

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Modelo para la mejora de las capacidades de comunicación y
aprendizaje mediante el uso de una *red mesh***

Trabajo de graduación presentado por Rodrigo Alejandro Arriaza Alonzo, Alejandro José Chaclán Gudiel, José Rodrigo Custodio Vargas, Álvaro Sebastian Galindo Calderón, Rodrigo Eduardo Barrios Estrada, Andres Alejandro Oliva Murga para optar al grado académico de Licenciados en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,

2019

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería




**Modelo para la mejora de las capacidades de comunicación y
aprendizaje mediante el uso de una *red mesh***

Trabajo de graduación presentado por Rodrigo Alejandro Arriaza Alonzo, Alejandro José Chaclán Gudiel, José Rodrigo Custodio Vargas, Álvaro Sebastian Galindo Calderón, Rodrigo Eduardo Barrios Estrada, Andres Alejandro Oliva Murga para optar al grado académico de Licenciados en Ingeniería en Ciencias de la Computación y Tecnologías de la Información


Guatemala,


2019

Vo.Bo.:

(f) 
MSc. Douglas Barrios

Examinador:

(f) 
MSc. Douglas Barrios

(f) 
Ing. Dennis Aldana

(f) 
Ing. Tomás Gálvez

(f) 
Ing. Melinton Navas

Fecha de aprobación: Guatemala, 29 de noviembre de 2019.

Lista de figuras	XI
Lista de cuadros	XII
Resumen	XIII
Abstract	XIV
1. Introducción	1
2. Antecedentes	4
2.1. Implementaciones actuales	4
2.1.1. RACHEL	4
2.1.2. Sayada, Túnez	4
2.1.3. Macha, Zambia	5
2.1.4. Lancashire, Reino Unido	5
3. Justificación	6
4. Objetivos	7
4.1. Objetivo general	7
4.2. Objetivos específicos	7
5. Marco teórico	9
5.1. Aprendizaje activo	9
5.2. Aprendizaje electrónico	9
5.3. Aplicación Web	10
5.3.1. Comparación aplicación web vs aplicación móvil	10
5.4. Tecnologías para el desarrollo de interfaz web	11
5.4.1. Angular	11
5.4.2. React.js	12
5.4.3. Vue.js	12
5.4.4. Comparación de los marcos de trabajo	13

5.5.	<i>Backend</i> de una Aplicación Web	14
5.5.1.	Arquitectura de tecnologías de la información	14
5.6.	Metodologías de desarrollo colaborativo con control de versiones	15
5.6.1.	Git Flow	16
5.6.2.	Desarrollo basado en Trunk	17
5.7.	Sockets	18
5.7.1.	Web Sockets	18
5.7.2.	Socket.io	18
5.8.	Web RTC	19
5.9.	Sistema distribuido	21
5.9.1.	Objetivos de un sistema distribuido	23
5.10.	Algoritmo <i>rsync</i>	23
5.10.1.	Go/Golang	24
5.10.2.	Goroutines	24
5.10.3.	SQLite	25
5.11.	Experiencia de Usuario	25
5.11.1.	Diseño centrado en las personas	25
5.11.2.	Framework ABCS	27
5.12.	Pruebas de Usabilidad	29
5.12.1.	Descripción general	29
5.12.2.	Crear un plan de pruebas	30
5.12.3.	Ejecutar las pruebas	31
5.12.4.	Analizar la información	32
5.12.5.	Crear reporte de pruebas	32
5.13.	Heurísticas	32
5.14.	Lenguaje <i>markdown</i>	34
5.15.	Estadística	34
5.15.1.	Media	34
5.15.2.	Varianza	35
5.15.3.	Desviación Estándar	36
5.16.	Red	36
5.16.1.	Modelo OSI	36
5.16.2.	Red en malla	37
5.16.3.	Red Comunitaria	37
5.16.4.	Red tipo ad hoc	38
5.17.	Tecnologías para arquitectura de redes	38
5.17.1.	Firmware	38
5.17.2.	Protocolos de enrutamiento	40
5.17.3.	Plan Estratégico	42
5.17.4.	Marco jurídico	43
5.17.5.	Proceso para la gestión de proyectos ante los Consejos de Desarrollo	44
5.18.	Redes WLAN	45
5.19.	Punto de acceso	46
5.20.	Punto de acceso no autorizado	46
5.21.	Portal captivo	47
5.22.	Disponibilidad de redes en TI	48
5.23.	Servidor web	48

5.24. NGINX	49
5.25. Protocolo de Seguridad	50
5.26. Protocolos de cifrado <i>Wi-Fi</i>	50
5.27. Niveles socioeconómicos en Guatemala	52
6. Marco metodológico	54
6.1. Diseño de la interfaz gráfica del sistema de comunicación	54
6.1.1. Prototipo en papel	54
6.1.2. Prototipo inicial	57
6.1.3. Implementación del diseño	58
6.1.4. Pruebas de usabilidad	59
6.1.5. Visualización final de la aplicación	60
6.2. Red tipo malla	63
6.2.1. Definición	63
6.2.2. Investigación	66
6.2.3. Diseño de prototipo	66
6.3. Diseño de la interfaz gráfica de la aplicación educativa	70
6.3.1. Enfoque y variables de investigación	70
6.3.2. Selección de tecnología	70
6.3.3. Prototipado	71
6.3.4. Plan de evaluación	71
6.3.5. Ejecución de las pruebas	73
6.4. Diseño del Backend para el sistema de comunicación	74
6.4.1. Implementación del Backend	75
6.4.2. Funcionalidad del sistema de comunicación	76
6.5. Diseño de la red	77
6.5.1. ¿Quiénes van a usar la red?	77
6.5.2. ¿Para qué se va a utilizar la red?	77
6.5.3. ¿Hasta qué punto se va a expandir la red?	79
6.5.4. ¿Valor de los datos para establecer seguridad?	79
6.5.5. ¿Presupuesto?	79
6.5.6. ¿Disponibilidad de la red?	79
6.5.7. ¿Qué topología?	79
6.6. Seguridad de la red	80
6.6.1. Identificación de recursos y requisitos de la red:	80
6.6.2. Identificación de potenciales riesgos, vulnerabilidades y la propuesta de medidas correctivas	82
6.6.3. Cifrado <i>Wi-Fi</i>	86
6.6.4. Portal captivo	86
6.7. Sistema de alimentación de energía	87
6.7.1. Solar Guatemala	88
6.7.2. SolarGuat	88
6.7.3. Kingo Energy	89
6.8. Acercamiento a las comunidades rurales	90
6.8.1. Identificación de poder dentro de la comunidad.	90
6.8.2. Integración del proyecto con la participación de la comunidad.	91
6.8.3. Involucramiento y retención de los participantes del proyecto.	91

7. Resultados	93
7.1. Sistema de comunicación	93
7.1.1. Diseño e implementación de la interfaz gráfica	93
7.1.2. Implementación del foro de educación activa	95
7.2. Diseño de protocolo	95
7.2.1. Prototipo red tipo malla	96
7.2.2. Red malla final	97
7.2.3. Acercamiento a las comunidades rurales	98
7.2.4. Prototipo de protocolo de comunicación e intervención para la comunidad objetivo	98
7.3. Diseño de la interfaz gráfica de la aplicación educativa	100
7.3.1. Resultados del prototipado	100
7.3.2. Resultados de las pruebas de usabilidad	107
7.4. Integración del sistema de comunicación con la interfaz gráfica de la aplicación educativa	109
7.4.1. Creación, modificación, eliminación y recuperación de anuncios	109
7.4.2. Creación, eliminación y recuperación de salas para mensajería instantánea	112
7.4.3. Creación, eliminación y recuperación de salas para retransmisión de audio y video	113
7.4.4. Funcionalidad del sistema de mensajería instantánea	115
7.4.5. Funcionalidad del sistema de retransmisión por audio y video	116
7.4.6. Integración del Frontend y el Backend	117
7.5. Diseño e implementación del replicador de archivos	119
7.5.1. Arquitectura del sistema objetivo	119
7.5.2. Funcionamiento del replicador	120
7.5.3. Dependencias del replicador y sus usos	121
7.5.4. Configuración del replicador	122
7.6. Diseño e implementación de la interfaz de programación de aplicaciones de la enciclopedia en línea	123
7.6.1. Diseño de los esquemas de la base de datos	124
7.6.2. Diseño e implementación de las peticiones de la interfaz de programación de aplicaciones	128
7.6.3. Procesador de archivos Markdown	133
7.7. Integración del sistema de replicación de archivos con la enciclopedia en línea y la plataforma de aprendizaje mediante foros activos	134
7.7.1. Integración con la enciclopedia en línea	134
7.7.2. Integración con la plataforma de aprendizaje mediante foros activos	136
7.8. Diseño de la red	137
8. Conclusiones	140
9. Recomendaciones	142
10. Bibliografía	146

11. Anexos	151
11.1. Diseño de una <i>red mesh</i>	151
11.2. Realización de la interfaz gráfica para la aplicación educativa	162
11.2.1. Guion para introducción a las pruebas de usabilidad	162
11.2.2. Resultados individuales de las pruebas de usabilidad	163
12. Glosario	165

Lista de figuras

1.	Porcentaje estudiantes inscritos según edad	1
2.	Porcentaje estudiantes en grado adecuado según edad	2
3.	Cantidad de usuarios de internet en Guatemala en los años 2014 y 2018 [6]	3
4.	Disminución en precio (en dólares) por megabit en el periodo 2011–2016 [6]	3
5.	Comparación flujo de información sitio web (izq) vs aplicación web (derecha)	10
6.	Comparación de estrellas en Github a lo largo del tiempo [20]	14
7.	Diferencia entre una arquitectura monolítica y de microservicios.	16
8.	Ejemplo del funcionamiento de Git Flow [27]	17
9.	Ejemplo del funcionamiento de Trunk [28]	17
10.	Comparación de <i>websockets</i> vs <i>poling</i> [30]	19
11.	Estructura triangular de <i>webrtc</i> [32]	20
13.	Sistema distribuido según Tanenbaum, la capa del sistema se comparte entre todas las computadoras y ofrece la misma interfaz a todas las aplicaciones	21
12.	Secuencia de eventos para establecer videollamada [33]	22
14.	Pasos para realizar pruebas de usabilidad	30
15.	Representación de estructura de una red en malla	38
16.	Ejemplo de compromiso accidental de LAN cableada en un entorno corporativo mediante el uso de un punto de acceso no autorizado. [70]	47
17.	Comunicación entre un servidor web y un navegador. [73]	49
18.	Página de inicio	55
19.	Pantalla de inicio de sesión	55
20.	Página principal	56
21.	Pantalla de chat	56
22.	Pantalla de clase en vivo	57
23.	Paleta de colores para la interfaz	58
24.	Izquierda pantalla anuncios, derecha pantalla creación de anuncios	58
25.	Creación de nuevo anuncio usando modal	59
26.	Pantalla de inicio de sesión	61
27.	Pantalla de anuncios desde el rol de administrador	61
28.	Pantalla de chat	61
29.	Vista final de la página de inicio	62

30.	Vista final de la página de inicio de sesión	62
31.	Vista final de la página de anuncios	63
32.	Arriba modal creación, abajo modal actualización	64
33.	Vista final de la página del chat	64
34.	Izquierda persona que inicia la llamada, derecha persona recibiendo la llamada	65
35.	Antenas UAP-AC-M Ubiquity	67
36.	Estructura de red mesh UniFi	68
37.	Probabilidad de precipitación en Ciudad de Guatemala	83
38.	Diagrama de funcionamiento de paneles solares	87
39.	Tiempos promedio para realizar las tareas	94
40.	Cantidad de clicks promedio para realizar las tareas	94
41.	Tasa de éxito para realizar las tareas sin requerir asistencia	95
42.	Se muestran los nodos vecinos dentro de la red malla	97
43.	Primera versión del editor de documentos <i>markdown</i>	101
44.	Segunda versión del editor de documentos <i>markdown</i> : título del documento .	102
45.	Segunda versión del editor de documentos <i>markdown</i> : secciones del documento	102
46.	Tercera versión del editor de documentos <i>markdown</i>	103
47.	Primera versión de la vista previa de documentos	103
48.	Segunda versión de la vista previa de documentos	104
49.	Tercera versión de la vista previa de documentos	104
50.	Primera versión del menú principal de la Wiki	105
51.	Segunda versión del menú principal de la Wiki	106
52.	Segunda versión del menú principal de la Wiki	106
53.	Menú para seleccionar un documento para su visualización o edición	107
54.	Respuesta de servidor a la solicitud GET de anuncios.	110
55.	Cuerpo de la solicitud POST de anuncios.	110
56.	Respuesta de servidor a las solicitudes POST, PUT y DELETE de anuncios en caso de haberse procesado exitosamente.	111
57.	Respuesta de servidor a las solicitudes POST, PUT y DELETE de anuncios en caso de haber ocurrido un error.	111
58.	Cuerpo de la solicitud PUT de anuncios.	111
59.	Respuesta del servidor a la solicitud GET de salas de mensajería instantánea.	112
60.	Cuerpo de la solicitud POST de salas de mensajería instantánea.	112
61.	Respuesta de éxito del servidor a las solicitudes POST y DELETE de salas de mensajería instantánea.	113
62.	Respuesta de error del servidor a las solicitudes POST y DELETE de salas de mensajería instantánea.	113
63.	Respuesta del servidor a la solicitud GET de salas de retransmisión de audio y video.	114
64.	Cuerpo de la solicitud POST de salas de retransmisión de audio y video. . . .	114
65.	Respuesta de éxito del servidor a las solicitudes POST y DELETE de salas de retransmisión de audio y video.	114
66.	Respuesta de error del servidor a las solicitudes POST y DELETE de salas de retransmisión de audio y video.	115
67.	Evento de conexión a una sala en específico.	115
68.	Evento para escuchar nuevos mensajes y la señalización de este al resto de la sala de mensajería instantánea.	115

