Observaciones
<p>Cuenta</p> <p><i>Una cuenta es una entidad financiera a la que se le presta un servicio.</i></p>	<ul style="list-style-type: none"> ▪ Nombre de cuenta ▪ Código de cuenta ▪ Nombre fiscal ▪ Tipo de cuenta ▪ Cuenta principal ▪ Contacto principal ▪ Identificador fiscal 1 ▪ Identificador fiscal 2 ▪ Central telefónica ▪ Otro número telefónico ▪ Número de fax ▪ Sitio web ▪ Correo electrónico ▪ Nombre dirección ▪ Calle 1 ▪ Calle 2 ▪ Calle 3 ▪ Ciudad ▪ Estado/provincia ▪ Código postal ▪ País/región ▪ Observaciones ▪ Asesor asignado

Continuación tabla 5: Entidades principales del sistema

Objeto de datos	Atributos
Contratista <i>Un contratista es una empresa subcontratada para realizar actividades.</i>	<ul style="list-style-type: none"> ▪ Nombre de contratista ▪ Código de contratista ▪ Nombre fiscal ▪ Identificador fiscal 1 ▪ Identificador fiscal 2 ▪ Central telefónica ▪ Otro número telefónico ▪ Número de fax ▪ Sitio web ▪ Correo electrónico ▪ Nombre dirección ▪ Calle 1 ▪ Calle 2 ▪ Calle 3 ▪ Ciudad ▪ Estado/provincia ▪ Código postal ▪ País/región ▪ Observaciones ▪ Asesor asignado ▪ Contacto principal ▪ Tipo de contratista ▪ Contratista principal
Contrato <i>Un contrato es un acuerdo entre la empresa y alguno de sus clientes.</i>	<ul style="list-style-type: none"> ▪ Descripción ▪ Cuenta ▪ Contratista ▪ Inicio del contrato ▪ Fin del contrato ▪ Detalle
Incidente <i>Un incidente es una solicitud de servicio</i>	<ul style="list-style-type: none"> ▪ Descripción ▪ Tipo de incidente ▪ Contrato ▪ Estado ▪ Prioridad ▪ Fecha de apertura ▪ Fecha de cierre ▪ Solución ▪ Contacto ▪ Cuenta ▪ Contratista ▪ Porcentaje completado ▪ Responsable(s)
Herramienta <i>Un objeto reusable.</i>	<ul style="list-style-type: none"> ▪ Equipo ▪ Tipo de equipo ▪ Unidades disponibles
Material <i>Un consumible</i>	<ul style="list-style-type: none"> ▪ Material ▪ Tipo de material ▪ Unidades disponibles ▪ Unidad de medida
Circular <i>Una circular es un mensaje o documento enviado a varios destinatarios.</i>	<ul style="list-style-type: none"> ▪ Título ▪ Cuerpo ▪ Vencimiento ▪ Confidencial ▪ Destinatarios
Anuncio <i>Un anuncio es un mensaje o documento que llega a todos los usuarios.</i>	<ul style="list-style-type: none"> ▪ Título ▪ Cuerpo ▪ Vencimiento

Ilustración 13: Relaciones entre las entidades del sistema



3. Modelo de escenarios. Los diagramas de *casos de uso* representan las interacciones entre los usuarios y el sistema. Cada caso de uso es simbolizado en los diagramas como una elipse y representa un escenario de interacción posible con el sistema. Cada tipo de usuario es representado por una figura humana.

En los siguientes casos de uso se asume que:

- “Usuario” se refiere a los usuarios técnicos, coordinadores y administradores.
- “Administrar” se refiere a las actividades de buscar, listar, mostrar, agregar, modificar y eliminar.
- “Elementos” se refiere a cualquiera de las entidades del sistema.

Ilustración 14: Casos de uso, todos los usuarios

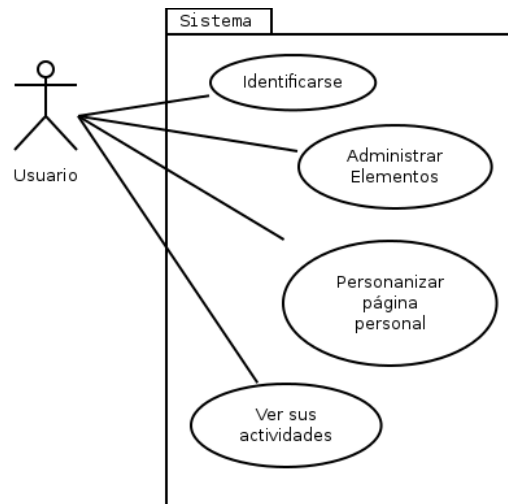


Ilustración 15: Casos de uso, coordinadores y administradores

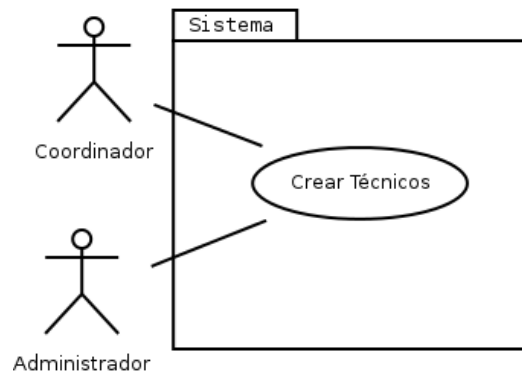


Ilustración 16: Casos de uso, administradores

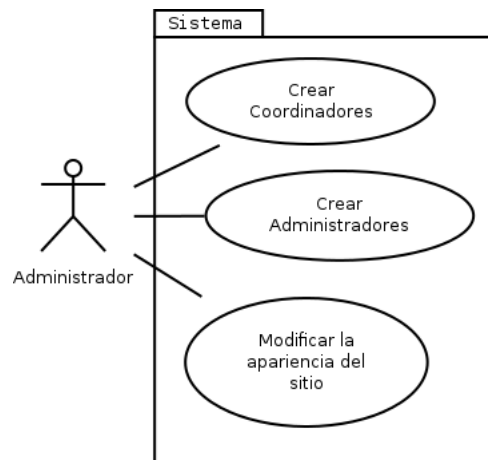
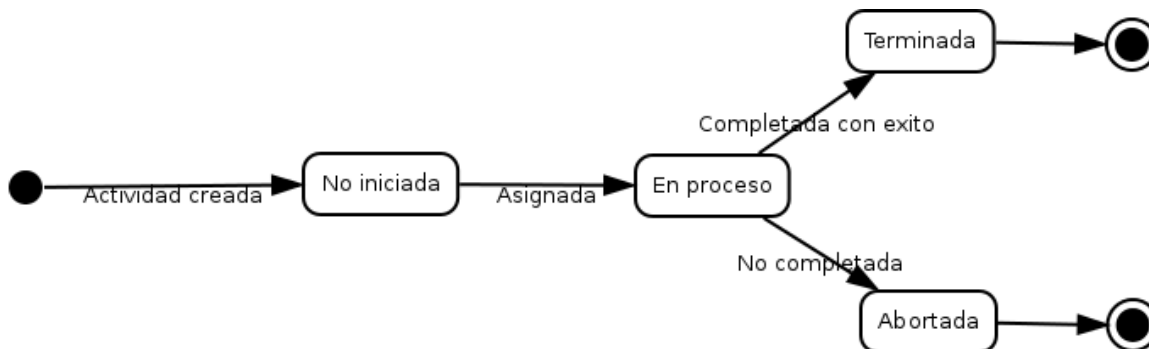


Ilustración 17: Diagrama de estados, Actividades



4. Usuarios del sistema. Dentro del sistema desarrollado existen tres tipos distintos de usuarios con diferentes funciones y permisos. El modelo del sistema fue construido de manera que Los usuarios en un nivel de la jerarquía, tienen los permisos de todos los usuarios en jerarquías menores.

Tabla 6: Usuarios del sistema

Usuario	Permisos
Técnico	<ul style="list-style-type: none">▪ Ver, modificar y eliminar el contenido de las Listas.
Coordinador	<ul style="list-style-type: none">▪ Coordinadores de usuarios.▪ Ver, modificar y eliminar el contenido de las Listas.▪ Crear contratistas y crear otros coordinadores y técnicos.
Administrador	<ul style="list-style-type: none">▪ Pueden crear cualquier tipo de usuarios.▪ Tienen control sobre todas las funcionalidades del sitio, incluyendo su apariencia y los campos de las listas.

5. Discusión. La técnica seleccionada para la obtención de requerimientos encajó muy bien con el planteamiento del problema. El resultado fueron los requerimientos puntuales que pudieron ser traducidos al diseño (en particular a los casos de uso) de una manera simple.

El resto del análisis general consistió principalmente en la definición de los datos. Aunque SharePoint proveía una capa de abstracción sobre la base de datos, analizar las características de todos los elementos del sistema fue un ejercicio muy similar al de diseñar una base de datos. Cada una de las entidades definidas debía ser trasladada a una lista de SharePoint, en donde sus atributos son representados por los campos. Era necesario prestar atención a los tipos de datos en cada atributo, y en ocasiones se tuvieron que definir estos tipos de datos. La información de cada una de estas entidades debía de ser provista al servicio web, quien actúa tomando el rol de un usuario técnico desde el punto de vista del sistema. La herramienta utilizada para realizar los diagramas se conoce como DIA y es un programa de código abierto sobre la licencia GPL. Fue muy simple de usar y muy amigable; además de ser muy completo.

Se consideraron otros diagramas para el análisis, como los diagramas de flujos de datos y los diagramas de carril. Sin embargo, SharePoint es responsable del flujo de datos internos del sistema, y los usuarios ingresan información directamente a las listas y a SharePoint en general (como en el caso de la autenticación), por lo que se decidió que no eran necesarios en esta etapa. De todas las entidades, la única que cambia de estados a través del tiempo son las actividades. Una actividad es creada por un contratista y asignada a un técnico. El técnico puede ver la actividad desde su página principal, o desde las vistas de actividades y cambiar su estado en cualquier momento. Si la actividad pasa su fecha de finalización y su estado aún es “en proceso” el sistema la considera como una actividad retrasada y la mostrará en la página principal de los responsables hasta que su estado cambie o sea eliminada.

C. Diseño

El diseño es el núcleo técnico de la ingeniería de software. Esta es la última actividad de modelado y establece una plataforma para iniciar con la generación de código. (Pressman, 2005)

1. Diseño arquitectónico. La arquitectura de un sistema de software es la estructura del sistema que incluye los componentes, las propiedades visibles externamente, y las relaciones entre ellos.