69.	Evento para detectar nuevos clientes y su respectiva señalización si es o no el transmisor.	116
70.	Evento para escuchar nuevos mensajes y la señalización de este al resto de la sala de retransmisión de audio y video.	116
71.	Evento para escuchar el SDP del cliente y su señalización al transmisor. . . .	116
72.	Evento para escuchar el SDP del transmisor y respectiva señalización al cliente nuevo.	117
73.	Evento para escuchar y señalar el evento <i>ICECandidate</i> del cliente nuevo al transmisor y viceversa.	117
74.	Funcionamiento del sistema de anuncios en la webapp.	118
75.	Funcionamiento del sistema de mensajería instantánea en la webapp.	118
76.	Arquitectura de los sistemas	119
77.	Diagrama de flujo del replicador	121
78.	Ejemplo de archivo de configuración del replicador	123
79.	Ejemplo de archivo de configuración del proceso de <i>rsync</i> en el nodo maestro que funciona con el archivo de configuración de la Figura 78	123
80.	Atributos de la tabla de usuarios	124
81.	Atributos de la tabla de roles	125
82.	Atributos de la tabla usuarios - roles	125
83.	Atributos de la tabla de grados	126
84.	Atributos de la tabla de curso	126
85.	Atributos de la tabla de clases textuales	127
86.	Diagrama entidad-relación utilizada por la interfaz de programación de aplicaciones de la enciclopedia en línea	128
87.	Ejemplo de archivo de Markdown sin procesar	134
88.	Ejemplo de archivo de Markdown procesado	134
89.	Archivo de configuración de <i>rsyncd</i>	135
90.	Archivo de configuración <code>config.toml</code> de replicador para sincronizar la carpeta de archivos Markdown de la interfaz de programación de aplicaciones de la enciclopedia en línea	135
91.	Salida del replicador indicando que se ha logrado sincronizar	136
92.	Salida de la interfaz de programación de aplicaciones de que ha iniciado correctamente	136
93.	Salida del replicador indicando que se ha logrado sincronizar	136
94.	Archivo de configuración de <i>rsyncd</i>	137
95.	Archivo de configuración <code>config.toml</code> de replicador para sincronizar la carpeta de archivos estáticos de la plataforma de foros activos	137
96.	Diagrama de red mesh.	138
97.	Demostración gráfica de la conectividad de los nodos a la red malla "mpmesh" 151	
98.	Configuración de <i>Network</i> , donde se establece una dirección estática para la interfaz de lan.	152
99.	Configuración de las antenas emisoras de wifi, del equipo, cada antena puede emitir una red diferente.	153
100.	Configuración para crear una red que soporte el tipo malla.	154
101.	Configuración de la interfaz de control para B.A.T.M.A.N. dentro de la configuración <i>network</i>	154

102.	Configuración de la interfaz encargada de establecer conexión de forma "física" bajo el control <code>bat0</code> dentro de la configuración. <i>network</i> .	154
103.	Consentimiento informado	155
104.	Consentimiento informado	156
105.	Consentimiento informado	157
106.	Consentimiento informado	158
107.	Diagrama red tipo malla	159
108.	Diagrama red tipo malla con conexión física a internet, por medio de una red existente	159
109.	Diagrama red tipo malla con conexión inalámbrica a internet, por medio de una red existente	160
110.	Diagrama red tipo malla con conexión física a internet, por medio de una red satelital	160
111.	Diagrama red tipo malla con conexión inalámbrica a internet, por medio de una red satelital	161
112.	Diagrama red tipo malla con conexión física a internet, por medio de una red LTE	161
113.	Diagrama red tipo malla con conexión inalámbrica a internet, por medio de una red LTE	162

Lista de cuadros

1.	Comparación entre las distintas tecnologías para desarrollo de aplicaciones . . .	11
2.	Comparación entre los 3 frameworks más usados	13
3.	Ponderación para las heurísticas	33
4.	Sintaxis básica de <i>Markdown</i>	35
5.	Comparación entre los distintos protocolos de enrutamiento	41
6.	Evolución de sistemas de codificación inalámbrica. [80]	52
7.	Resultados de las pruebas de usabilidad	60
8.	Tabla de especificación de antena Ubiquity UAP-AC-M	67
9.	Evaluación de riesgos - IRR	82
10.	Tabla de comparación entre firmware y características	96
11.	Cuadro de compatibilidad entre firmware y protocolo de enrutamiento	97
12.	Tabla comparativa de tiempo de respuesta	97
13.	Completación de las tareas	107
14.	Estadísticas del tiempo invertido por los usuarios para realizar cada tarea (segundos)	108
15.	Comparación de tiempos (segundos) entre usuarios y el desarrollador	108
16.	Cantidad de pasos que tomó cada tarea	109
17.	Cuadro comparativo de proveedores de paneles solares en Guatemala.	138
18.	Resultados de la prueba realizada al usuario 1	163
19.	Resultados de la prueba realizada al usuario 2	163
20.	Resultados de la prueba realizada al usuario 3	163
21.	Resultados de la prueba realizada al usuario 4	164

El objetivo general del trabajo es crear un modelo para introducir a una comunidad servicios de comunicación por medio de una red en malla. La razón de usar una red en malla o una *mesh network*, es porque este tipo de tecnología permite que las personas conectadas a la misma no dependen de tener una conexión a internet, sino que todos los servicios funcionen de manera local.

Por lo tanto, se busca facilitar la comunicación dentro de la comunidad, utilizando tres formas: un tablero de noticias importantes, un chat instantáneo y clases con video y audio. Al mismo tiempo, habilitar servicios educativos, como un foro de educación y una biblioteca virtual disponible en todo momento, para los miembros conectados de la comunidad.

Este módulo se enfoca principalmente en el desarrollo e implementación de la interfaz gráfica de los sistemas de comunicación, utilizando el marco de trabajo llamado Vue.js y la librería WebRTC para el funcionamiento del chat y de las videollamadas.

The main objective of this project is to create and implement a model to introduce communication and education services to a rural community using a mesh network. The reason to use this type of technology is due to its independence to an internet connection, making it possible for the communication services to keep on functioning without internet.

Furthermore, the project aims to facilitate communication between members of the community in three different ways: the creation of a news post, an instant chat and videoconferences for school lessons. Concurrently, it enables an active education forum and a virtual library available at all times.

This module focuses on the design and implementation of the front end of the web application, regarding the communication services, using the Vue.js framework and the library WebRTC.

Actualmente, en las áreas rurales de Guatemala se tiene poco acceso a tecnología debido a las condiciones y ubicación en donde se encuentran. Según un informe de GlobalData, en Guatemala únicamente el 25 % tendrá durante los próximos 5 años línea con tecnología 4G y en el 2018 el consumo promedio de GB al mes fue de 0.4. Esto refleja que por razones socio económicas gran parte de la población tiene poco acceso a internet. [1]

Según las entrevistas realizadas, actualmente para solucionar la falta de una red de internet en las áreas rurales, se almacena la información directamente en los discos duros de las computadoras que se distribuyen a las escuelas. Esto es un proceso largo y tedioso para los que tienen que preparar el contenido a enseñar, debido a que tienen que distribuir la información manualmente, computadora por computadora.

Por otro lado, la cobertura educativa en Guatemala es uno de los principales problemas actualmente, especialmente en el área rural. En el año 2016, a nivel nacional la tasa de escolarización, que indica el porcentaje de menores de edad que están en el sistema educativo fue de 22 %. La división por edad se puede ver en la siguiente gráfica:

Además, en este mismo año se observa que, en el área rural únicamente un tercio de los alumnos tenían la edad apropiada para su grado correspondiente.

[2]

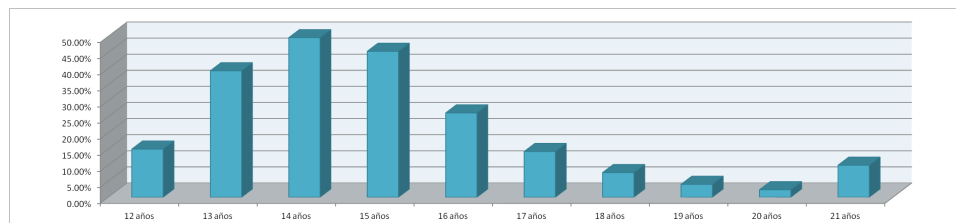


Figura 1: Porcentaje estudiantes inscritos según edad

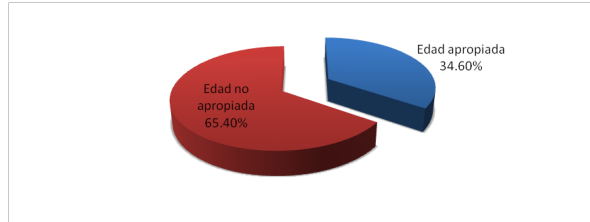


Figura 2: Porcentaje estudiantes en grado adecuado según edad

Según un estudio realizado por la UNESCO, la infraestructura y las condiciones de la escuela afectan el rendimiento de los estudiantes. Del estudio se concluye que “la existencia de espacios complementarios como laboratorios de computación y biblioteca, áreas verdes y áreas de juego en los centros educativos contribuyen al desarrollo cognitivo y psicomotriz de los estudiantes”. [3]

Por lo tanto, se observa que la situación educativa y el acceso a internet son problemas que enfrenta la población rural y que están relacionados entre sí, ya que con una mejor infraestructura tecnológica y con mejores servicios educativos de bajo costo, se podría mejorar el desempeño de los estudiantes. Además, con una mayor cobertura de internet en áreas rurales, se da la oportunidad de acceder a recursos educativos, para las personas que por alguna dificultad no puedan acceder a la escuela.

De la mano, según los datos de la Superintendencia de Telecomunicaciones, en el 2018 hubo más de 20 millones usuarios de telefonía móvil, lo cual indica un alto índice de usuarios con celular. Por lo que se propone generar la infraestructura tecnológica, implementando una red en malla y los servicios, para que las personas puedan usar sus celulares y acceder a internet y recursos educativos.

Guatemala es un país que actualmente se encuentra en el puesto 127 en el índice de desarrollo del Programa de las Naciones Unidas para el Desarrollo [4]. Su sector económico más grande es la agricultura y según el Instituto Nacional de Estadística [5] se estima que alrededor del 51.5% de la población vive en áreas rurales del país. Además según un informe de la CEPAL la pobreza del país excede el 50% haciendo que Guatemala se encuentre dentro los primeros 10 países más pobres de toda América Latina.

Sin embargo, a pesar de estos índices, según *Think with Google* [6], Guatemala no está tan mal en el área de tecnología ya que es el primer país con más usuarios de internet de Centroamérica con 7.2 millones de usuarios, representando esto un 42.0% de penetración de internet, cabe destacar que estos porcentajes se dieron debido a que en el período de 2014–2018, la tasa de crecimiento fue de 167%. Aumentando de la cantidad de usuarios de 2.7 millones a los 7.2 millones actuales tal como se puede ver en la Figura 3.

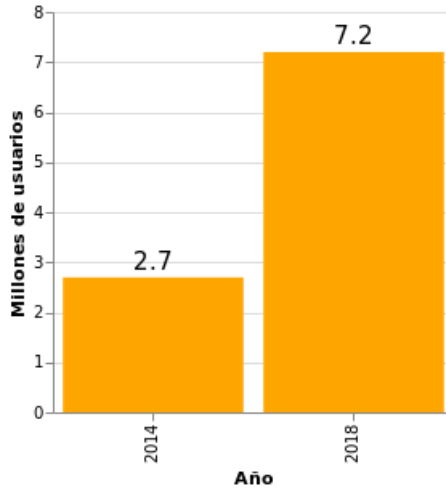


Figura 3: Cantidad de usuarios de internet en Guatemala en los años 2014 y 2018 [6]

Este aumento está fuertemente relacionado a la disminución del precio por megabit [6] a lo largo de los años tal y como se puede observar en la Figura 4

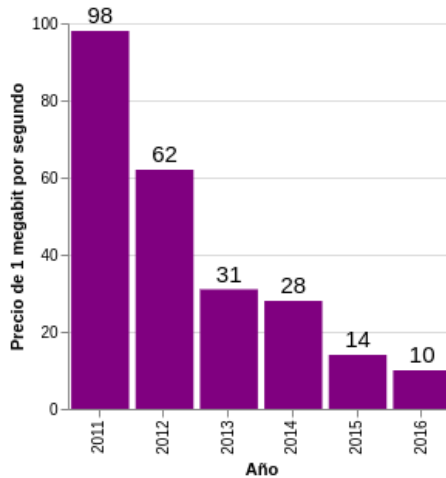


Figura 4: Disminución en precio (en dólares) por megabit en el periodo 2011–2016 [6]

El presente proyecto trata de aprovechar este crecimiento para impulsar uno de los sectores más afectados por la pobreza del país, el sector educativo mediante la implementación de dos servicios educativos que faciliten la comunicación y transmisión de material educativo a un área rural del país utilizando una red en malla de bajos recursos.

2.1. Implementaciones actuales

Luego de introducir lo que se realizó en este trabajo, se presentan otras investigaciones y proyectos similares. Los proyectos y trabajos que se consideraron fueron aquellos que incluían temas relacionados a proveer servicios mediante una red en malla en condiciones precarias. Así como aquellos proyectos que tratan de llevar material educativo a comunidades de escasos recursos.

2.1.1. RACHEL

El proyecto RACHEL es un servidor portable *plug-and-play* el cual se encarga de almacenar páginas web educativas para que puedan ser accedidas de manera local y sin acceso a internet. Lo que realiza el proyecto en RACHEL es descargar las páginas web directamente y almacenarlas en el dispositivo que se haya comprado, este dispositivo expone una aplicación web localmente la cual es posible acceder desde cualquier navegador de algún dispositivo que se encuentre dentro de la misma red.

Actualmente tienen a la venta dos dispositivos disponibles, el RACHEL-Pi y RACHEL-Plus, el primero es el dispositivo más barato posee varias limitaciones de hardware ya que como su nombre lo indica corre sobre una *Raspberry Pi* y según el sitio web del proyecto debería ser utilizado solamente para demostraciones, mientras que el segundo modelo ya es para colocarlo y utilizarlo por largos períodos de tiempo [7].

2.1.2. Sayada, Túnez

Sayada, Túnez, es un pequeño pueblo con gente de pocos recursos, dedicada principalmente a la pesca. Luego de la revolución del 2011, este pueblo decidió formar parte de un

experimento digital y establecer una red local separada físicamente, usando antenas colocadas en varios techos de la comunidad. Esto con el objetivo que las personas se puedan comunicar más fácilmente y con mayor seguridad que en el internet abierto, el cual estaba fuertemente vigilado por el gobierno anterior. Antes de la implementación en vivo en Túnez, se realizaron pruebas piloto en barrios pobres de Detroit y se demostró que esta tecnología podría servir como recurso útil de comunicación en catástrofes naturales, como lo fue el huracán Sandy. [8]

2.1.3. Macha, Zambia

Por otro lado, en la comunidad rural de Macha, Zambia, se llevó a cabo el proyecto de empoderar a su población. Para esto se les capacitó para ejecutar y mantener una red *mesh* administrada localmente. Esta cuenta con 52 nodos mesh y 99 dispositivos proveyendo servicios a la comunidad, haciendo de esta una de las redes locales más grandes de África. Los usos que la población le dan a la red son variados, pero principalmente es usado por miembros del Macha Mission Hospital, cuenta con servicios de e-learning para que sus usuarios puedan atender cursos en línea y algunos agricultores la usan para entender mejor la biodiversidad de sus cultivos y mejorar la cosecha. [9]

2.1.4. Lancashire, Reino Unido

En 2003, residentes de un pequeño pueblo llamado Wray, localizado en Lancashire, Reino Unido, recurrieron a la Universidad de Lancaster para buscar una solución a la falta de internet en su comunidad. La comunidad sentía que la falta de internet estaba afectando negocios locales, la educación y a la comunidad en sí. La Universidad de Lancaster decidió implementar una red *mesh* a lo largo de todo el pueblo. Lograron poner una conexión de 10 Megabytes en el nodo inicial ubicado en la escuela del pueblo. Este proyecto le dio internet a todo el pueblo por 3 años, no solo ayudando a la comunidad, sino que brindó nueva información sobre los problemas técnicos y sociales que conlleva instalar una red *mesh* en una comunidad. [10]

En vista del problema principal que demuestra que la falta de acceso a internet influye en la baja tasa educativa en Guatemala, se propone construir un modelo de un foro de aprendizaje activo que funcione sobre una *Mesh Network*, o red en malla. Esta garantiza el acceso a sus servicios independientemente del acceso a internet y permite que la comunidad esté en constante comunicación.

Debido a la falta de acceso a servicios de internet, especialmente en comunidades pequeñas, existe la necesidad de un servicio auto-sostenible y más simple para facilitar el uso de ciertos recursos en línea. Específicamente, una red en malla permite redundancia, por lo que aumenta la disponibilidad del servicio de internet y permite una mejora en la educación tecnológica.

Según la UNESCO, las TIC contribuyen a un acceso universal educativo, equidad en la educación, calidad de enseñanza y aprendizaje. Además, el uso de la tecnología mejora la gestión, gobernanza y administración de la educación [11]. Por lo tanto, la implementación de un modelo educativo funcionando sobre la red en malla, sería de gran beneficio para la comunidad.

4.1. Objetivo general

Crear un modelo para introducir a una comunidad servicios de información y comunicación por medio de una red en malla adecuada a las condiciones y necesidades existentes.

4.2. Objetivos específicos

- Establecer sobre la red en malla un servicio de comunicación.
- Desarrollar una aplicación *web* educativa, que pueda ser utilizada por profesores escolares para publicar material educativo para sus estudiantes.
- Implementar un foro de educación activa que funcione en una red en malla de pocos recursos
- Crear un foro de aprendizaje activo mediante un servicio de comunicación en línea, que incluya comunicación textual y videollamadas o llamadas de voz.
- Eliminar la necesidad del uso de una aplicación externa del lado del usuario para la transmisión de audio y video.
- Brindar acceso al servicio de comunicación aunque el acceso a internet sea limitado o nulo.
- Diseñar e implementar un sistema distribuido que pueda replicar archivos en servidores que se encuentran dentro de una red en malla de bajos recursos.
- Diseñar e implementar la interfaz de programación de aplicaciones de una plataforma educativa cuyo propósito es asistir a los profesores a crear clases textuales de una enciclopedia en línea.

- Integrar el sistema de replicación de archivos con los servicios educativos en la red en malla, para permitir el acceso a recursos sin necesidad de tener una conexión a Internet constante.
- Estructurar una red mesh para una comunidad.
- Establecer los pasos necesarios que conlleva un protocolo de comunicación e intervención para las comunidades objetivo.
- Diseñar un modelo de *Red Mesh* que cuente con protocolos que garanticen la seguridad de la red y la información en la misma.
- Diseñar un sistema de alimentación de energía que asegure el funcionamiento continuo de los elementos en la red.

5.1. Aprendizaje activo

El aprendizaje activo o *active learning* es una metodología de enseñanza cuyo objetivo es lograr que el estudiante participe y se involucre más en el proceso de aprendizaje. Cuando los estudiantes pueden participar activamente mediante actividades cooperativas y centradas en ellos, pueden retener un mayor porcentaje del material. Varios estudios demuestran que cuando una persona recibe una clase magistral o una presentación solo retiene el 5 %, pero si se le permite participar activamente retiene al menos el 70 %. [12]

Para lograr una metodología de aprendizaje activo se recomienda incentivar la participación de los estudiantes mediante distintas actividades en un ambiente donde se sientan cómodos y sus ideas sean respetadas. Por otro lado, el maestro debe servir como un facilitador y usar diferentes estrategias como hacer actividades uno a uno o en grupos pequeños, así como hacer preguntas constantemente usando diferentes formas de comunicación [12].

5.2. Aprendizaje electrónico

El aprendizaje electrónico o *e-learning*, es según [13], "todas aquellas metodologías, estrategias o sistemas de aprendizaje que emplean tecnología digital y/o comunicación mediada por ordenadores para producir, transmitir, distribuir y organizar conocimiento entre individuos, comunidades y organizaciones".

Un elemento muy importante en el desarrollo de estas metodologías es la implementación de un sistema que provea en la misma plataforma diversos recursos, actividades interactivas, y administrar o gestionar los alumnos, pudiendo asignar calificaciones.

Tal sistema está compuesto por un módulo llamado *ILS (Integrated Learning System)*, que contiene herramientas como chat, aula virtual, foros y área de acceso a contenido; un

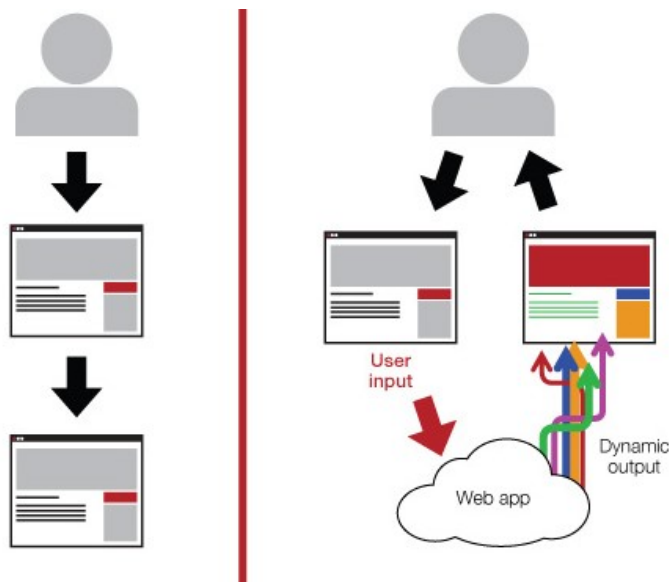


Figura 5: Comparación flujo de información sitio web (izq) vs aplicación web (derecha)

área de gestión de contenidos o LCMS *Learning Content Management System*; y un LMS que lleva registro de la situación de cada alumno. [13]

5.3. Aplicación Web

Una aplicación web es un sistema de software que es accesible mediante la web, o de forma más específica a través de un navegador. Usa tecnologías web y debe usar tecnologías estándar hasta donde sea posible, ya que es común que una *webapp* deba comunicarse con otras aplicaciones web. Además, es importante mencionar que una aplicación de este tipo se le considera como un programa altamente dinámico. [14]

En los primeros años del Internet, la web consistía únicamente de páginas web. Estas son en esencia, repositorios de información que contienen documentos estáticos. Los navegadores web o *web browsers* servían el propósito de acceder a estos repositorios y mostrar los documentos de una forma visualmente aceptable. De esta forma, el flujo de información era de una sola vía: del servidor web al browser. [15]

Actualmente, la mayoría de sitios web son ya realmente aplicaciones web. Son altamente funcionales y dependen de una comunicación de dos vías entre el servidor y el cliente. A diferencia de una página estática, una aplicación soporta registro, inicio de sesión, creación de contenido por un usuario e incluso transacciones financieras, entre otros. [15]

5.3.1. Comparación aplicación web vs aplicación móvil

Un caso de uso especial para las aplicaciones web, es cuando se desarrolla para dispositivos móviles. Dado que actualmente, casi cualquier celular tiene un navegador web, un

usuario puede, desde su celular, acceder a páginas y contenido en línea. Sin embargo, con el auge de los *smartphones*, también surge otra metodología de desarrollo específica para estos dispositivos. A esto se le conoce como aplicaciones nativas y están diseñadas para un sistema operativo específico.

Característica	Web	Nativo
Lenguajes	JS, HTML, CSS	Java, Kotlin, Swift
Plataforma	Múltiple	Android o iOS de forma separada
Distribución	Sin intermediario	A través de app stores
Soporte	Solo para una plataforma	Múltiples plataformas en simultáneo
Performance	Más lento	Rápido por conexión con OS

Cuadro 1: Comparación entre las distintas tecnologías para desarrollo de aplicaciones

[16]

5.4. Tecnologías para el desarrollo de interfaz web

Un *web framework* es un entorno de trabajo cuyo objetivo es facilitar el desarrollo de aplicaciones web. Ayuda al desarrollador a evitar tareas repetitivas y aumenta su productividad. Se basa en principios como DRY (*Don't Repeat Yourself*) y KISS (*Keep It Simple*) [17]. Actualmente existen varios marcos de trabajo, pero los más utilizados se listan a continuación:

5.4.1. Angular

Angular fue desarrollado y es usado por Google y otras empresas como Paypal y Microsoft. Es un framework de JavaScript, pero también involucra el uso de TypeScript, el cual es un subconjunto de Javascript. Hay varias versiones de Angular, pero en la actualidad Angular 4 es la más usada. [18]

Entre las principales ventajas que tiene Angular se encuentran:

- Soporte de Single Page Applications: donde se carga solo una página html y los componentes se actualizan dinámicamente, lo cual provee una mejor experiencia reduciendo el tiempo de carga percibido
- Modularidad: El código se divide en bloques llamados módulos, los cuales pueden ser reutilizados en diferentes partes de la aplicación. Esto ayuda a que en proyectos muy grandes, se dividan los equipos y trabajen en componentes distintos.
- Interfaz declarativa: Angular usa el lenguaje de marcado HTML (*HyperText Markup Language*) para definir la interfaz de una aplicación. HTML es más intuitivo y declarativo que JavaScript, por lo que facilita organizar la estructura.

5.4.2. React.js

React no es realmente un *framework*, sino una biblioteca de Javascript para construir interfaces web. Sin embargo, muchas personas lo consideran un *framework* y se le compara en ese contexto [19]. React fue creado y es usado por Facebook e Instagram.

Debido a que React solo se encarga de la parte de la interfaz, el proceso de desarrollo requiere de agregar un compilador para los archivos .JSX, distintos módulos para la estructura, un método de declarar las rutas y un manejador de estados. Para el compilador, por lo general se utiliza Babel y para el manejo de módulos Webpack [20].

Un elemento importante de React es el manejo de estados. El estado es responsable para el manejo de datos dentro de un componente. Una de las funciones principales de React es *setState()*. Cuando esto se ejecuta se dispara un método que vuelve a renderizar la interfaz del componente [20].

Al igual que Angular tiene la ventaja de tener una naturaleza declarativa para la interfaz, pero a diferencia de, el flujo de información de la aplicación es sólo de una vía, de tal manera que solo el componente padre puede actualizar el componente hijo y no al revés [21].

5.4.3. Vue.js

Vue fue creado por Evan You, extrabajador de Google, bajo la filosofía de tomar las cosas que más le gustaban de Angular y convertirlo en algo más liviano. Una de sus mejores características es su naturaleza progresiva, que permite añadir sólo las funcionalidades que sean necesarias de forma modular. Esto facilita el desarrollo y permite un funcionamiento más eficiente del lado del cliente. Además, esta naturaleza progresiva, permite que se pueda usar en proyectos pequeños y grandes [22].