Tabla 7: Arquitectura estratificada

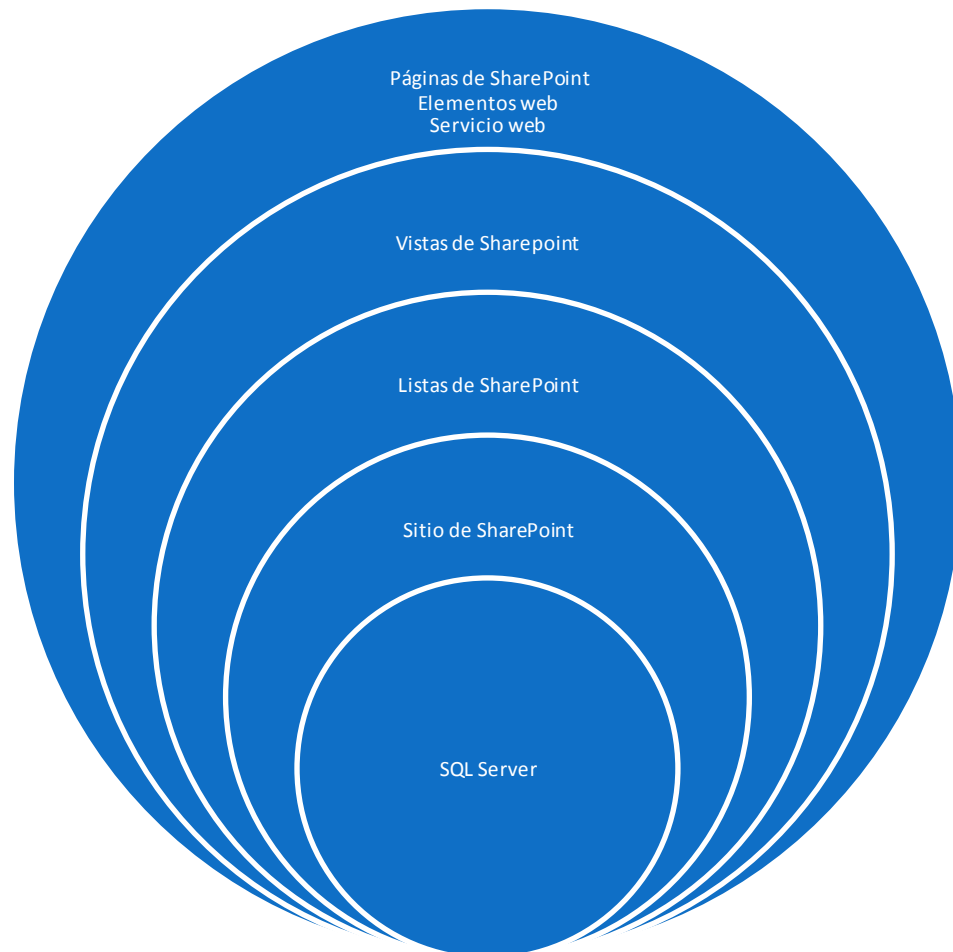
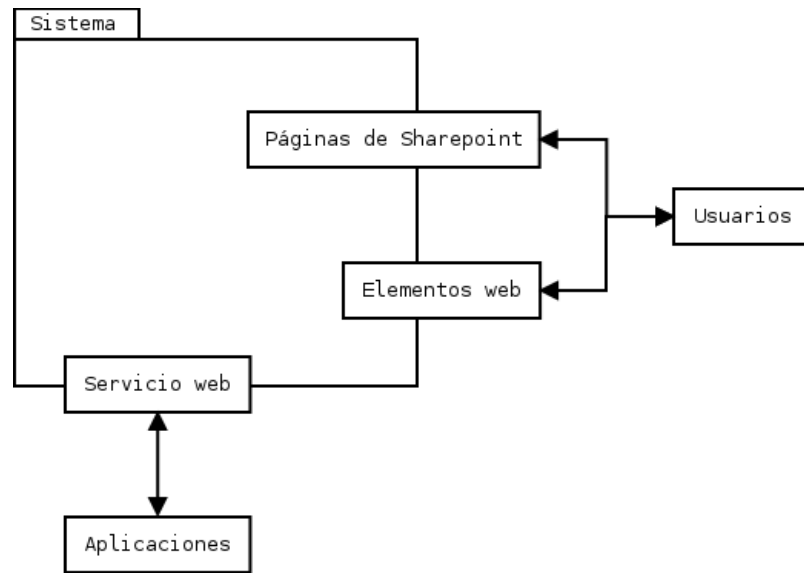


Ilustración 18: Diagrama de contexto



La arquitectura estratificada mostrada en la [Ilustración 17](#) y el diagrama de contexto en la **Error! Reference source not found.** son la manera más general de representar el sistema.

a. Interfaces externas del sistema. Para que la información entre o salga al sistema debe pasar por alguna de las siguientes interfaces en el nivel más alto:

- **Página de SharePoint:** La interfaz gráfica principal del sistema está construida sobre páginas de SharePoint escritas en ASPX y XSL. Los usuarios desde un navegador web, luego de identificarse pueden utilizar estas páginas para recuperar, ingresar, modificar o eliminar información en las vistas. Los usuarios administradores pueden alterar la apariencia o incluso el contenido de estas páginas.
- **Elementos web:** Son componentes con los que se construyen las páginas de SharePoint. Se programaron algunos elementos web para agregar al sistema funcionalidades no provistas por SharePoint.

- Servicio web: El servicio web provee acceso a las vistas de SharePoint a través de métodos para obtener información, ingresarla, modificarla y eliminarla. El servicio web fue creado para actuar como interfaz con otras aplicaciones.

b. Vistas de SharePoint. Una vista es una representación de los datos. En la mayoría de los casos los usuarios no esperan tener acceso a la totalidad de la información, y de tenerlo, podría parecerles abrumador. Las páginas de SharePoint hacen uso de las vistas para organizar la información, mostrando la cantidad esperada en el momento esperado. Las vistas modifican la información almacenada en las listas.

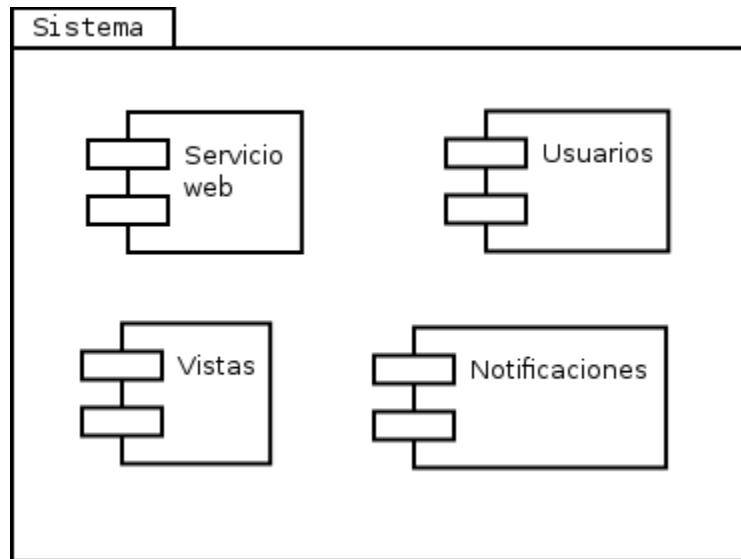
c. Listas de SharePoint. Es la tecnología de SharePoint para abstraer el concepto de “tabla” de una base de datos. Ocultan la información sobre llaves primarias y foráneas, así como de varios tipos de datos. Las listas deben ser representadas por vistas para poder ser accedidas.

d. Sitio de SharePoint. El sitio de SharePoint organiza las páginas, vistas y elementos web. Además pretende abstraer el concepto de “base de datos”. Desde este nivel de abstracción, todos los componentes de niveles más altos son considerados como una lista, inclusive las páginas y elementos web. Los usuarios administradores pueden cambiar parámetros generales de la apariencia del sitio.

e. Base de datos. Esta es la capa más baja de la jerarquía. Se comunica de manera directa únicamente con el sitio de SharePoint. En la base de datos está guardada toda la información necesaria para reconstruir el sitio, como las configuraciones, páginas, listas, vistas, documentos y la información de los usuarios.

2. Diseño de componentes. Un componente es un bloque de construcción de un software. Residen dentro de la arquitectura e interactúan entre sí.

Ilustración 19: Componentes del sistema



a. Usuarios. La tarea de este componente es la de agregar usuarios a SharePoint. Hace uso de las funcionalidades de Active Directory. Para poder agregar el usuario al sistema, es necesario primero crearlo en una *unidad organizacional* particular dentro del *dominio* de usuarios. Luego debe formar parte de un *grupo* de usuarios particular según los permisos que tendrá dentro del sistema.

b. Notificaciones. Este es un componente encargado de administrar las listas responsables de la comunicación entre los usuarios del sitio. Provee las funciones para enviar circulares y anuncios. También incluye una funcionalidad muy básica para llevar control de la ubicación de los usuarios.

c. Vistas. El componente de vistas está compuesto en realidad de varios componentes. La idea general es presentar al usuario la información de los objetos de datos más importantes y actuar como una interfaz que permita su mantenimiento. Los componentes que lo conforman se describen a continuación.

- Vistas principales. Estas vistas proveen funciones para buscar, detallar, listar, insertar, modificar y borrar la información de:
 - Cuentas
 - Contactos
 - Contratistas
 - Contratos
 - Incidentes
- Vistas secundarias. Estas vistas proveen funciones para detallar, listar, insertar, modificar y borrar la información de:
 - Asuntos
 - Direcciones de contratista
 - Direcciones de cuenta
 - Equipos
 - Equipos por actividad
 - Materiales
 - Materiales por actividad
 - Memos
 - Notas de actividad
 - Notas de contratista
 - Notas de cuenta
 - Notas de incidente
 - Servicios de contratos
 - Soluciones
 - Tipos de actividades
 - Tipos de contratista
 - Tipos de contratos
 - Tipos de cuentas
 - Tipos de equipos
 - Tipos de incidente
 - Tipos de materiales
- Actividades. Provee las funciones de las vistas a la lista de actividades. Adicionalmente, permite desplegar un calendario mensual, semanal o diario de las actividades en ejecución para una fecha específica.
- Contratistas. Este componente complementa la funcionalidad de la vista principal de contratistas para que los usuarios agregados a esta lista puedan acceder al sistema como usuarios técnicos.
- Reservas: Complementa la funcionalidad de la vista secundaria para materiales por actividad y la vista secundaria para equipos por actividad. Permite llevar control de las unidades disponibles de cada material o equipo tomando en cuenta las actividades que lo necesitan.

d. Servicio web. El componente de servicio web pretende actuar como una interfaz del sistema a la que se pueden conectar aplicaciones que extiendan la funcionalidad del mismo.

Un servicio web es un conjunto de protocolos y estándares para la comunicación entre sistemas, que permite la conexión entre estos a través de internet sin importar la plataforma sobre la cual están implementados.

Este módulo provee al sistema los siguientes servicios:

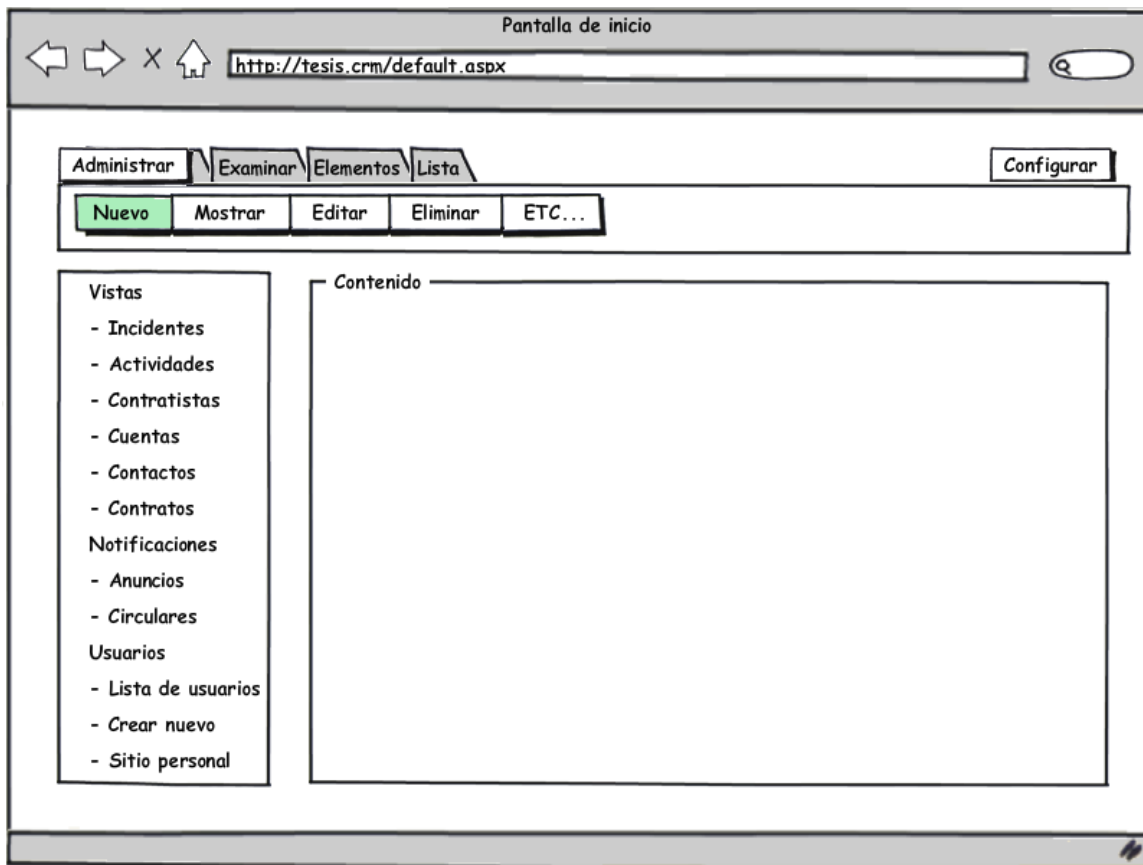
- Servicio de listas: Este servicio permite a una aplicación externa obtener los elementos de una o de todas las listas en el sistema. La aplicación que lo solicite debe proveer un nombre de usuario y contraseña, así como el nombre de la lista. Adicionalmente puede solicitar un bloque de resultados, por ejemplo, los elementos comenzando con el número 10 y hasta el número 20. Esta funcionalidad es muy útil cuando se programa una aplicación que solo muestra un grupo a la vez. También se pueden solicitar los elementos que cumplan con cierto criterio, por ejemplo, aquellos cuyo nombre comience con la letra “a” y termine con el número “1”.
- Servicio de usuarios: Al igual que con el servicio de listas, se debe proveer un nombre de usuario y contraseña para acceder este servicio. El módulo retorna una lista con información de todos los usuarios del sistema.
- Servicio de actualización de datos: Este servicio permite a clientes externos crear, modificar y eliminar elementos. El cliente debe indicar su usuario y contraseña, así como el número de ID del elemento que desea modificar o eliminar. El servicio retorna si la modificación se realizó con éxito.

Estos servicios retornan la información en un documento de XML que puede ser analizado fácilmente por cualquier tipo de aplicación.

3. Diseño de la interfaz de usuario. El diseño de la interfaz gráfica es muy importante en cualquier proyecto de software pues esta es una gran parte de lo que el usuario percibe del programa. Es importante que provea acceso a todas las funcionalidades de manera ordenada, que sea amigable y que la apariencia de cada uno de los componentes sea consistente.

Originalmente se trabajó sobre SharePoint 2007, y luego se migró el sistema a la plataforma SharePoint 2010. Las interfaces gráficas originales desarrolladas en SharePoint 2007 se pueden encontrar en el anexo.

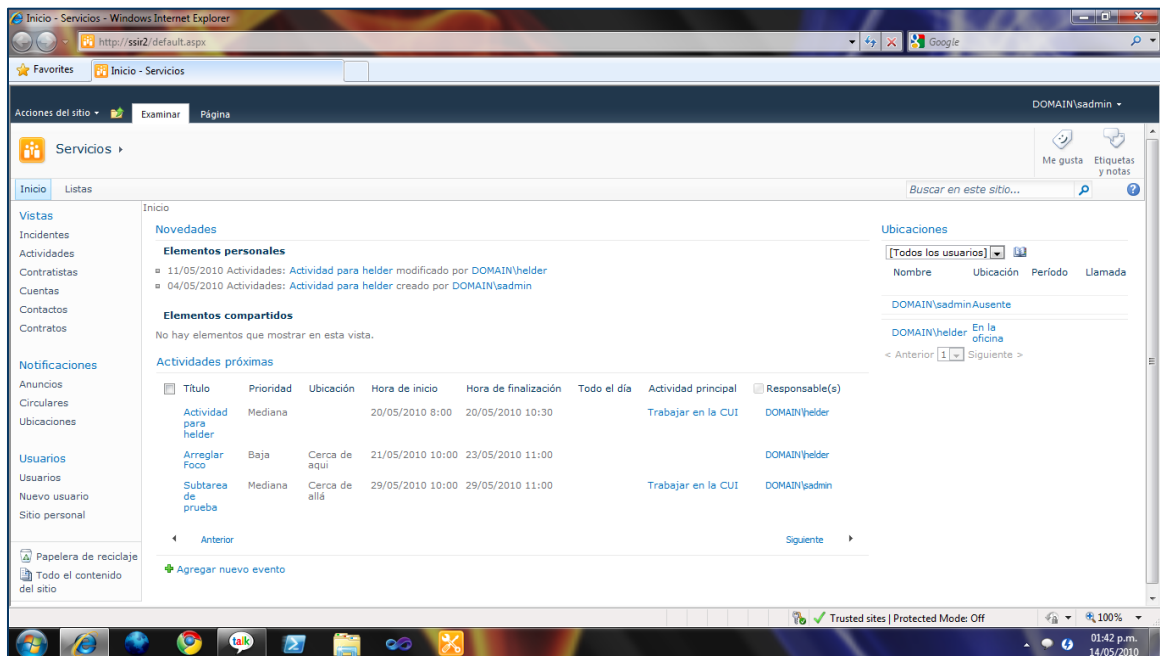
Ilustración 20: Maqueta de la apariencia



a. Esquema general. Todas las páginas del sitio se diseñaron para tener la apariencia que se muestra en la Ilustración 21. El sistema se desarrolló sobre la plataforma Microsoft SharePoint, la cual provee muchas herramientas para la construcción del sitio. Todas las páginas cuentan con un “listón” en la parte superior a través del cual se puede acceder a las funciones de la página, a la configuración del sitio y a las opciones de personalización. El listón cuenta además, en cualquiera de las páginas, con una pestaña para facilitar la navegación en el sitio llamada “examinar”.

En el lado izquierdo, las páginas cuentan con un menú de acceso rápido, a las funciones más utilizadas del sitio. El área denominada “contenido” varía en cada una de las páginas.

Ilustración 21: Pantalla de inicio

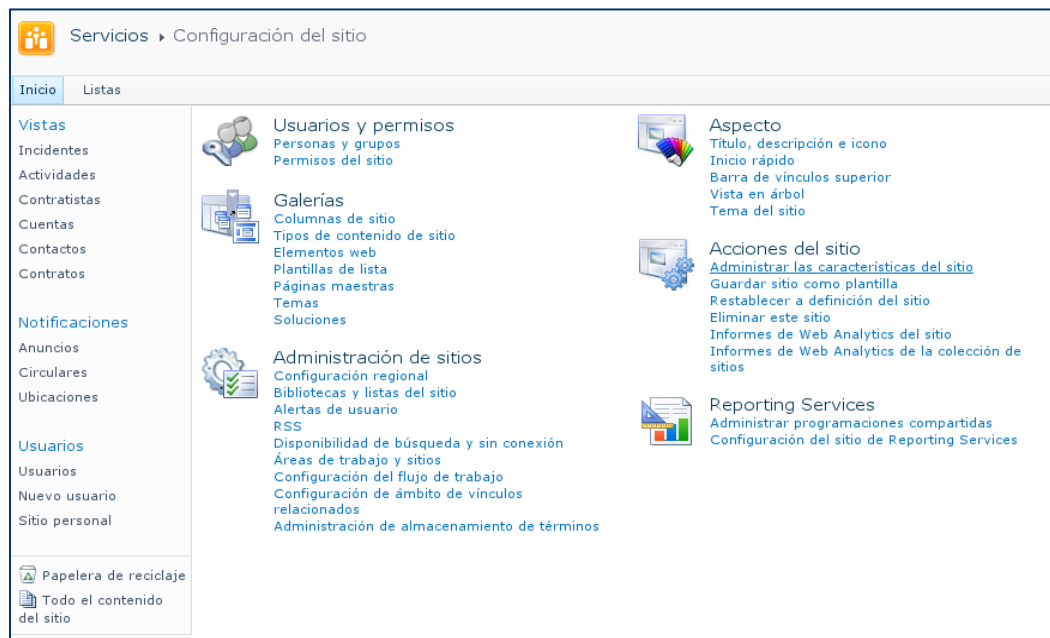


b. Pantalla de inicio. La pantalla de inicio pretende ser un punto de acceso rápido a las funciones importantes del sistema. Se decidió enfatizar las tareas asignadas al usuario que tengan una fecha próxima de cierre, que sean de prioridad alta y aún no estén terminadas, o que estén retrasadas.