A diferencia de React, Vue cuenta con *two-way data binding* o unión de la información de dos vías, de tal manera que al usar la directiva v-model, los datos se sincronizan con cualquier evento de entrada de datos, cada componente evalúa si ha cambiado alguna entrada que le indique si debe volver a renderizar. Con esto logra optimizar el tiempo de carga del DOM, que es el Modelo de Objetos del Documento por sus siglas en inglés. [23].

Además,Vue es un *framework* basado en componentes y cada archivo .vue cuenta con tres partes:

- `<template></template>`: La parte donde se declaran los elementos de HTML a incluir, como `<div>`, `<h1>`, `<p>`,...
- `<script></script>`: En esta parte se incluye la funcionalidad usando javascript
- `<style></style>`: Aquí se modifica la forma visual, agregando css al componente.

De forma interna, la plantilla o *template* se compila en las funciones de renderización del DOM virtual. Esto significa que se guarda una copia en memoria del DOM original, a la cual se le denomina DOM virtual. Al combinar esto con el sistema de reactividad, Vue

es capaz de determinar la cantidad mínima de elementos que se deben volver a dibujar y aplicar menos modificaciones al DOM. [20]

5.4.4. Comparación de los marcos de trabajo

Característica	Angular	React	Vue
Lanzamiento	2010	2013	2014
Tamaño aproximado (kB)	500	100	80
Responsable mantenimiento	Google	Facebook	Evan You
Lenguaje	TypeScript	JavaScript	JavaScript
Flujo de datos	Bidireccional	Unidireccional	Soporta ambos
Curva de aprendizaje	Alta	Mediana	Baja
Ideal para proyectos	Grandes	Grandes/medianos	Medianos/pequeños

Cuadro 2: Comparación entre los 3 frameworks más usados

De forma general, los tres son bastante similares, ya que todos comparten generalidades en su filosofía, lo cual se demuestra en su naturaleza basada en componentes. Sin embargo, una diferencia notable es que Angular de cierta forma obliga a los desarrolladores a usar TypeScript, ya que toda su documentación está en este lenguaje. Aunque esto pueda ser una ventaja para los desarrolladores con conocimiento en programación orientada a objetos, esto genera que proyectos simples se vuelvan complejos. De igual forma React recomienda fuertemente que los archivos sean .jsx, lo cual puede resultar extraño, ya que se genera una mezcla entre JavaScript y HTML. Por otro lado, Vue le da más libertad a los programadores al no imponer ninguna restricción.

De igual manera, Angular y Vue concuerdan al implementar la estructura de sus componentes en tres partes: template, script y style, pero Angular recomienda usar archivos distintos para cada parte, mientras Vue no, ya que se argumenta que al ser un solo archivo, insta al desarrollador a escribir un código más delgado.

Otro aspecto importante a considerar, es que de estos tres, Angular es el único que es verdaderamente un framework, ya que cubre más etapas del desarrollo, mientras que Vue y React son en realidad librerías que se comparan en este contexto porque ofrecen las mejores prácticas, y también tienen otras herramientas recomendadas para cubrir otros aspectos del desarrollo como:

- Manejo de estados: Vue con Vuex, React con Redux
- *Routers* o manejo de rutas: Vue con Vue-Router y React con React-Router

En cuanto a la curva de aprendizaje, Vue tiene la menor de las tres. Se enfoca bien en las buenas prácticas comprobadas y cuenta con una excelente documentación en línea co-creada por la comunidad. React tiene una curva media, la cual es más sencilla para los desarrolladores con profunda experiencia en JavaScript. Finalmente, Angular tiene la más empinada, y aunque al principio pueda parecer sencillo, tiene ciertos conceptos que pueden ser bastante complejos de dominar [20].

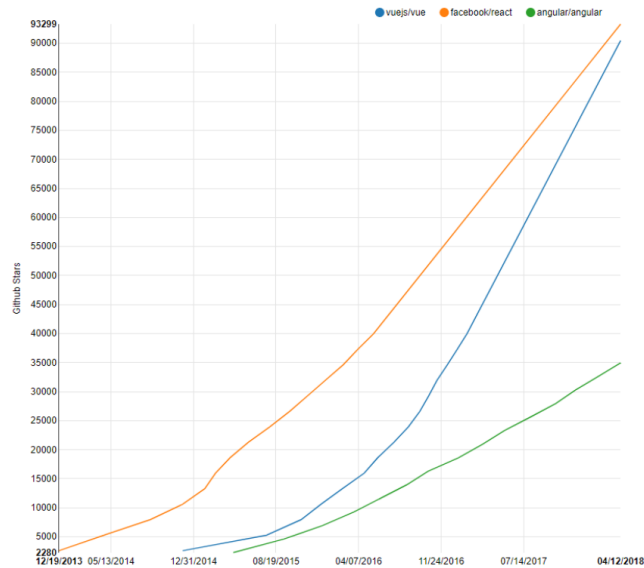


Figura 6: Comparación de estrellas en Github a lo largo del tiempo [20]

5.5. *Backend* de una Aplicación Web

Es la parte de la aplicación no visible pero vital para su funcionamiento. Algunas de las acciones que lleva a cabo el *Backend* es el desarrollo de forma modular para simplificar procesos, conexión con la base de datos o con otros servidor, además de velar por la seguridad del sitio en cuestión.

5.5.1. Arquitectura de tecnologías de la información

El mayor objetivo de una arquitectura es reducir el número de incertidumbres a la hora de describir las propiedades esenciales de un sistema, o solución. Dichas incertidumbres pueden ser creadas por la complejidad del sistema que se quiere desarrollar, pero también, por el nivel de innovación que se quiere introducir. En sistemas complejos normalmente se necesitan lidiar varios parámetros, varios componentes, diversas interfaces hacia otros sistemas, entre otros. Por lo tanto, se puede decir que la complejidad y la innovación son medidas para la dificultad de la implementación de un sistema en su ambiente.

Proyectos que son complejos requieren de una arquitectura sólida, normalmente no diseñada únicamente por un arquitecto, sino por un equipo multifacético, capaz de analizar el problema desde varias perspectivas. Cuando se introduce nuevas herramientas o complejidades técnicas, una arquitectura es necesaria y es la clave del éxito del proyecto.

La implementación de una arquitectura al dominio de una aplicación ayudará a proveer que sus propias características pongan en peligro la estructura existente de sistemas y aplicaciones. Aplicaciones complejas necesitan de una guía. Estas se benefician de un enfoque que explore las posibilidades y desafíos de su estructura contra lo que se busca de la funcionalidad de esta [24].

Arquitectura monolítica

Este tipo de arquitecturas están divididas por capas con la diferencia entre la de micro-servicios es que esta posee un único ejecutable lógico. Esto quiere decir que la aplicación se encuentra almacenado en un único servidor [25]. Algunas ventajas de este tipo de arquitecturas son:

- Facilidad de desarrollo: Tomando en cuenta que es el tipo más común, se encuentra una enorme cantidad de opciones de Entorno de desarrollo integrado (IDE).
- Simplicidad de las pruebas.
- Sencillez de despliegue.
- Viabilidad de escalado: Aunque aumente la cantidad de carga que se ve a ejecutar en la app, se puede modificar una duplicación de servidores tras un balanceador de carga.

Una de las desventajas más relevantes es el cambio que se debe de hacer desplegando la aplicación completa, lo que también afecta directamente las pruebas funcionales.

Arquitectura de microservicios

En una arquitectura de microservicios su funcionamiento radica en la división de módulos independientes y se comunican entre sí mediante mecanismos de interacción tales como APIs o RPC(*Remote Procedure Communications*). El escalado funcional el cual es la base de los microservicios, consiste en que cada una de las divisiones puedan evolucionar sin necesidad de afectar al resto [25]. Parte de las ventajas que presenta este tipo de arquitecturas son las siguientes:

- La cantidad de información que se maneja por servidor es menor.
- Las transacciones de información son más rápidas.
- Las caídas de un servidor afectan menos usuarios.
- Consumo menor de memoria.

Sin embargo, como todo, posee algunos inconvenientes tales como que la complejidad es mayor, ya que hay que idear una estrategia diferente para cada partición. Un ejemplo de la diferencia tanto en la arquitectura Monolítica y de Microservicios se puede ver en la Figura 7.

5.6. Metodologías de desarrollo colaborativo con control de versiones

Un sistema controlador de versiones es capaz de guardar los cambios hechos a un archivo o a un conjunto de archivos, y permitirle al usuario regresar a una versión anterior. Es un

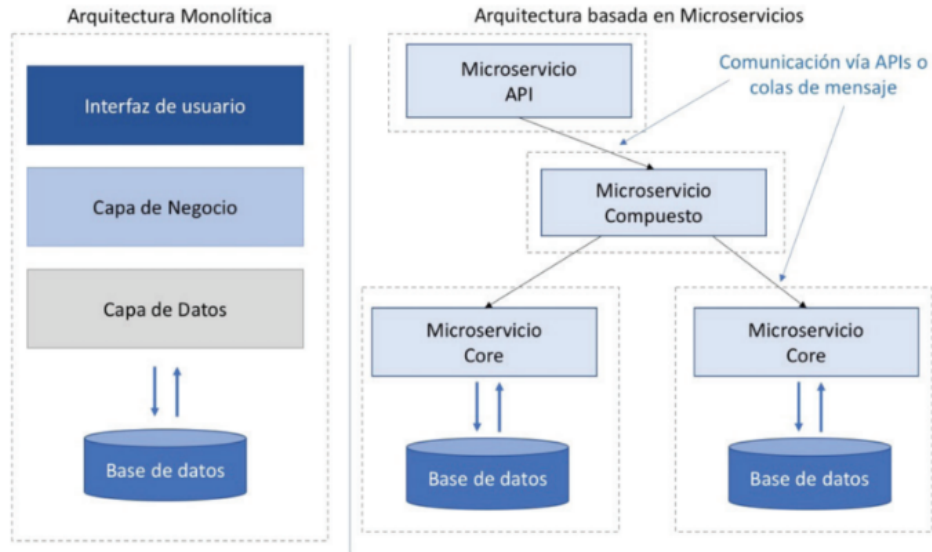


Figura 7: Diferencia entre una arquitectura monolítica y de microservicios.

software que monitorea cambios en los archivos y permite agregar notas sobre estos para identificarlos fácilmente. Con la implementación de un controlador de versiones se evita tener varias copias del mismo archivo pero con nombres diferentes como "proyecto v1.doc", "proyecto v2.doc", etc. [26].

Existen varios sistemas de control de versiones en el mercado, pero el más usado es Git, proyecto iniciado por Linus Torvalds. La arquitectura de este sistema está diseñada pensando en velocidad, atomicidad, performance y seguridad. [26].

Dentro de Git, un mismo proyecto puede tener varios *branches* o ramas, además de la rama principal. Estas pueden ser utilizadas de varias formas como para trabajar una nueva característica por aparte, o cuando un proyecto maneja diferentes versiones por separado. Para el uso correcto de *branches*, existen varias metodologías de las cuales las dos más usadas se describen a continuación.

5.6.1. Git Flow

El flujo de trabajo de Git Flow usa un repositorio central como un hub de comunicación para todos los desarrolladores. Cada desarrollador trabaja localmente y sube sus cambios en un branch al repositorio central.

Git Flow maneja dos ramas principales: master y develop. En el master branch se almacenan las versiones oficiales que se han publicado, mientras que la rama de desarrollo sirve para integrar nuevas características. Cada característica o *feature* nueva, tiene que nacer de la rama de desarrollo y trabajarse en una rama aparte. Cuando se termina un *feature* se regresa a develop y después de una cantidad definida por el equipo de características nuevas, se publican los cambios a la rama principal. [27]

Adicionalmente, se incluye una rama de *release* que sirve para preparar un nuevo lanza-

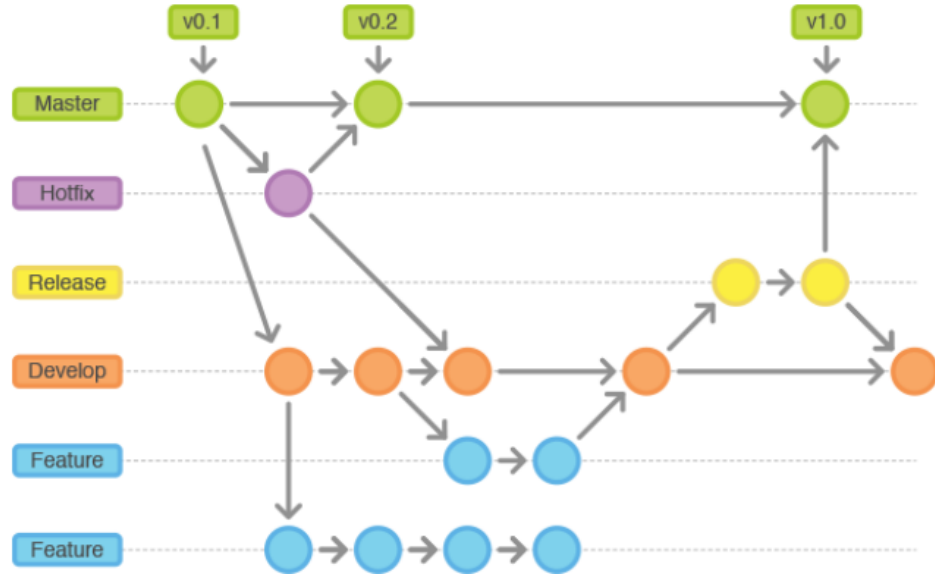


Figura 8: Ejemplo del funcionamiento de Git Flow [27]

miento y una rama *hotfix* para cuando se descubre un error en la versión de producción y se debe reparar rápidamente.