La pantalla de inicio tiene las siguientes características:

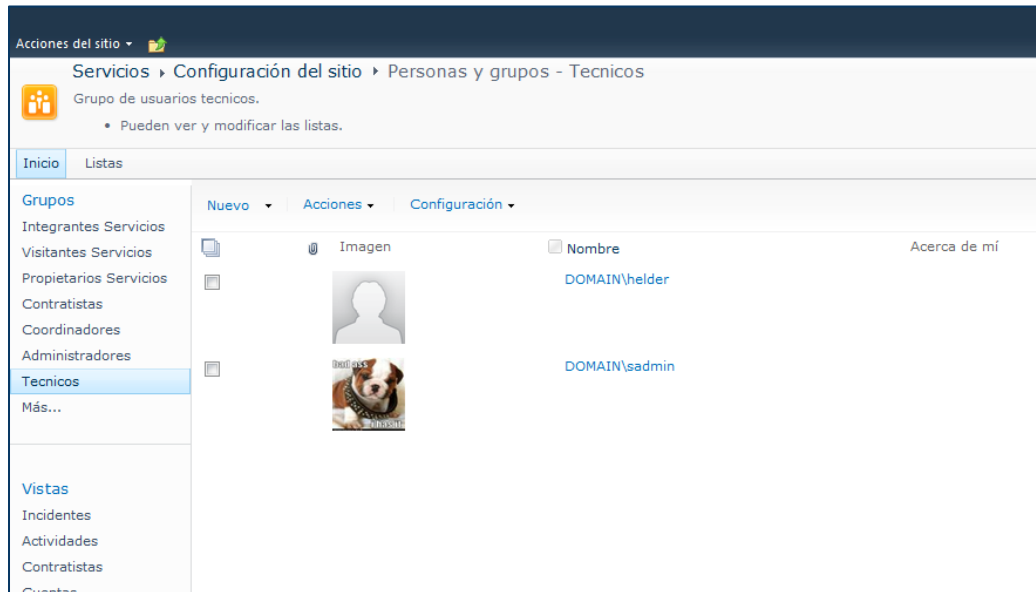
- Lista de mensajes recibidos: Muestra los mensajes internos del sistema que han sido enviados por otros usuarios.
- Listado de las actividades próximas. Muestra las actividades más importantes para el usuario y se consideró que es lo más importante para muchos usuarios que visiten el sitio.

Ilustración 22: Pantalla de configuración



c. Pantalla de configuración. La pantalla de configuración es una función provista por SharePoint a través de la cual se puede acceder a la configuración del sitio. Solo los usuarios de tipo administrador tienen acceso a esta pantalla.

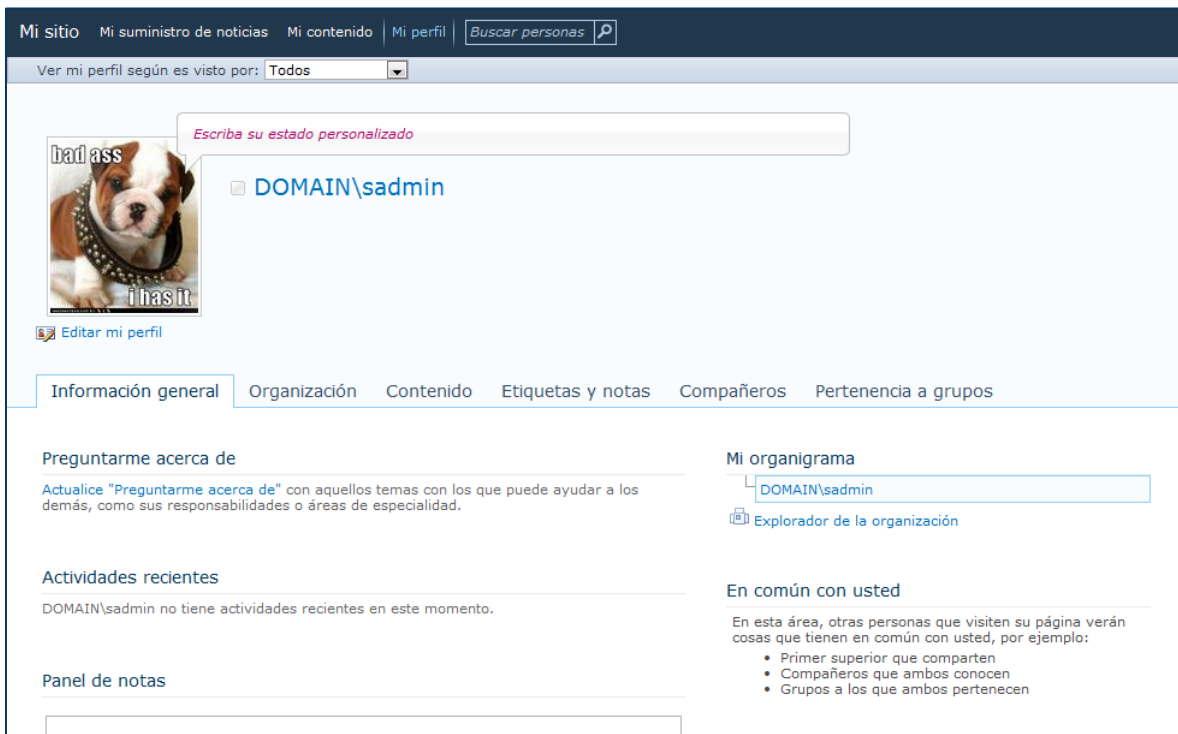
Ilustración 23: Pantalla para listar usuarios



d. Pantalla para listar usuarios. En la pantalla para listar usuarios es posible acceder a la lista de usuarios en el sistema. El menú del sitio cambia para mostrar los grupos de usuarios; al hacer clic en alguno de estos grupos, la lista será filtrada para mostrar únicamente a los integrantes de ese grupo.

De acuerdo a los permisos de cada usuario, es posible que no se muestren todos los usuarios o que no pueda acceder a cierta información de otro usuario.

Ilustración 24: Página personal



e. Página personal. Un usuario puede acceder a su página personal a través del botón de configuración, en la parte superior derecha de la página. Además, a través de la lista de usuarios, es posible acceder a la página personal de los demás usuarios del sitio. En su página personal, un usuario puede personalizar la información que desea que otros usuarios vean. Esta información incluye información de su puesto en la organización, de sus actividades recientes, de sus grupos de usuarios y algunas notas personales.

Ilustración 25: Nuevo usuario

The screenshot shows a web interface for creating a new user. At the top, there is a navigation bar with 'Acciones del sitio', 'Examinar', and 'Página'. Below this, the breadcrumb path is 'Servicios > Documentos compartidos > Predeterminada'. A sub-header reads 'Comparta un documento con el grupo agregándolo a esta biblioteca de documentos.' The main content area has a left sidebar with 'Inicio' and 'Listas' tabs, and a 'Vistas' section containing links for 'Incidentes', 'Actividades', 'Contratistas', 'Cuentas', 'Contactos', and 'Contratos'. Below 'Vistas' are 'Notificaciones' with links for 'Anuncios', 'Circulares', and 'Ubicaciones'. The main form area contains the following fields and controls:

Nombre de usuario	<input type="text"/>
Grupo	<input type="text" value="Tecnicos"/>
Contraseña	<input type="password"/>
	<input type="button" value="Generar"/> Click para generar una contraseña.
Confirmar contraseña	<input type="password"/>

f. Nuevo usuario. Los usuarios administradores y coordinadores pueden crear otros usuarios desde la página **Nuevo usuario**. Deben ingresar un nombre único para el usuario y especificar el grupo al que pertenecen. También deben ingresar la contraseña con la que este usuario se identificará. Esta página incluye la función para generar contraseñas seguras automáticamente.

Ilustración 26: Pantalla para buscar y listar elementos

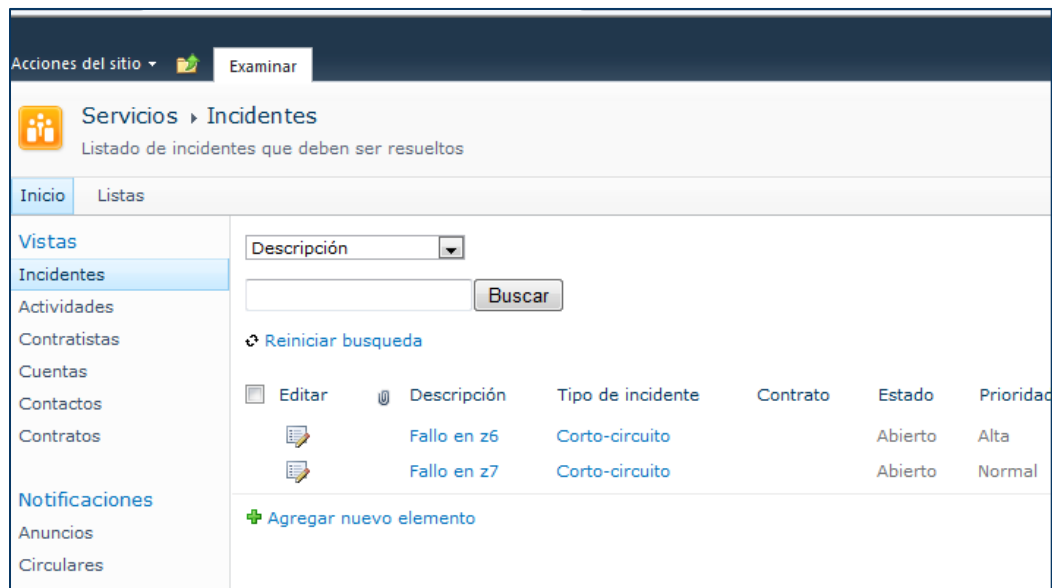
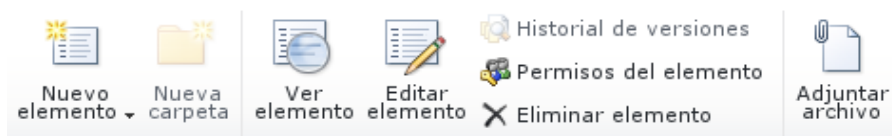