5.6.2. Desarrollo basado en Trunk

En esta metodología, solo hay una rama principal, la cual recibe el nombre de *trunk* la cual es el equivalente a *master*. El objetivo principal de esta metodología es evitar conflictos al unir versiones. Cada desarrollador sube sus cambios directamente a la única rama principal. Estos cambios deben ser pequeños y deben pasar antes una prueba pre-integración para asegurarse que no se esté subiendo un cambio que provoque un error.

Cuando se está trabajando en equipos grandes, sí se permite que se creen ramas aparte por cada *feature*, pero estas deben tener un tiempo de vida corto y luego regresar a *trunk*. Luego de una cantidad determinada de cambios, se toma la última versión y se publica un nuevo *release* [28].

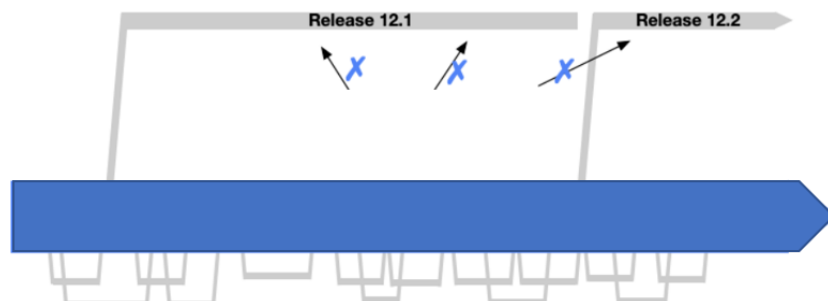


Figura 9: Ejemplo del funcionamiento de Trunk [28]

5.7. Sockets

Un *socket* es una estructura básica de comunicación entre procesos. Se define como una tupla de dirección IP y puerto, por ejemplo: 192.182.0.0:80. Sin embargo, para lograr establecer un canal de comunicación se necesita una pareja de sockets, de tal forma que se tenga la tupla local y la tupla extranjera. La mayoría de aplicaciones que usan TCP y UDP para establecer una comunicación lo hacen creando sockets y luego se definen operaciones sobre estos [29]. Entre las operaciones que se pueden realizar en un socket están:

- Operaciones de control: Asociar un número de puerto a un socket, o iniciar, aceptar y cerrar una conexión
- Transmisión de datos: Escribir y leer datos a través del socket a otra aplicación.
- Operaciones de status: Obtener la dirección IP asociada al socket

5.7.1. Web Sockets

Normalmente, en aplicaciones cliente-servidor, por cada interacción que se requiera entre las dos partes, se establece una conexión, se envía una nueva solicitud del cliente al servidor y luego se recibe la respuesta. A esta interacción se le conoce como *polling*. Sin embargo, en sistemas en tiempo real donde no se puede predecir un intervalo de actualización de datos, usar esta técnica puede ser muy tardado y costoso, ya que puede ser que la información en el servidor no esté disponible cuando el cliente haga la solicitud y se deba crear cada vez una nueva conexión.

Por otro lado, al utilizar *web sockets*, una vez se haya establecido la primera conexión, el servidor enviará la información cada vez que ocurra el evento, sin que el cliente le tenga que pedir la actualización. Esto reduce la latencia y permite que cualquiera de las dos partes envíe solicitudes sin depender de la actualización de la otra parte. [30]

Por ejemplo, si un chat en tiempo real utiliza polling, se tendría que preguntar al servidor cada cierta cantidad de tiempo si hay algún nuevo mensaje. Esto generaría muchas solicitudes innecesarias, ya que la mayoría del tiempo el servidor respondería que no y no se actualizaría el cliente, o posiblemente el intervalo sea muy largo y el usuario no reciba los mensajes de forma instantánea. Mientras que al utilizar web sockets, el servidor le avisaría al cliente que los datos ya han cambiado.

Como se observa en la Figura 10, para establecer comunicación cliente-servidor con *web sockets*, solo se envía una vez la solicitud de conexión y se reutiliza hasta que se desee cerrar. Esta conexión es bidireccional. Esto reduce considerablemente la latencia, ya que el servidor puede enviar información al cliente sin esperar por una solicitud. [30]

5.7.2. Socket.io

Socket.io es una herramienta para crear aplicaciones en tiempo real con comunicación bidireccional entre el cliente y el servidor. Utiliza la tecnología de *web sockets* y además

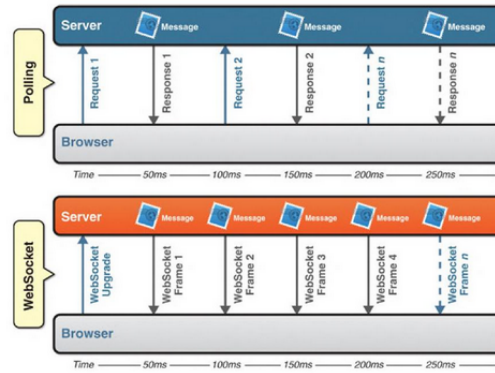


Figura 10: Comparación de *websockets* vs *polling* [30]

tiene la capacidad de recaer a polling si no hay Web Sockets disponibles. Es una librería *open source* creada por Guillermo Rauch utilizando la tecnología de *engine.io*, la cual es una capa de abstracción encima de Web Sockets. [31]

Una de las ventajas principales de esta librería es que tiene una sintaxis tan simple que pareciera que únicamente reacciona a eventos. Además, es posible utilizar *Nodejs* con *socket.io*, lo cual permite tener la misma sintaxis en el cliente y en el servidor.

5.8. Web RTC

Web Real Time Communication es un estándar que permite la comunicación *peer to peer* a través de un navegador web con HTML5. Es una evolución disruptiva, ya que da el poder a los desarrolladores web de construir aplicaciones con comunicación multimedia sin usar extensiones propietarias [32].

WebRTC extiende la ideología de una aplicación web cliente-servidor, al establecer una comunicación *peer to peer*, lo cual significa que se establece un canal directo entre dos usuarios. Cada uno ejecuta el cliente, que por lo general es el mismo programa obtenido de la misma aplicación. Se inicia la conexión enviando señales al servidor mediante *web sockets*. Luego, cuando ya se tiene una conexión entre los dos clientes, la información puede fluir directo de cliente a cliente sin intervención del servidor. Una arquitectura de esta forma se observa en la Figura 11

El API de WebRTC aprovecha las funcionalidades de los navegadores web de comunicación en tiempo real, la cual provee lo necesario para establecer canales de audio y video. Estos flujos de información de audio y video son encriptados antes de ser enviados. Esta comunicación se basa en tres conceptos principales: *Media Stream*, *Peer Connection* y *DataChannel*. [32]

- **Media Stream:** Es una abstracción de un flujo de datos de audio y video. Esta representación sirve para poder manipular el flujo de datos, de tal forma que se pueda mostrar, grabar o enviar a otro nodo. Si el flujo viene de otro nodo se le denomina flujo remoto o *remote stream*, y si es el flujo del nodo local se le llama flujo local o *local stream*. Este

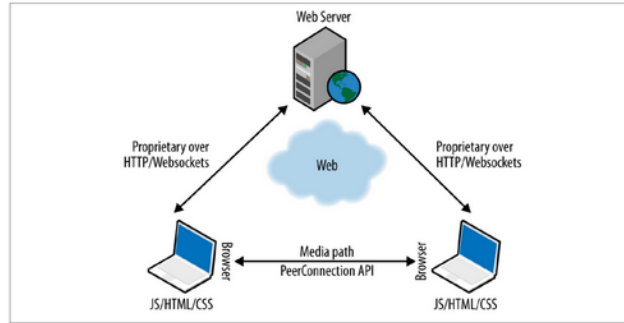


Figura 11: Estructura triangular de webrtc [32]

último representa los datos capturados por una cámara web o micrófono. La aplicación web debe solicitar permisos al usuario para acceder al hardware, mediante la función `getUserMedia()`

- Peer Connection: permite la comunicación directa entre dos usuarios de navegador a navegador. Se coordina mediante un canal de señales con el servidor web, normalmente a través de web sockets.
- DataChannel: Diseñado para proveer un servicio de transporte genérico, de tal forma que los navegadores web puedan intercambiar datos de forma bidireccional. En este canal, se provee una solución transversal que garantiza la confidencialidad, autenticación e integridad. Cuando se establece la conexión *peer to peer*, se crea un *DataChannel*, el cual sirve para enviar y recibir el flujo de datos.

Para poder establecer una conexión *peer to peer*, se utiliza la interfaz *RTCPeerConnection* que conecta el cliente local con un usuario remoto. La interfaz provee métodos para iniciar la conexión, mantenerla, monitorearla y finalmente cerrarla. Una de las funciones principales es *RTCPeerConnection.createOffer()*, la cual crea una oferta de SDP (*SessionDescription-Protocol*), que es el estándar para describir una conexión *peer-to-peer*. Esto contiene el *codec*, dirección del origen, y la información del audio y video. A la interfaz *RTCPeerConnection* se le añaden las pistas de audio y video locales y remotas. Además, se requiere que se configure cual es el SDP local y remoto usando las funciones *RTCPeerConnection.setLocalDescription* y *RTCPeerConnection.setRemoteDescription*.

Signaling Server

Aunque *WebRTC* permita establecer la conexión entre dos clientes, es necesario un tercer elemento que le pueda indicar a los clientes cómo encontrarse. Este proceso se conoce como *Signaling* o señalización. El trabajo de un servidor de señalización es servir como intermediario entre los dos *peers* para que estos puedan establecer una conexión y al mismo tiempo minimizar su exposición de compartir información sensible.

WebRTC no obliga al desarrollador a usar un mecanismo en específico para comunicarse con el *Signaling Server*. No obstante, sí se debe usar el *framework ICE*, que significa *Interactive Connectivity Establishment*. Este sirve para que los dos dispositivos se puedan encontrar

a sí mismo, sin importar la topología de la red e incluso si el dispositivo está dentro de una NAT [33].

El proceso de establecer una conexión para una videollamada se puede ver en 12, en donde se muestra un ejemplo de los eventos que suceden cuando una persona (ej: Naomi) llama a otra (ej: Priya). Además se observa en qué momentos se utilizan las funciones de *RTCPeerConnection* mencionadas anteriormente.

5.9. Sistema distribuido

Dado que el objetivo de este trabajo incluye que los servicios educativos desarrollados funcionen en una red de tipo malla que no siempre tendrá acceso a internet, se necesita de un sistema que sea consistente en cada uno de los nodos en donde sera desplegado, para que así los usuarios de cada nodo tengan acceso a la misma información. Para esto es necesario tener un mecanismo de sincronización en todas las replicas de los servicios que actúe al momento en que la red tenga acceso a internet. Esto se puede lograr mediante la implementación de un sistema distribuido que se encargue de sincronizar recursos.

De acuerdo con Tanenbaum, “Un sistema distribuido es una colección de computadoras independientes que se le presentan a sus usuarios como un solo sistema coherente” [34]. Esta definición incluye varios aspectos importantes, el primero es que los componentes dentro del sistema distribuido son autónomos, es decir manejan sus recursos por si solos. El otro aspecto es que debido a que todos los componentes se tienen que mostrar al usuario como un solo sistema, debe de existir un tipo de comunicación que permita que todos estos puedan colaborar de cierta manera para resolver las solicitudes de los usuarios.

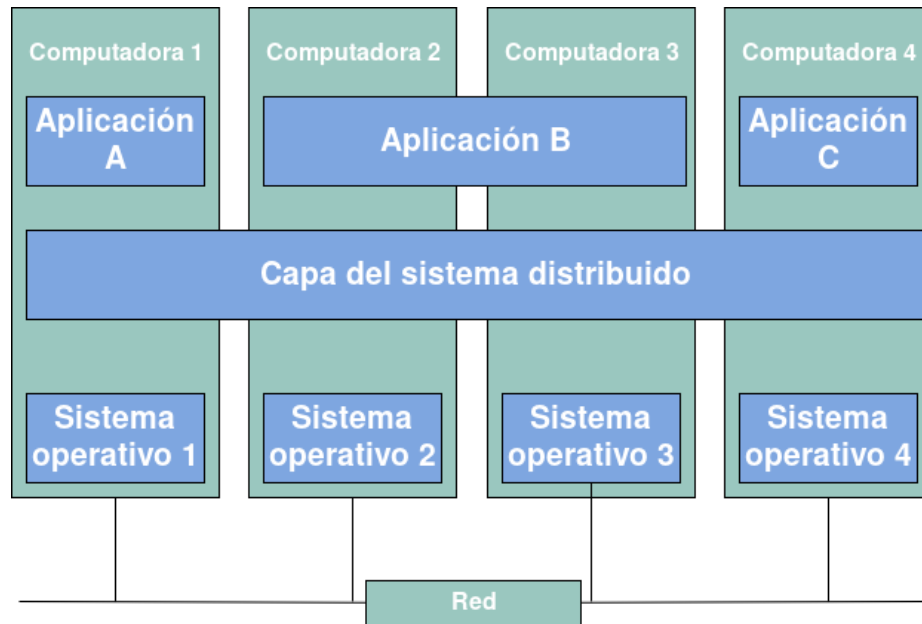


Figura 13: Sistema distribuido según Tanenbaum, la capa del sistema se comparte entre todas las computadoras y ofrece la misma interfaz a todas las aplicaciones

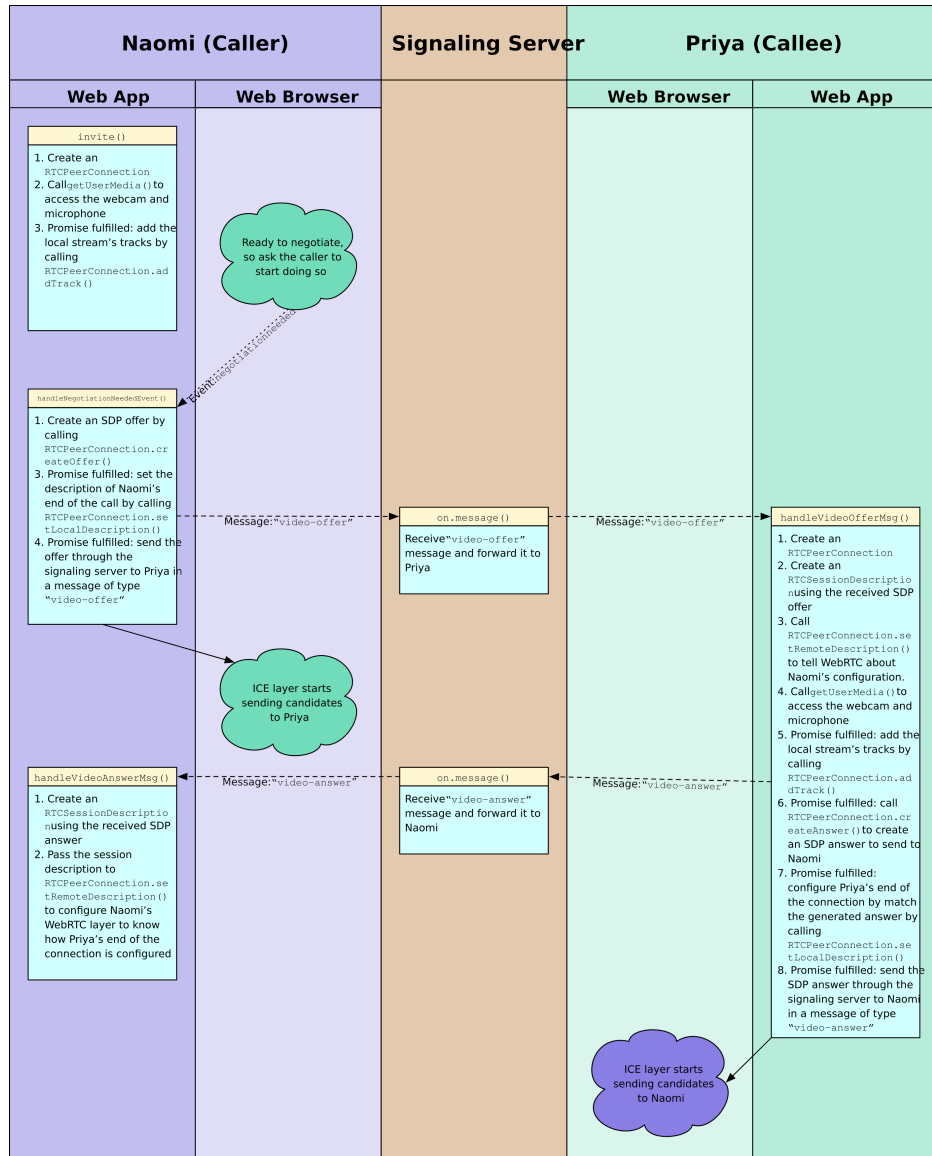


Figura 12: Secuencia de eventos para establecer videollamada [33]

5.9.1. Objetivos de un sistema distribuido

- Los recursos deben ser accesibles: Uno de los objetivos principales de un sistema distribuido es exponer recursos remotos a un usuario de una manera fácil, para que este pueda hacer uso de ellos a su gusto.
- Debe ser transparente: Para lograr que los recursos puedan ser accedidos fácilmente el sistema debe esconder el hecho que los recursos están distribuidos a través de múltiples computadoras. Un sistema que se le presenta a sus usuarios como que fuera una única instancia se dice que es un sistema transparente.
- Abierto: Un sistema distribuido debe ser abierto ya que debe ofrecer sus servicios de acuerdo a un conjunto de reglas de algún estándar que describa la sintaxis y la semántica de dichos servicios.
- Escalable: Un sistema distribuido siempre debe permitir expandir sus recursos para acoplarse a los requerimientos de los usuarios. Es posible medir la escalabilidad de un sistema en tres distintas dimensiones, la primera es respecto a su tamaño, lo que significa que se pueden agregar más usuarios y recursos al sistema. La segunda es respecto a su ubicación geográfica, lo que indica que los componentes del sistema pueden residir a distancias largas, y la tercera es la escalabilidad en su administración, la que significa que el sistema puede seguir siendo fácil de administrar mientras más recursos se le agregan.

5.10. Algoritmo *rsync*

Rsync es un algoritmo publicado por Tridgell [35] en su tesis doctoral “Efficient Algorithms for Sorting and Synchronization” en la cual lo describe como “un algoritmo que opera intercambiando bloques de información firmados seguidos por un algoritmo de búsqueda utilizando un *hash* simple para re ordenar arbitrariamente dichos bloques”

El algoritmo fue diseñado con los siguientes tres objetivos en mente:

1. Velocidad
2. Buena utilización de memoria
3. Flexibilidad

Dado que el algoritmo fue desarrollado debido a la frustración que Andrew Tridgell sentía al enviar cambios de un código fuente a través de una red limitada, en su diseño se estableció que tuviera las siguientes propiedades:

- El algoritmo debe de funcionar en datos arbitrarios, no solamente texto.
- La información total transmitida debe ser del tamaño del archivo de diferencia comprimido.

- Debe ser rápido para archivos y colecciones grandes.
- El número de viajes de ida y regreso de los paquetes transmitidos debe ser la cantidad mínima posible para disminuir la latencia.
- De ser posible el algoritmo debe ser barato en recursos computacionales.

Estas propiedades hicieron que *rsync* fuera el algoritmo indicado para este proyecto ya que satisface la necesidad de ser funcional en una red de pocos recursos.

Cabe destacar que en la actualidad el término *rsync* se refiere a dos elementos a la vez:

- El algoritmo descrito en esta sección
- Una herramienta que se encuentra en la mayoría de sistemas operativos Linux y *Unix-like* que implementa el algoritmo descrito anteriormente y que es utilizado para la transferencia de archivos locales y remotos.

5.10.1. Go/Golang

Para el desarrollo del sistema distribuido encargado de replicar archivos en varios servidores de la red en malla y para interfaz de programación de aplicaciones de la aplicación web para clases textuales se decidió utilizar el lenguaje de programación *Go*, también conocido como *Golang*.

Go es un lenguaje fuertemente tipificado, estático y compilado; el cual fue creado con la motivación de querer tener un lenguaje expresivo, conciso, limpio y eficiente [36]. La fuerza principal de *Go* es su mecanismo de concurrencia que al ser fácil de utilizar y liviano en recursos computacionales, lo hace ideal para aprovechar todos los recursos de computadoras de varios núcleos y computadoras dentro una red. Debido a esta última característica, gigantes tecnológicos como Uber [37] y Netflix [38] han optado por utilizar el lenguaje en varios de sus proyectos internos y de código abierto.

5.10.2. Goroutines

Las *goroutines* son funciones o métodos que se ejecutan concurrentemente con otras *goroutines* en la mismo espacio de direcciones, cabe destacar que no son lo mismo que un hilo, corrutinas, procesos, etc. Ya que funcionan de manera muy distinta, debido a que son ejecutadas en el entorno de ejecución del lenguaje y no en el sistema operativo, y además son multiplexadas con los hilos del sistema operativo. Las *goroutines* son bastante ligeras comparado con otros mecanismos de concurrencia de otros lenguajes como los hilos de Java, puesto que las *goroutines* cuestan un poco más que alojar un espacio para pila, aproximadamente 8KB, mientras que en Java cuesta arriba de 256KB. [39]

5.10.3. SQLite

SQLite como lo define su documentación [40] es una librería que implementa un motor de base de datos transaccional sin muchas dependencias, sin servidor, sin archivos de configuración. A diferencia de otras bases de datos, SQLite posee rasgos comparables entre otros sistemas de lectura y escritura de archivos o bien, que lo hacen diferente de otros motores de base de datos SQL.

Una herramienta que no necesita de instalación ni configuraciones iniciales para funcionar. No hay necesidad de un administrador para iniciar instancias de base de datos ni para asignar permisos en los usuarios. Un diferenciador se encuentra en la utilización de un archivo de disco único para el manejo de la lectura y escritura de datos siempre y cuando el archivo de disco y el directorio que lo contiene tienen permiso de escritura. Otras tecnologías SQL manejan sus datos como una colección de archivos en un directorio accesible únicamente por el administrador de dicha base, lo cual aumenta la seguridad y puede mejorar en rendimiento si se configura escritura directa a archivos de disco, pero su nivel de complejidad en el mantenimiento de la base de datos y el costo de configuración inicial incrementa de igual manera. SQLite hace este proceso menos complejo y más eficiente con la configuración inicial [40].

Aunque se oye arriesgado que SQLite no posea ningún servidor y que se deba colocar los datos en un único equipo, la base de datos se ve beneficiada en comunicación optimizada con los archivos de disco y evitando otras vulnerabilidades de seguridad como *sniffing*¹ de paquetes a través de la red.

Una peculiaridad de SQLite que llama mucho la atención es que dependiendo del uso y la forma en que se compiló la librería, es decir que opciones del compilador se utilizaron y cuál es la plataforma objetivo, la base de datos puede llegar a ser aproximadamente 35 % más rápida que las operaciones de escritura y lectura nativas de un sistema de archivos [41]. Así también es posible reducir el tamaño de la librería a menos de 600KiB. Lo que la hace ideal para ser utilizada en sistemas donde la memoria es muy limitada [42].

5.11. Experiencia de Usuario

5.11.1. Diseño centrado en las personas

Ritter *et al.* [43] recomienda que, cuando se diseña un sistema con interfaz de usuario, se debe utilizar un concepto llamado diseño centrado en las personas. La premisa central del diseño centrado en las personas es la siguiente:

Entender a las personas ayuda a construir mejores tecnologías y sistemas interactivos.

Esto es fundamental si se desea que el sistema sea tanto útil como usable. Si no se toma en cuenta cada una de las características de los usuarios, se corre un mayor riesgo que el sistema se lleve al fracaso, haciendo que los usuarios no acepten el sistema si no aporta ni se

¹*sniffing* es un tipo de ataque que consiste en el monitoreo y captura paquetes que pasan por una red, con el objetivo de espiar el contenido

acopla a la manera en que trabajan, que los usuarios sientan frustración mientras intentan que el sistema funcione como ellos desean, o que no sea adoptado por completo por los usuarios.

También se explica que “entender” a las personas consiste en lo siguiente:

- Observar y documentar qué es lo que las personas hacen: usar métodos apropiados para obtener resultados creíbles y así diferenciar simples anécdotas de información útil.
- Entender por qué las personas hacen lo que hacen: desarrollar *insights* acerca de las motivaciones de una persona por realizar una acción.
- Entender y predecir cuándo la gente es propensa a realizar cosas: entender los patrones del comportamiento de las personas.
- Entender cómo escogen las personas hacer las cosas que hacen: entender las opciones que las personas tienen o perciben que tienen, entender sobre las restricciones que tienen y guiar sobre qué opciones tienen realmente.

[43]

Se puede recalcar la importancia de seguir estos principios al ejemplificar un accidente aéreo ocurrido en el año 1989, en un vuelo de la aerolínea británica Kegworth. El accidente de la aerolínea Kegworth fue provocado tanto por fallos mecánicos como humanos, los últimos relacionados principalmente a usabilidad del sistema de vuelo para los pilotos y por falta de comunicación entre la tripulación. El análisis de este accidente provee los siguientes *insights*:

- Se debe entender cómo los usuarios extraen información de las interfaces de usuario para tomar decisiones y realizar acciones.
- Se debe entender cómo los usuarios crean modelos mentales para ayudarse a utilizar el sistema.
- Se debe entender cómo los usuarios aprenden a utilizar nuevos y, particularmente, más complejos sistemas.
- Se debe entender cómo características sociales, incluyendo la comunicación, pueden afectar cómo los usuarios utilizan un sistema.
- Se debe entender que el lugar y el tiempo en el que se utiliza el sistema es importante, y muchas veces distinto al ambiente de desarrollo.

Un error que suele cometerse al diseñar interfaces de usuario, es el error fundamental de atribución en el diseño. Este error consiste en asumir que los usuarios del sistema tienen las mismas capacidades y características del desarrollador. Los usuarios de un sistema usualmente no usan igual de bien el sistema como el desarrollador porque no lo conocen tan bien como él. Si se entiende al usuario que utilizará el sistema a construir, existen tres beneficios que se obtienen de esto:

- Productos más usables: Entender a las personas puede ayudar a diseñar sistemas que son más usables, con una curva de aprendizaje más baja y más eficientes.
- Ahorros financieros: crear interfaces de usuario que faciliten su uso puede ahorrar tiempo utilizado por personas dedicadas a brindar soporte.
- Sistemas más seguros: una interfaz clara hacia el usuario puede prever errores en el flujo de los programas, como el ingreso de datos erróneos o sensibles.