Ilustración 27: Funciones de una lista



g. Pantalla para buscar y listar elementos. Cada una de las vistas principales del componente de vistas cuenta con una página para listar elementos. En el centro de la pantalla se presenta una lista de todos los elementos relacionados con el usuario. Al hacer clic en el nombre del elemento, se le mostrará al usuario una ventana emergente con el detalle de ese elemento. De la misma manera, al hacer clic en Editar o Agregar nuevo elemento se le presentará una ventana emergente en donde puede editar o insertar un elemento respectivamente. En la lista de elementos es posible seleccionar una o varias filas, y en el listón se presentarán las acciones que se pueden llevar a cabo con esos elementos, estas pueden observarse en la Ilustración arriba.

La lista de elementos puede en ocasiones ser muy grande. Por lo tanto, los elementos se muestran en grupos de 10, y se creó una función de búsqueda. La función de búsqueda filtra la lista para mostrar únicamente los elementos que coincidan con el criterio de búsqueda.

Ilustración 28: Calendario de actividades mensual

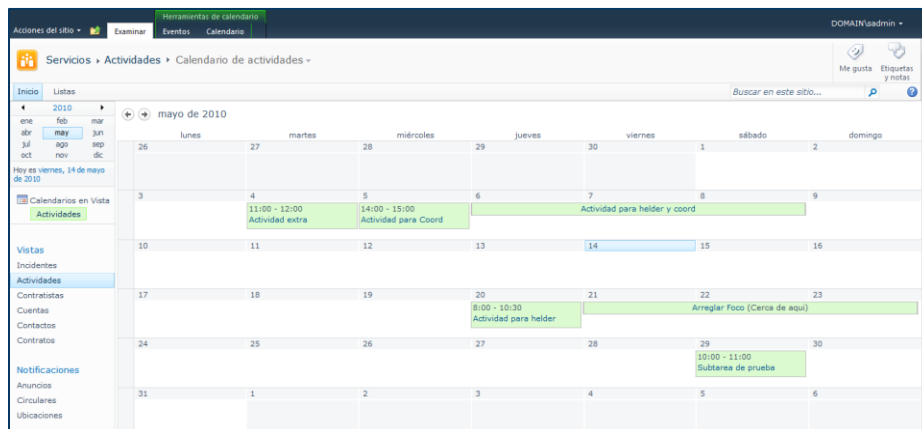


Ilustración 29: Calendario de actividades semanal

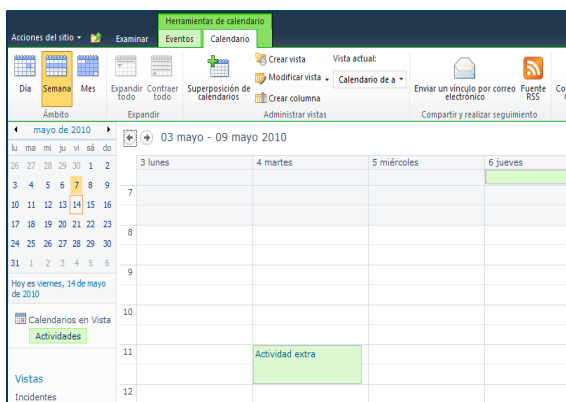
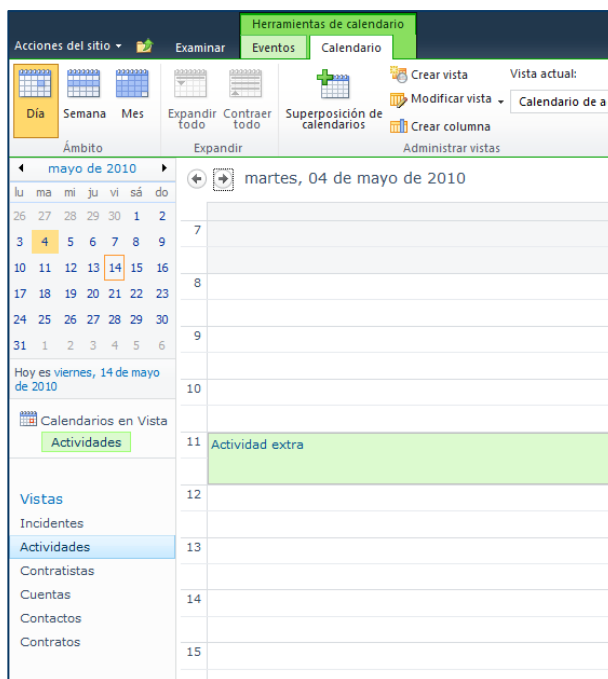


Ilustración 30: Calendario de actividades diario



h. Calendarios de actividades. Las actividades pueden ser mostradas de manera mensual, semanal y diaria. Para cambiar de calendario se puede hacer clic en el calendario adecuado en el listón. En el menú se agrega un pequeño calendario mensual para facilitar la navegación. Haciendo clic en una actividad, el usuario puede ver el detalle de ésta.

Ilustración 31: Detalle de elemento

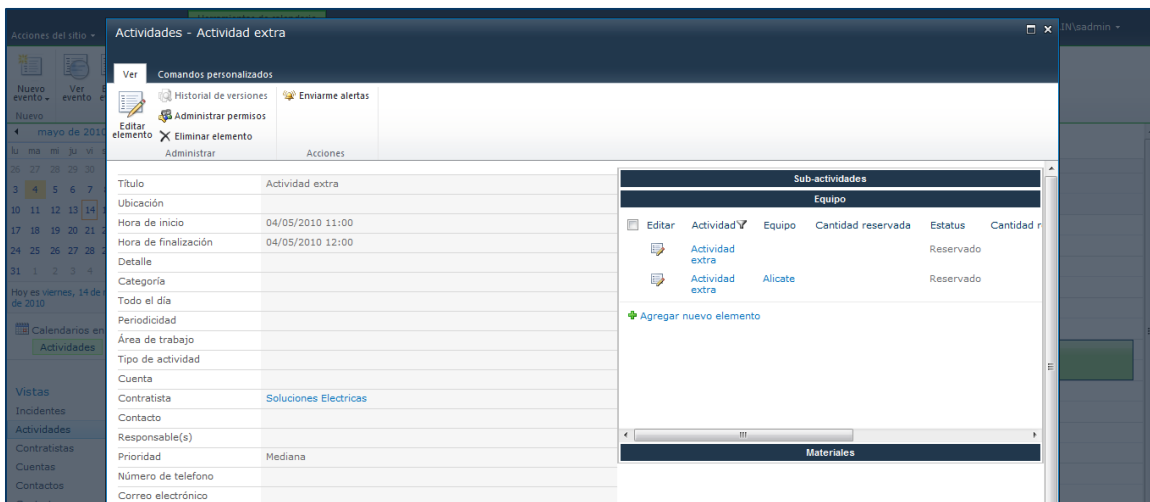
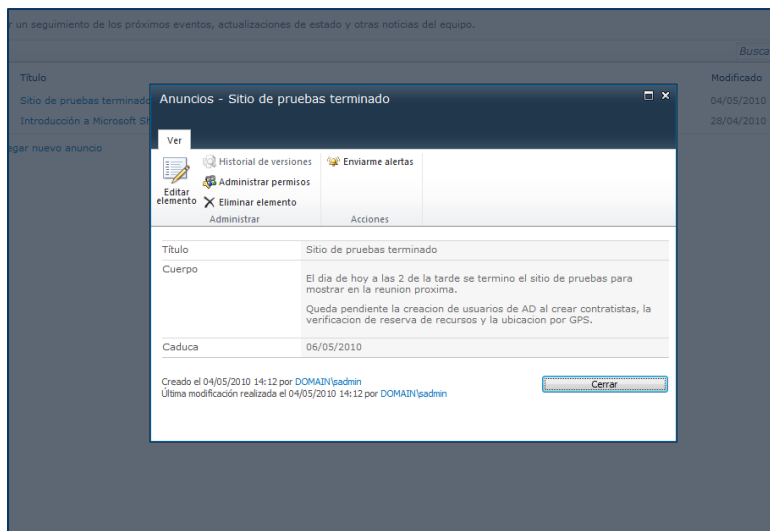
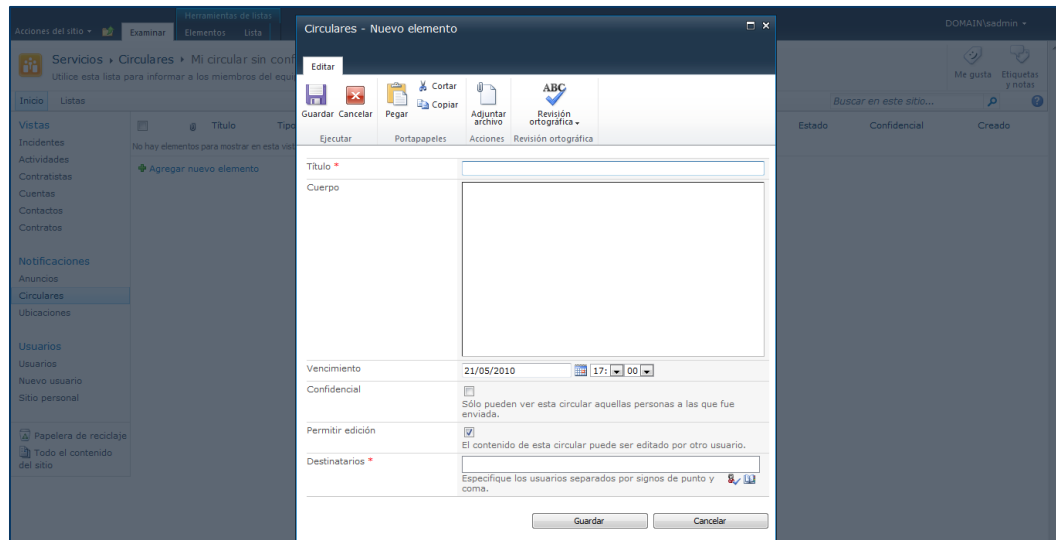