5.11.2. Framework ABCS

Existe también un *framework* o marco de trabajo que ayuda a organizar las características más relevantes del ser humano al diseñar sistemas usables, llamado ABCS por las siglas formadas por cuatro aspectos:

Antropometría

Esta área se interesa en los aspectos físicos, tanto del usuario como del sistema. Muchos de estos aspectos físicos son estudiados en el área de factores humanos y ergonómica, y se suele aplicar a equipo de oficina como escritorios y sillas. Específicamente para sistemas computacionales, estos problemas suelen estar relacionados a los controles del sistema (botones, *inputs*, etc.) y que su tamaño sea el correcto para poder ser manipulados por una amplia gama de usuarios.

Estos aspectos han cobrado mucho más interés en los últimos años, con el auge de los dispositivos móviles y *tablets* en el mercado, cuyos tamaños importan para diferentes escenarios de usabilidad. Las interfaces que corren en estos dispositivos incluyen tanto aspectos computacionales como su naturaleza física y las oportunidades y restricciones que estos conllevan.

Comportamiento(*Behavior*)

Cuando se habla de aspectos de comportamiento del usuario, se refiere a las respuestas básicas que un usuario puede producir ante distintos tipos de estímulos. Este factor se construye por encima del aspecto antropométrico, de manera que los aspectos físicos del cuerpo se usan para producir estos comportamientos simples.

Existe un aspecto de comportamiento muy utilizado en usabilidad, llamado “funciones forzadas”. Esto consiste en aspectos del producto o sistema que sugieren al usuario un uso o estilo particular en la interacción. Un ejemplo muy claro es el de las puertas: las puertas que tienen un manubrio sugieren que deben ser haladas para abrirse, mientras que una puerta sin manubrio sugiere que debe ser empujada [44].

Además de entender las bases de cómo es que los usuarios se comportan, también es importante el por qué se comportan de cierta manera. La motivación de una persona por

realizar una actividad puede variar de un individuo a otro, muchas veces por factores inherentes a la persona, pero también muchas veces por factores externos: si la actividad es parte de su trabajo o recreacional, si se realiza en una oficina o en un lugar público, etc.

Cognición

El nivel cognitivo utiliza los dos niveles y se construye a partir de ellos. En este nivel, se estudia cómo el usuario piensa acerca de la tarea que desea realizar y también acerca del sistema en el que la realiza, además de sus capacidades cognitivas básicas y avanzadas. Entre estas capacidades se incluye todos los sistemas de memoria que el usuario tiene a su disposición, además de la organización de esta memoria y cómo es usada por un procesador central. Las capacidades avanzadas incluyen cómo la atención y el aprendizaje afectan estos sistemas y procesos.

El trabajo realizado en este nivel suele incluir observaciones sobre la herramienta o el entorno que se está usando, preguntándose por qué se usa y cuándo se usa. Esto es particularmente necesario, porque las características de los usuarios varían más en este nivel que en los dos anteriores. Esta variación se da principalmente por el nivel de experiencia de los usuarios, que puede ser obtenido por medio de entrenamiento, educación formal, uso previo, estilo personal y elección de estrategias.

Un mejor entendimiento de cómo los usuarios piensan y sienten puede ser útil para crear mejores diseños en un sistema. Un sistema mejorado puede ser producto de entender el esfuerzo mental que los usuarios invierten en realizar una tarea, en términos de los mecanismos de procesamiento de información que hay en su pensamiento. Esto incluye, por ejemplo, la cantidad de símbolos que una persona es capaz de recordar. Tener esto en cuenta, ayuda a entender cómo dispositivos complejos con varias funcionalidades pueden ser más accesibles a usuarios no expertos, por medio de brindar información, guiar al usuario y al no pedir que ejecuten tareas complicadas.

Factores sociales (*Social Factors*)

El último nivel que toma en consideración el *framework* es el social. La principal pregunta que realizarse en este nivel es: ¿cómo interactúan los usuarios del sistema con otras personas mientras ejecutan sus tareas? En muchos casos, esta interacción consiste en que el usuario debe realizar la tarea en conjunto con otros usuarios para apoyar y mediar la comunicación. En otros casos, los usuarios usan el sistema como un servicio hacia otros usuarios, como por ejemplo los cajeros de bancos, empleados de call center, pilotos de aviones, etc.

Igual que los anteriores, este nivel se construye a partir del nivel anterior. Es particularmente importante el nivel cognitivo, incluyendo el modelo mental de los otros usuarios. La motivación de un usuario por trabajar en equipo con otros puede variar de un sistema a otro, por factores similares a los del nivel de comportamiento. Características organizacionales, profesionales, y culturales también se incluyen en este nivel. Algunos ejemplos de estos pueden ser la percepción de los colores según la cultura: el color verde no siempre significa “Aceptar” o “Confirmar”, por poner un ejemplo [43].

5.12. Pruebas de Usabilidad

5.12.1. Descripción general

Según el Departamento de Salud y Recursos Humanos del Gobierno de Estados Unidos [45], las pruebas de usabilidad se refieren a evaluar un producto realizando pruebas con los potenciales usuarios de este último.

Al probar la usabilidad de un producto o sitio web con un grupo objetivo de usuarios, los desarrolladores pueden determinar si los usuarios pueden usar el producto o sitio web de manera fácil e intuitiva. Los desarrolladores generalmente llevarán a cabo estudios de usabilidad en cada iteración del producto, desde su desarrollo inicial hasta su lanzamiento. Esto permite a los desarrolladores descubrir cualquier problema con la experiencia del usuario del producto, decidir cómo solucionar estos problemas y determinar si el producto es suficientemente utilizable.

Durante un estudio de usabilidad, el moderador pide a los participantes que completen una serie de tareas mientras el moderador o un ayudante observa y toma notas. Al observar a sus usuarios reales navegar por su producto o sitio web y escuchar sus opiniones y preocupaciones al respecto, se puede ver cuándo los participantes pueden completar tareas rápida y exitosamente y dónde encuentran problemas y experimentan confusión. Después de realizar su estudio, se analizan los resultados y se informa cualquier *insight* o descubrimiento.

Algunos beneficios que pueden obtenerse sobre realizar pruebas de usabilidad son los siguientes:

- Las pruebas de usabilidad proporcionan un examen imparcial, preciso y directo de la experiencia del usuario de un producto o sitio web. Al probar la usabilidad en una muestra de usuarios que están separados del equipo de desarrollo, los comentarios pueden resolver la mayoría de los debates que tiene el equipo de desarrollo.
- Las pruebas de usabilidad son convenientes. Para llevar a cabo un estudio, todo lo que se debe hacer es encontrar un lugar para realizarlo y tener algún medio para documentar (grabadora, cuaderno y lápiz, cámara, etc).
- Las pruebas de usabilidad pueden indicar qué hacen los usuarios en el sitio o producto y por qué toman estas decisiones, lo cual va de la mano con las capas del *framework* ABC explicado en la sección 4.2.2.
- Las pruebas de usabilidad permiten abordar los problemas del producto o sitio web antes de gastar recursos, como dinero o tiempo, creando algo que termina teniendo un diseño deficiente.
- En cuanto a aspectos de negocio, el diseño intuitivo aumenta el uso del cliente y sus resultados, impulsando la demanda del producto en cuestión.

[46]

Los pasos para ejecutar pruebas de usabilidad son variados: los principales autores proveen distintos marcos de trabajo y procesos para realizar estas pruebas, además que casi toda bibliografía indica que estos pueden variar dependiendo también del tipo de producto y de sus características. [47] propone un marco de trabajo general para realizar las pruebas, en el que se lleven a cabo cuatro fases principales, desglosadas en varias actividades. Esto se encuentra resumido en la Figura 14, y también se explica con mayor detalle a continuación.



Figura 14: Pasos para realizar pruebas de usabilidad

5.12.2. Crear un plan de pruebas

Alcances

Con definir los alcances, se refiere a decidir sobre qué áreas del producto se desea realizar las pruebas. Recomienda listar todas las áreas posibles e ir refinando la lista. Un número máximo de doce tareas es recomendable.

Reclutar usuarios

Reclutar usuarios es una actividad clave del paso inicial. El reclutamiento se puede realizar de acuerdo con la demografía (edad, lugar donde viven, etc.) o psicografía (antecedentes cognitivos: si cumplen con las aptitudes para desarrollarse en el escenario propuesto). Idealmente, se debe reclutar considerando ambos aspectos, pero se debe tener en cuenta que con la psicografía se puede obtener información más relevante sobre el uso y la adopción de su solución. En cuanto a la cantidad de usuarios a reclutar, un número recomendable es de que cinco usuarios por cada segmento o perfil de usuario [48].

Identificar objetivos

Esto se trata de identificar lo que se quiere lograr con esta prueba, lo que se está buscando y lo que se quiere demostrar a las partes interesadas. También pueden definirse objetivos para el usuario, que son los que definen las tareas que este deberá realizar en la evaluación y que reflejan el futuro uso del producto final.

Establecer métricas

Las métricas brindan una descripción común basada en hechos del desempeño de los usuarios sobre las tareas definidas, sobre las cuales se pueden tomar decisiones informadas de diseño. Esto es importante por los siguientes motivos:

1. Ayuda a hacer concretas las recomendaciones de usabilidad, influir y cambiar la opinión que se tenía sobre el producto.
2. Informa al equipo de desarrollo acerca de cómo es la realidad de los usuarios.
3. Ayuda a iterar el proceso de desarrollo y validar conceptos de diseño.
4. Provee objetividad para generar debates en torno a cambios a realizar.
5. Presenta una guía para la toma de decisiones basada en hechos.

Algunos ejemplos de métricas pueden ser: el tiempo en completar una tarea, el rendimiento de la tarea, cantidad de clicks para completarla, la tasa de éxito, la velocidad, el cumplimiento de objetivos, etc. Existen muchos tipos de métricas, cuáles se escojan dependerá de la naturaleza del proyecto.

5.12.3. Ejecutar las pruebas

Más que ejecutar las pruebas, se habla acerca de ser un "facilitador" para las mismas. Esto consiste en lo siguiente:

1. Pedir a los usuarios que externalicen sus pensamientos y sentimientos al interactuar con la solución.
2. Mantener el entorno de prueba lo más realista posible, no intentar minimizar las distracciones.
3. Documentar la sesión con vídeo/audio o, en su defecto, tomar notas, ya sea con una estructura o sin ella.
4. No guiar al usuario en los pasos que debe tomar.
5. No elaborar conclusiones durante la sesión de prueba.
6. Recordar siempre que no se trata de lo que se considera que es una buena experiencia de usuario, sino de cómo el usuario percibe la solución: escuchar, más que hablar.

[47]

5.12.4. Analizar la información

Una vez que se hayan terminado las sesiones de prueba, se recomienda sentarse, analizar la información y sacar conclusiones. No existe una regla común para hacer esto, pero cuando se tenga toda la información necesaria, se debe buscar las tendencias que surjan, tomar nota de los posibles problemas y las posibles soluciones.

5.12.5. Crear reporte de pruebas

Se debe crear un informe de prueba cada vez que se realice una prueba de usabilidad y se debe almacenar con cualquier otra documentación de prueba del producto. Se recomienda que incluya por lo menos las siguientes secciones:

1. Resumen de: breve resumen de lo que se probó, quién o quiénes realizaron la prueba, el material utilizado y una breve descripción de todos los hallazgos además de mencionar el objetivo de la sesión.
2. Metodología: explicar cómo se realizaron las sesiones, las tareas o escenarios que se probaron, las métricas que se seleccionaron y una breve descripción de cada segmento de usuarios.
3. Resultados de la prueba: resumir todos los resultados de las métricas que se eligieron. Un cuadro de Excel puede ser útil para este caso.
4. Hallazgos y recomendaciones: enumerar todos los hallazgos, tanto positivos como negativos. Los resultados positivos ayudarán al evaluador a saber si se está en el camino correcto. Para los resultados negativos, proporcionar propuestas para resolverlos.

5.13. Heurísticas

Los datos cuantitativos o métricas para las pruebas de usabilidad, pueden ser basados en heurísticas. Una heurística es un principio o regla que ayuda a guiar una decisión de diseño o a criticar una decisión de diseño ya realizada. Dix [49] recomienda utilizar el conjunto de heurísticas desarrolladas por Jakob Nielsen y Rolf Molich, el cual ha sido adoptado actualmente por la mayoría de los expertos en Interacción Humano-Computador como un estándar para este campo. Este acercamiento consiste en que el evaluador califica cada una de las heurísticas con un puntaje de 0 a 4 según la gravedad del problema observado (ver Cuadro 3).

Cuadro 3: Ponderación para las heurísticas

Calificación	Título	Descripción
0	Sin problemas	No existe un problema.
1	Problema estético	Corregir si el tiempo lo permite.
2	Problema menor de usabilidad	Baja prioridad de corrección.
3	Problema mayor	Corrección debe tener alta prioridad.
4	Catástrofe	Arreglar antes que salga a producción.

Otro concepto que se utilizó dentro de la investigación es el de prototipado horizontal. El prototipado horizontal consiste en realizar una simulación de la interfaz de usuario en la que la funcionalidad lógica no es requisito para la prueba. Esto significa que el usuario evaluado puede ejecutar todos los comandos de navegación y búsqueda, pero sin necesariamente recibir información real como respuesta [48].