Ilustración 32: Detalle simple



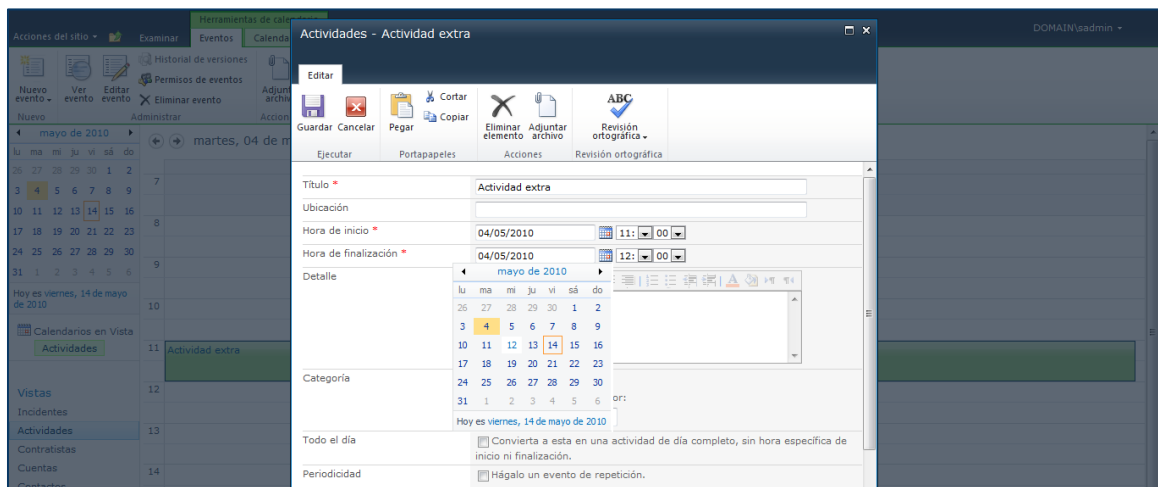
i. Detalle de elemento. Al hacer clic a algún elemento de cualquiera de las listas, una ventana emergente se muestra con el detalle de éste. El listón de esta ventana contiene los comandos para editar o eliminar el elemento. En el lado derecho de la ventana se agrega un control del tipo “acordeón” con el cual se pueden ver las listas relacionadas con el elemento.

Ilustración 33: Nuevo elemento



j. Nuevo elemento. Al seleccionar la opción de agregar un nuevo elemento, al usuario se le presenta una ventana emergente con los campos que necesita llenar. El listón de esta ventana contiene las acciones que se pueden realizar con el elemento nuevo, como adjuntar un archivo y revisar la ortografía de los campos.

Ilustración 34: Editar elemento



k. Editar elemento. El usuario puede seleccionar editar un elemento de una lista; al hacerlo, se le presenta una pantalla emergente similar a la de **Nuevo elemento**, pero con los datos del elemento previos a la edición. Tanto la pantalla de **Nuevo elemento**

como la de **Editar elemento** cuentan con un pequeño calendario para los datos de tipo fecha.

4. Diseño de seguridad de la información. Los principios básicos de la seguridad de la información son la confidencialidad de los datos, su integridad y su disponibilidad.

a. Confidencialidad. Se refiere a prevenir la divulgación de información a personas o sistemas no autorizados. El sistema requiere que los usuarios se identifiquen antes de poder ingresar, a través de un dominio de usuarios, un nombre y una contraseña. Se utilizó la herramienta **Active Directory** para cumplir este propósito. Antes de poder utilizar el sitio, un usuario debe ser agregado al sistema por un usuario ya existente (y que tenga esta habilidad). Para los usuarios que traten de conectarse al sistema a través del servicio web, este nombre y usuario deben ser enviados como parámetro, codificados como se describe en el Anexo C. Toda la información retornada por el servicio web está cifrada utilizando el algoritmo Triple DES, la clave para revertir el proceso consta de dos partes, una que únicamente el cliente conoce y otra que es enviada oculta en el mensaje. El cliente combina ambas claves y de esta manera puede acceder a la información. Se seleccionó el algoritmo Triple DES para maximizar la compatibilidad del servicio web con aplicaciones de la framework compacta de .NET, pues este es el único algoritmo provisto por este framework para esta tarea. (Krutz & Russell, 2003)

b. Integridad. Se refiere a mantener los datos libres de modificaciones no autorizadas. La integridad se asegura en el sistema a través del modelo de vistas y el modelo de grupos provisto por SharePoint. Únicamente los usuarios que crean la información o que son responsables de ella son capaces de modificarla. Asimismo, los usuarios pertenecientes a ciertos grupos tienen diferente nivel de acceso a la información. En cuanto al servicio web, este trabaja de manera transaccional, y el usuario debe identificarse cada vez que requiera acceso a los datos; juzgar si tiene

acceso se le delega a SharePoint. Si el mensaje que se retorna fue modificado en el camino hacia el cliente, será imposible decodificarlo. (Krutz & Russell, 2003)

c. Disponibilidad. Se refiere a la condición de que la información debe encontrarse a disposición de quienes deben acceder a ella. Esta característica de seguridad depende principalmente de la máquina servidor en el que se implemente la solución, y del acceso a internet de las máquinas clientes. Por su parte, el sistema debe ser estable, para estar disponible el tiempo que el servidor lo esté. (Krutz & Russell, 2003)

5. Diseño de pruebas. Las pruebas tanto del servidor como del servicio web se realizaron siguiendo una metodología de “caja blanca”. En esta metodología, el programador tiene acceso a las estructuras de datos internas y a las interfaces externas del sistema. Siguiendo dicha metodología se realizaron tres tipos de pruebas: (Krutz & Russell, 2003)

a. Pruebas de la cobertura del código. En estas pruebas se introdujeron al servidor datos de ejemplo considerando la mayor cantidad de casos posibles con el objetivo de evaluar la capacidad del sistema para administrar esa información. La información insertada fue modificada y eliminada extensivamente, tanto desde el servidor como desde el servicio web.

b. Pruebas de mutación e inyección de faltas. Estas pruebas consisten en deliberadamente modificar el código, insertando errores lógicos para probar cómo reaccionan los sistemas de seguridad. Se realizaron principalmente para comprobar el comportamiento del servidor en el caso de que el servicio web estuviera enviando datos inconsistentes.

c. Pruebas estáticas. Durante las pruebas estáticas, el código no es ejecutado, sino que es revisado por el programador para depurarlo, optimizarlo y documentarlo. Durante el desarrollo del proyecto se realizaron pruebas estáticas en muchos momentos; en particular durante la migración de SharePoint 2007 a 2010 pues era muy importante que el código fuera muy general para que continuara siendo útil luego del cambio de plataforma.

d. Pruebas de seguridad. Las pruebas de seguridad que se realizaron consistieron en tratar de obtener información del sistema usando programas comúnmente utilizados por atacantes. Este tipo de pruebas en realidad no encaja en la metodología de “caja blanca” sino en la metodología de “caja negra”.

Se atacó principalmente la comunicación entre los clientes y el servicio web utilizando analizadores de paquetes de red. Los analizadores que se utilizaron fueron **ethereal** y **Cain**, los cuales fueron seleccionados luego de investigar las herramientas más comúnmente utilizadas para este tipo de ataques. Originalmente la implementación utilizaba HTTPS sin el respaldo de una autoridad de certificados, pero el software Cain fue capaz de obtener la información enviada fácilmente utilizando una técnica conocida como **ARP Poisoning**, con la que se identifican los nombres y direcciones de los nodos de la red. El procedimiento seguido para romper la seguridad fue:

- Abrir el canal seguro entre el cliente y el servidor.
- Desde una tercera máquina obtener tanto la dirección del cliente como la del servidor.
- Hacer creer al servidor que ésta máquina es el cliente, y al cliente que es el servidor. Esto es posible sólo por que ambos están dispuestos a aceptar cualquier certificado, ya que no se cuenta con uno oficial.
- Recibir los paquetes tanto del cliente como del servidor, guardar una copia, y enviarla a su destinatario.

Luego de identificar esta vulnerabilidad en el diseño, se decidió codificar el mensaje, y de esta manera se aseguró que aunque una tercera máquina obtenga los datos, no pueda comprender su contenido.

6. Diseño de implementación. Como se ha mencionado en el alcance, la implementación del sistema está fuera del alcance del proyecto, pero se recomienda utilizar una técnica de adopción paralela para instalar el sistema desarrollado en una empresa que ya cuenta con algún tipo de sistema para el manejo de las relaciones con los clientes. Si la empresa no cuenta con un sistema de este tipo previamente, se recomienda una adopción inmediata. Por supuesto, cada empresa debe decidir cómo implementar el software según sus condiciones, pero no se recomienda una adopción gradual, pues los componentes dependen mucho los unos de los otros como para ser utilizados de manera individual. (Neal, 2010)

a. Adopción paralela. Durante la implementación paralela se trata de disminuir el riesgo al instalar y correr el nuevo sistema sin desactivar el viejo. Ambos se utilizarán juntos durante un periodo de tiempo prudente, determinado por la organización. Al final de este periodo, si se considera que el nuevo sistema cumple los requerimientos para que la empresa pueda funcionar sin el viejo, entonces se deshabilita el viejo. Es responsabilidad de la empresa la planeación y control de esta implementación. Se diseñó el sistema para hacer la adopción lo más sencilla posible, utilizando nomenclatura y posicionamiento de componentes de la misma manera que lo hacen otros software de CRM, específicamente Microsoft Dynamics, SAP y Oracle Siebel. (Neal, 2010)

7. Discusión. Originalmente, se diseñaron e implementaron los componentes del sistema en SharePoint 2007, y a mediados del proyecto, se decidió cambiar la plataforma a SharePoint 2010. Cambiar la plataforma a un ambiente que aún no estaba disponible al público general presentaba una buena oportunidad para obtener una ventaja competitiva y una manera de posicionar el sistema un paso delante de la competencia. Además, la apariencia y la estabilidad del sistema prometían mejorar.

El diseño de cada uno de los componentes se llevó a cabo previo a su implementación. Gracias a ello, el diseño general resultó muy modular y adaptable. La migración entre las plataformas no fue muy complicada, a pesar que todos los componentes tuvieron que ser creados de nuevo. Durante la migración se mejoró aún más la apariencia gráfica, pues se cambiaron algunos elementos de SharePoint por código de JavaScript, que es mucho más flexible.

Los componentes en los que se dividió el sistema fueron adecuadamente seleccionados. Gracias a su tamaño y complejidad fue posible implementar cada uno y crear un sistema completo. Sin embargo, originalmente no se identificó el componente encargado del manejo de contratistas, pero gracias a su similitud con el componente de cuentas, integrarlo al sistema no fue una tarea complicada.

SharePoint provee una interfaz gráfica para el sitio que tiene una apariencia aceptable, pero que es muy difícil de cambiar. Inclusive el administrador sólo tiene opciones como el color o el tipo de menú, y éstas se redujeron aún más con SharePoint 2010. Para poder crear la interfaz de este sistema se trabajó directamente sobre el código XSL de las páginas utilizando SharePoint Designer. Ésta resultó ser una herramienta inestable pues no mantenía consistencia entre el código que se editaba y el que se utilizaba para desplegar las páginas; además, en varias ocasiones solía cerrarse inesperadamente, lo que la hacía difícil de utilizar.

V. Discusión

La metodología de desarrollo permitió seguir un procedimiento ordenado y eficiente, que fue capaz de cumplir los objetivos a pesar de los imprevistos. La sincronización con el proyecto paralelo de la interfaz móvil se llevó a cabo de manera casi transparente, gracias a la flexibilidad de la tecnología de servicios web, que es un estándar para la comunicación entre sistemas que no depende de la plataforma de estos. Los lenguajes de programación y representación seleccionados probaron ser herramientas prácticas y de fácil aprendizaje. Finalmente, se considera que la comunicación con el gerente de ventas fue constante y permitió que el sistema avanzara en la dirección correcta.

La plataforma SharePoint para el desarrollo fue una gran fuente de herramientas en varios niveles de abstracción, con una gran variedad de funcionalidad y capacidades de comunicación con muchas otras herramientas de Microsoft. Sin embargo, se comportó de una manera inestable, principalmente en su versión 2007. Algunos de los problemas encontrados con SharePoint 2007 fueron problemas de acceso al sitio, problemas con el reconocimiento de los elementos web y problemas con la administración de las páginas de contenido. SharePoint Designer es el programa utilizado para editar las páginas del sitio, y aunque su funcionalidad es completa, presentaba problemas como inestabilidad e inconsistencia del contenido en el editor y el contenido real en la página.

Fue posible crear un sistema con el potencial de mejorar el servicio al cliente que una empresa de servicios es capaz de prestar. Sin embargo, es responsabilidad de la empresa que adquiera el software hacer un uso adecuado de sus herramientas, reportando todos los incidentes y las actividades de servicio; manteniendo al día el estado de las actividades y los materiales y herramientas e ingresando la información de cada cliente nuevo.

VI. Conclusiones

- El sistema diseñado cumple con las funcionalidades necesarias para cumplir los objetivos propuestos.
- Se definió una estructura simple, pero funcional. Se consideraron correctamente los componentes necesarios para la construcción del módulo de servicios, y se logró que la comunicación entre cada uno de estos fuera fluida y ordenada. Esto permite un sistema con una apariencia consistente y un tiempo de respuesta relativamente corto.
- Se seleccionaron adecuadamente los paquetes de software que aseguran escalabilidad, portabilidad y funcionalidad del sistema. Estos paquetes se integraron entre sí principalmente gracias a sus capacidades nativas, lo cual garantiza un sistema estable. Microsoft provee plataformas adicionales que facilitan la integración entre sus paquetes.
- La selección de servicios web como la interconexión entre los proyectos facilitó el desarrollo. De esta manera, el tiempo limitado de desarrollo fue aprovechado más eficientemente. Esto fue un factor clave que permitió un enfoque dirigido a funcionalidad entre proyectos y no a comunicación.
- Implementar todos los componentes del sistema sobre una plataforma común permitió la creación de un sistema estable y con una apariencia integrada. Esto no sólo facilita el desarrollo, sino que mejora la experiencia de los usuarios de la aplicación.

VII. Recomendaciones

- Expandir este sistema para agregarle funcionalidad y hacerlo una solución aún más atractiva. El servicio web puede ser usado por varios tipos de interfaces móviles, aplicaciones de escritorio, o para procesos automatizados.
- Si se implementa un sistema similar, se recomienda utilizar una plataforma base distinta a SharePoint. El enfoque que Microsoft está dando a esta línea de productos está enfocado en la creación de comunidades virtuales, en donde los usuarios interactúan entre sí y comparten contenido, por lo que hacer un sistema en donde hay que restringir el acceso a algunas funciones para algunos usuarios es muy complicado.
- Agregar al sistema la funcionalidad de graficar en un mapa la posición física de los técnicos. Es posible almacenar en una lista la latitud y longitud provista desde un dispositivo GPS y crear una vista para esta lista en donde se muestre la ubicación.
- Crear los módulos de ventas y mercadeo para así contar con un CRM completo. Sin embargo, se recomienda también mantener una versión reducida del CRM con el módulo de servicios a un precio menor.
- Agregar la capacidad para el manejo de flujos de trabajo de la empresa. SharePoint proporciona herramientas para la creación de flujos de trabajo, se recomienda diseñar una estructura para la administración de grupos de trabajo basada en listas y habilitar estas herramientas.

VIII. Bibliografía

- aiveo. (Febrero de 2009). *CRM Comparison*. Recuperado el Julio de 2010, de <http://www.slideshare.net/gallouche1/crm-comparison-2009>
- Baetjer, J. (1998). *Software as Capital*. IEEE Computer society press.
- Bucholtz, C. (5 de Abril de 2010). *ERP vs. CRM: the differences - and how they're changing*. Recuperado el Octubre de 2010, de Forecasting Clouds: <http://www.forecastingclouds.com/articles/6538/erp-vs-crm-the-differences-and-how-theyre-changing/>
- D. Ross, K. S. (1977). *A language for communicating ideas*. IEE.
- Gartner. (2009). *What's 'Hot' in CRM Applications in 2009*. Stamford, Connecticut, EEUU: Ed. Thompson.
- Goldensoft. (Junio de 2010). *¿Qué es un CRM?* Recuperado el Noviembre de 2010, de Managers Magazine: <http://managersmagazine.com/index.php/2010/06/%C2%BFque-es-un-crm/>
- IEEE. (2004). *SWEBOK*. EEUU: Computer Science Press.
- Kendall, K. &. (2005). *Análisis y diseño de sistemas* (Sexta ed.). México: Pearson.
- Knorr, E., & Gruman, G. (s.f.). *What cloud computing really means*. Obtenido de InfoWorld: <http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031>
- Krutz, R., & Russell, D. (2003). *The CISSP prep guide*. Indianapolis, Indiana, EEUU: Wiley.
- LapouChnian, A. (2005). *Goal Oriented Requierements Engineering*. Junio: University of Toronto.
- Neal, H. (11 de Marzo de 2010). *ERP Implementation Strategies – A Guide to ERP Implementation Methodology*. Recuperado el Octubre de 2010, de Software Advice: <http://www.softwareadvice.com/articles/manufacturing/erp-implementation-strategies-1031101/>

- Netcraft. (27 de Marzo de 2010). *March 2010 Web Server Survey*. Recuperado el 19 de Septiembre de 2010, de Netcraft:
http://news.netcraft.com/archives/2010/03/17/march_2010_web_server_survey.html
- Pressman, R. (2005). *Ingeniería de software* (Sexta ed.). México: MCGrawHill.
- TEC. (2010). *Relationship Management*. Recuperado el Octubre de 2010, de Technology evaluation centers: <http://relationship-management.technologyevaluation.com/>
- Turbit, N. (27 de Junio de 2005). *Defining the Scope of a Project*. Recuperado el Abril de 2010, de The PROJECT PERFECT White Paper Collection:
http://www.projectperfect.com.au/downloads/Info/info_define_the_scope.pdf
- wathiserp. (2010). *ERP Definition*. Recuperado el Octubre de 2010, de What is ERP?:
<http://wathiserp.net/>

IX. Anexos

A. Revisión de términos

.NET: es un **framework** desarrollado por Microsoft para permitir a las aplicaciones abstraer el hardware en el que corren. También se ha enfocado al desarrollo rápido de aplicaciones y en la integración de varios lenguajes de programación.

Abstracción: se refiere, en el contexto de informática a ocultar detalles y hacer énfasis en conceptos más generales.

Applet: es una pequeña aplicación dentro de un programa mayor que cumple una tarea específica. En el contexto de la plataforma Java, un applet agrega funcionalidad a las páginas de internet que HTML no puede proveer.

Arquitectura de software: es la estructura general del software, la organización de los componentes del programa, llamados módulos. (Pressman, 2005)

Base de datos: Es un conjunto de datos pertenecientes a un mismo contexto y almacenados digital y sistemáticamente para su posterior uso.

Ciente: cualquier persona que tiene un interés en el éxito del resultado del proyecto: gerentes de negocios, usuarios finales, gente de apoyo, etcétera. (Pressman, 2005)

Componente de software: se refiere a cualquier recurso desarrollado con un fin específico, y que puede formar, en conjunción con otros componentes, un sistema funcional.

Computación en nube (*Cloud computing*): La computación en nube consiste es un modelo en donde la computación está basada sobre el Internet. Los recursos de software e información llegan a los dispositivos que los solicitan desde un servidor remoto y no están instalados directamente en estos.

DBMS: siglas en inglés para sistema manejador de bases de datos (*DataBase Management Sistem*). Es un sistema que se encarga de gestionar la inserción, modificación y eliminación de información en una base de datos.

Dominio (de usuarios): un dominio en el contexto de Microsoft Windows y Active Directory, es un grupo de computadoras que comparten una base de datos de usuarios.

ERP: siglas en inglés para planificación de recursos empresariales (*Enterprise Resource Planning*). Es un sistema integrado basado en sistemas de cómputo para la gestión de recursos internos y externos de una empresa. Su propósito es facilitar el flujo de información entre las funciones de la empresa. Un CRM puede ser considerado como un componente de un ERP, el componente que involucra a los clientes. El CRM puede alimentar varios componentes de un ERP, apoyando en el área de entrega de productos y los datos de servicio.

Flujos de trabajo: se refiere al estudio y gestión de las operaciones de una actividad de trabajo, sus tareas, cómo se lleva a cabo, cómo se sincroniza con otras tareas y actividades, cómo fluye la información, y cómo se le da seguimiento al cumplimiento de las tareas.

Framework: un framework de software es una colección de librerías que provee una funcionalidad general, y que pueden ser expandidas por sus usuarios para proveer funcionalidad específica.

Grupo (de usuarios): es un conjunto de usuarios que comparten algunas características, como permisos de acceso.

HTML: siglas en inglés para lenguaje de marcado de hipertexto (*HyperText Markup Language*). Es un lenguaje diseñado para describir la estructura y el contenido de las “páginas” que forman Internet.

HTTP y HTTPS: siglas en inglés para protocolo de transferencia de hipertexto (*HyperText Transfer Protocol*). Describen el protocolo (formato) y la tecnología para enviar mensajes a través de la web. HTTPS es la versión segura de HTTP que trata de garantizar la confidencialidad de los mensajes sobre una red insegura hasta que lleguen a su destinatario.

IDE: siglas en inglés para ambiente de desarrollo integrado (*Integrated Development Environment*). Un IDE es una herramienta que los programadores utilizan para realizar software, provee funcionalidades como coloreado de sintaxis, acceso al compilador y en ocasiones funcionalidades para la depuración del software.

IEEE: Siglas para *Institute of Electrical and Electronics Engineers*, es una organización global de profesionales en el área de ingeniería sin ánimo de lucro. Cuenta con más de 395,000 miembros en 150 países. Ha sido responsable de estandarizar tecnologías y ha ganado el respeto de muchos ingenieros en todo el planeta.

Interfaz gráfica: es un programa que utiliza elementos gráficos para representar la información en una aplicación. Su principal uso es proporcionar a los usuarios un entorno visual sencillo con el que puedan comunicarse con el sistema.

Java: es una plataforma desarrollada por Sun Microsystems. La plataforma incluye una máquina virtual que puede ejecutar programas sin que dependan del sistema operativo,

un compilador para el lenguaje Java que crea código objeto para esta máquina y un conjunto de librerías que los desarrolladores pueden utilizar.

JavaScript: es un lenguaje de programación basado que puede ser ejecutado por navegadores web para hacer páginas más dinámicas.

Modularidad del software: es el concepto de dividir los sistemas de manera que sus componentes provean una funcionalidad específica.

Módulo: es una parte de un software de computación que realiza una o varias de las tareas del sistema.

Sistema operativo: es un software que actúa como interfaz entre el hardware y las aplicaciones de usuario.

Tabla: es un conjunto de registros dentro de una base de datos que tienen características (campos) similares.

UML: siglas en inglés para Lenguaje Unificado de Modelado (*Unified Modeling Language*) es un lenguaje estándar utilizado para visualizar, modelar y construir un sistema de software.

Unidad organizacional: es una clasificación de usuarios dentro de un dominio de Active Directory. Permite la existencia de usuarios con el mismo nombre en el dominio.

Widget: es una pequeña aplicación ejecutada por un motor para la ejecución de widgets. Su propósito es generalmente dar fácil acceso a funciones frecuentemente usadas y proveer de información visual, pero es posible utilizarlos de muchas otras maneras.

B. Revisión de literatura

Pressman, R. Ingeniería de software. Este libro detalla todos los procedimientos y características de la ingeniería de software. Fue muy utilizado en la creación del capítulo de teoría, pues su manera de presentar los conceptos es clara, ordenada y está bien ejemplificada. También fue utilizado para la definición del procedimiento a seguir para el análisis y diseño.

Kendall & Kendall. Análisis y diseño de sistemas. Este libro se enfoca en explicar el rol del analista de sistemas. Se utilizó como guía durante el análisis y diseño, y como complemento al libro de Pressman durante la elaboración de la teoría relacionada a esto.

LapouChnian. Goal Oriented Requirements Engineering. Es un libro que se dedica a explicar las metodologías de requerimientos orientadas a metas. Fue una guía para aprender estas metodologías, y una valiosa referencia para apoyar en la selección de una.

Knorr, E., & Gruman, G. What cloud computing really means. Este es un artículo publicado en la revista electrónica InfoWorld. Se considera una referencia valiosa pues fue difícil encontrar información reciente sobre el tema de computación en nube, y éste es muy completo para explicar los temas importantes relacionados.

SWEBOK. Este libro es el resultado del esfuerzo de la IEEE por unificar la información al respecto de la ingeniería de software. El libro presenta definiciones puntuales y detalladas de los temas relacionados con la ingeniería de software, y además son respaldadas por una organización importante como la IEEE.

C. Algoritmo para la codificación de la información de usuario

El algoritmo para la codificación fue desarrollado por Helder Gramajo, las especificaciones del algoritmo son las siguientes:

1. Componentes. Para funcionamiento de este algoritmo se necesitan los siguientes elementos.

- La lista de caracteres permitidos.
- La tabla de codificación.

2. Formato de la tabla de codificación. La tabla consiste en filas y columnas de letras, las cuales no se repiten ni en filas ni en columnas. De esto viene el apodo de este algoritmo “sudoku” pues es similar al juego con el mismo nombre, que se basa en una tabla numérica.

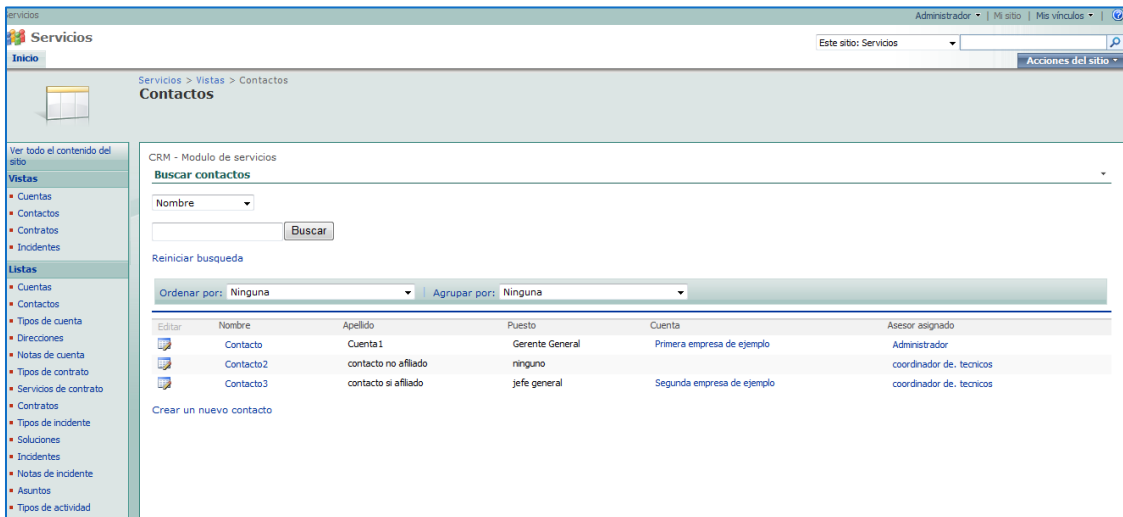
3. Algoritmo. El pseudocódigo del algoritmo es como se describe a continuación:

- Tomar el primer carácter para codificar
- Escoger un carácter aleatorio entre los caracteres validos
- En la tabla, se localiza la columna perteneciente al carácter para codificar, y la fila perteneciente al carácter aleatorio.
- La celda encontrada es el carácter codificado
- Se toma el siguiente carácter, y se repiten los pasos del 2 al 5 hasta que ya no queden caracteres para cifrar.

El proceso para obtener el texto a partir del texto cifrado es simplemente tomar el carácter codificado y el carácter correspondiente en la llave, y localizar la columna en donde se encuentran. El carácter correspondiente a esa columna es el texto original.

D. Interfaz gráfica en SharePoint 2007

Pantalla para listar y buscar elementos:



Pantalla de detalle de elemento simple:



Pantalla de detalle de elemento complejo:

CRM - Modulo de servicios
< Buscar otra cuenta

Detalle de la cuenta Crear una nueva cuenta

Nombre de la cuenta:	Primera empresa de ejemplo	Contacto principal:	Contacto	Nombre dirección:	Direccion central
Tipo de cuenta:	Cuenta de ejemplo	Cuenta principal:		Calle 1:	6ta av.
Código de cuenta:	007	Número de central telefónica:	24764951	Calle 2:	14-33
Nombre fiscal:	EMPRESA 1	Otro número telefónico:	24378051	Calle 3:	zona 9
Tipo de cuenta:	Cuenta de ejemplo	Número de fax:	22554488	Ciudad:	Guatemala
Asesor asignado:	Administrador	Sitio web:	http://www.google.com	Estado/provincia:	Guatemala
Identificador Fiscal 1:	44721241	Correo electrónico:		Código postal:	7
Identificador Fiscal 2:				País/Región:	Guatemala

editar

Contactos | Direcciones | Contratos | Incidentes | Actividades | Notas

Editar	Nombre	Apellido	Puesto	Asesor asignado	Cuenta
	Contacto	Cuenta1	Gerente General	Administrador	Primera empresa de ejemplo

Nuevo contacto

Pantalla de edicion de elementos:

Administrador | Mi sitio | Mis vínculos

Este sitio: Servicios

Servicios > Contratos > Mantenimiento de computadoras en Empresa 1 > Editar elemento

Contratos: Mantenimiento de computadoras en Empresa 1

Adjuntar archivo | Eliminar elemento | Ortografía... * indica un campo obligatorio

Descripción * Mantenimiento de computadoras en Empresa 1

Cuenta Primera empresa de ejemplo

Tipo de contrato Contrato de mantenimiento de equipo

Inicio del contrato 07/04/2010
DD/MM/AAAA

Fin del contrato 09/10/2010
DD/ ← octubre de 2010 →

Detalle

lu	ma	mi	ju	vi	sá	do
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Creado el 07/04/2010 11:29 por coordinador de...
Última modificación realizada el 07/04/2010 11:29

Hoy es
Jueves, 15 de abril de 2010