Las heurísticas propuestas por Nielsen y Molich son las siguientes:

1. Visibilidad del estado del sistema: mantener a los usuarios siempre informados de qué sucede en el sistema, a través de respuestas en una cantidad razonable de tiempo.
2. Relación entre el sistema y el mundo real: el sistema debe hablar el mismo lenguaje que el usuario, con palabras, frases y conceptos que son familiares para él.
3. Control del usuario y libertad: el usuario escoge funciones por equivocación y escoge una opción debidamente identificada para regresar un paso atrás.
4. Consistencia y estándares: el usuario no debe dudar si palabras, situaciones o acciones significan lo mismo bajo distintos contextos.
5. Prevención de errores: hacer que sea difícil cometer errores para el usuario. Dar preferencia de prevención de errores sobre despliegue de mensajes de error.
6. Reconocimiento antes de memoria: hacer que los objetos, las acciones y las opciones sean visibles, para que el usuario no tenga que recordar información de un diálogo a otro.
7. Flexibilidad y eficiencia de uso: permitir al usuario realizar acciones frecuentes de manera rápida cuando este posee más experiencia en el sistema.
8. Diseño estético y minimalista: los diálogos no deben tener información irrelevante o innecesaria.
9. Ayudar al usuario a reconocer, diagnosticar y recuperarse de errores: los mensajes de error deben ser expresados en lenguaje común y deben indicar el problema y una propuesta de solución.
10. Ayuda y documentación: toda información necesaria para brindar ayuda y documentación debe ser fácil de buscar, debe estar enfocada en las tareas de los usuarios y debe listar pasos concretos.

[48]

5.14. Lenguaje *markdown*

Markdown es un lenguaje de marcado que permite agregar formato a documentos de texto simples. Un lenguaje de marcado es un lenguaje de computadora que permite definir elementos a partir de etiquetas, las cuales varían en su sintaxis según el lenguaje en cuestión. *Markdown* fue creado en el año 2004 por John Gruber, y es uno de los lenguajes de marcado más utilizados.

Utilizar *markdown* difiere significativamente de otros editores de texto como *Word*. En *Word*, por ejemplo, se presionan botones en una interfaz gráfica o atajos en el teclado para darle un formato a los elementos, y estos se actualizan inmediatamente. Por otro lado, en un documento de *markdown*, se agrega una sintaxis directamente al texto al cual se quiere agregar formato.

Algunas de los beneficios de utilizar *markdown* para crear documentos son las siguientes:

- *Markdown* tiene muchas aplicaciones útiles. Se utiliza para crear sitios en internet, documentos, libros, presentaciones, correos electrónicos y documentaciones técnicas.
- *Markdown* es portable. Los archivos que contienen texto con formato *Markdown* se pueden abrir utilizando varias aplicaciones existentes. Eso se encuentra en contraste con las aplicaciones como *Microsoft Word* que bloquean su contenido en un formato de archivo propietario.
- *Markdown* es independiente de la plataforma o sistema operativo. Se puede crear texto con formato *Markdown* en cualquier dispositivo que ejecute cualquier sistema operativo.
- *Markdown* está en todas partes. Los sitios de internet como *Reddit* y *GitHub* admiten *Markdown*, y muchas aplicaciones de escritorio y de internet lo admiten.

[50] Algunos elementos básicos de la sintaxis *markdown* son descritos en el Cuadro 4.

[50]

5.15. Estadística

Durante el análisis de resultados del proyecto, se utilizaron las siguientes medidas numéricas de estadística descriptiva para obtener datos cuantitativos sobre las pruebas de usabilidad.

5.15.1. Media

La media, o valor medio, es considerada por Anderson et al [51] como la medida de ubicación más importante para una variable numérica, puesto que proporciona una medida

Cuadro 4: Sintaxis básica de *Markdown*

Elemento	Sintaxis
Encabezados	# Título 1 ## Título 2 ### Título 3
Negrita	**texto en negrita**
Cursiva	*texto en cursiva*
Lista ordenada	1. Primer elemento 2. Segundo elemento 3. Tercer elemento
Lista sin orden	- Primer elemento - Segundo elemento - Tercer elemento
Línea horizontal	—
Hipervínculo	[título](https://www.ejemplo.com)
Imagen	[texto alternativo](ejemplo.jpg)

de la ubicación central para los datos. Si los datos son para una muestra, la media representa por la letra \bar{x} ; si son para una población, se denota por la letra griega μ .

$$\bar{x} = \frac{\sum x_i}{n} \quad (1)$$

La ecuación 1 expresa cómo calcular la media de una muestra, donde n es la cantidad de datos usados para el cálculo de la medida y $\sum x_i$ es la suma de todos los datos.

[51]

5.15.2. Varianza

La varianza es una medida que toma en cuenta todos los datos para representar la variabilidad de los mismos. Se basa en la diferencia entre el valor de los datos individuales $\sum x_i$ y la media \bar{x} . La varianza de una muestra se denota por s^2 , y es descrita por la ecuación 2 donde, al igual que en la media, n es la cantidad de datos usados para el cálculo de la medida y $\sum (x_i - \bar{x})^2$ representa la suma de las diferencias entre cada dato y la media, elevando cada diferencia al cuadrado.

$$s^2 = \frac{\sum (x_i - \bar{x})^2}{n - 1} \quad (2)$$

[51]

5.15.3. Desviación Estándar

La desviación estándar es la raíz cuadrada positiva de la varianza. Basándose en la notación para la varianza, la desviación estándar se denota como s si es para una muestra (ver ecuación 3)

$$s = \sqrt{s^2} \quad (3)$$

La relevancia de la desviación estándar como medida numérica yace en el hecho de que las unidades de la varianza se encuentran elevadas al cuadrado, como puede verse en el numerador de la fracción dentro de la ecuación 2. Como la desviación estándar es la raíz cuadrada de la varianza, las unidades cuadradas de la varianza se convierten en las mismas unidades de los valores estudiados. Por esta razón, la desviación estándar es más fácil de comparar con la media y con otros estadísticos que miden las mismas unidades que los datos originales.

[51]

5.16. Red

5.16.1. Modelo OSI

La Organización Internacional para la Normalización (ISO) realizó investigaciones acerca de los esquemas de red. La ISO lanzó en 1984 un esquema descriptivo, el cual proporciona a los fabricantes un estándar que asegura la compatibilidad entre distintos tipos de tecnología de red a nivel mundial.

Este modelo cuenta con siete capas:

1. Física: Transmisión binaria
2. Enlace de datos: Acceso a los medios
3. RED: Direccionamiento y mejor ruta
4. Transporte: Conexiones de extremo a extremo
5. Sesión: Comunicación entre *Hosts*
6. Presentación: Reprerisión de datos
7. Aplicación: Proceso de red a aplicaciones

Este modelo se utiliza principal mente para las comunicaciones de red, permitiendo a los usuarios ver las funciones de red que se producen en cada capa. [52]

5.16.2. Red en malla

Una “*mesh network*” o red en malla, es un conjunto de nodos conectados entre sí, cuyo propósito es establecer una conexión robusta y estable entre los nodos, a un costo reducido. Una red en malla consiste en al menos dos nodos que se conecten entre sí. Los usuarios acceden a la red a través de los “*mesh routers*”, que son los nodos secundarios, y al mismo tiempo se conectan con los otros usuarios de la red como se muestra en la Figura 3.

Si se desea integrar una conexión a Internet, como mínimo se debe establecer conexión a un nodo de la red, y así este puede proveer conexión a los demás nodos.

A diferencia de una red inalámbrica tradicional, una red *mesh* es auto-organizada y auto-configurada. Esto implica que los nodos de la red, establecen de forma automática la configuración y la conectividad. Esto tiene como beneficio un bajo costo de instalación, fácil mantenimiento de red, robustez y mayor cobertura. Permite la integración con otras redes inalámbricas como Wi-Fi.

Entre las principales características se encuentran:

- Mayor confiabilidad: provee caminos redundantes entre receptor y emisor, eliminando fallas por un punto particular. Esto significa que, si un nodo falla, la red encuentra otro camino de establecer conexión
- Bajo costo de instalación en comparación a una red que necesita de cableado de cobre para el transporte de datos. Para establecer conexión a internet en un área mediana mediante el uso tradicional de *routers*, se requiere instalar varios puntos de acceso, cada uno conectado mediante una conexión física (cable) a internet. Mientras que una red *mesh*, sólo ciertos puntos de acceso requieren estar físicamente conectados, y los demás se conectan de forma inalámbrica, reduciendo costos de infraestructura.
- Fácil escalabilidad: Esta red puede agregar nodos de forma automática siempre y cuando compartan la misma configuración

Como mencionado anteriormente, al menos un nodo principal debe estar conectado a internet (“*Mesh Gateway*”). Los *mesh routers* se conectan al *gateway* y luego los clientes a los *routers*. De esta forma, los clientes tienen acceso a internet y al mismo tiempo están conectados entre sí. [53]

5.16.3. Red Comunitaria

La definición establecida por la Declaración de la primera Cumbre Latinoamericana de Redes Comunitarias, llevada a cabo en septiembre de 2018, es:

Las redes comunitarias son redes de propiedad y gestión colectiva de la comunidad, sin finalidad de lucro y con fines comunitarios; se constituye como colectivos, comunidades indígenas u organizaciones de la sociedad civil sin fines de lucro, que ejercen su derecho a la comunicación, bajo principios de participación democrática de sus miembros, equidad, igualdad de género, diversidad y pluralidad.



Figura 15: Representación de estructura de una red en malla

Las redes comunitarias están estructuradas para ser abiertas, libres y respetar la neutralidad de la red. Estas redes dependen de la participación activa de las comunidades locales en la concepción, desarrollo, implementación y gestión de infraestructura compartida como un recurso común, propiedad de la comunidad y operadas de forma democrática. [54]

5.16.4. Red tipo ad hoc

Una red tipo ad hoc, consiste en un conjunto de nodos móviles, los cuales se configuran de forma automática. La cual permite comunicación por toda la red de forma inalámbrica, ya que cada nodo esta designado a servir como un repetidor, obteniendo una red descentralizada. Gracias a que esta red no necesita una infraestructura y se puede organizar de forma automática, esta puede ser desplegada de forma rápida para proveer una comunicación robusta en situaciones hostiles. Lo que lo hace apropiada para situaciones de uso militar. Este tipo de red cuenta con limitaciones muy marcadas, dentro de ellas, la calidad de señal emitida ya que esta se degrada rápidamente en factor a la distancia, por lo cual el área efectiva de transmisión es limitada. [55]

5.17. Tecnologías para arquitectura de redes

5.17.1. Firmware

El *firmware* es un programa de software grabado en un dispositivo de hardware, como teclados, disco duro, BIOS o tarjetas de vídeo. Está programado para dar instrucciones permanentes para comunicarse con otros dispositivos y realizar funciones como tareas básicas de entrada / salida. El *firmware* generalmente se almacena en la ROM flash (memoria de solo lectura) de un dispositivo de hardware. Se puede borrar y reescribir.

El *firmware* se diseñó originalmente para software de alto nivel y se podía cambiar sin tener que cambiar el hardware por un dispositivo más nuevo. El *firmware* también conserva las instrucciones básicas para los dispositivos de hardware que los hacen operativos. Sin *firmware*, un dispositivo de hardware no funcionaría [56].

Dentro de los *firmwares* mas utilizados para redes de tipo libre *Mesh* podemos encontrar tres:

- DD-WRT

DD-WRT es un *firmware* de formato libre basado en Linux, diseñado como una alternativa para reemplazar el *firmware* que viene de fabrica en modelos específicos de routers, esta esta desarrollado por *Sveasoft's software decision*, y es basado en el *firmware* de Linksys. Este firmware facilita la eliminación de las restricciones que el *firmware* de fabrica genera, por lo cual provee las habilidades de controlar las características de la red. [57]

Plataforma: Linux

Fortalezas:

- Soporte a muchos *routers*
- Soporte integrado de *OpenVPN*
- Soporte de QoS
- Variedad de opciones y herramientas

Debilidades:

- Difícil de encontrar versiones nuevas para algunos *routers*
- Necesita de conocimiento moderado en redes

- Tomato

Tomato es un firmware relativamente nuevo, de formato libre el cual trae incluido herramientas de interfaz gráfica, monitorio de trafico entre otros, esta basado en Linux, fue creado por Jonathan Zarate en 2006, basándose en HyperWRT, fue publicado en junio de 2010. Este firmware igual que los otros permite la instalación de paquetes de herramientas para la configuración de la red. [58]

Plataforma: Linux

Fortalezas:

- Interfaz moderna
- Altas velocidades
- Monitorio en tiempo real incluido
- Fácil de operar
- Soporte integrado de *OpenVPN*

Debilidades:

- Comunidad pequeña
- Soporte limitado
- Relativamente nuevo
- No cuenta con un catalogo grande de *routers* soportados

- OpenWRT

OpenWRT a es un firmware de formato libre el cual esta construido desde 0, no tiene base de ningún otro firmware a diferencia de DD-WRT, lo cual lo hace altamente modificable. Este firmware provee la libertad de gestionar paquetes opcionales y desarrollados por la comunidad, lo que facilita el desarrollo de proyectos basados en redes. [59]

Plataforma: Linux

Fortalezas:

- Multiples Opciones y Herramientas
- Soporte integrado de *OpenVPN*
- Soporte de QoS
- Habilidad de opciones a bajo nivel

Debilidades:

- No es amigable con el usuario
- Requiere de conocimiento avanzado en redes
- Soporte limitado de *routers*
- Requiere mas tiempo para preparar

5.17.2. Protocolos de enrutamiento

- OLSR

Thomas Clausen y Philippe Jacquet son los creadores de hipercom INRIA. Este es un protocolo proactivo basado en la optimización de los protocolos link-state. OLSR es un protocolo, en el cual periódicamente se envían mensajes para mantener las tablas de enrutamiento actualizadas. OLSR funciona perfectamente en redes con un alto número de usuarios y con una topología cambiante. De igual manera, para obtener un control, se intercambian mensajes periódicamente los cuales aprenden la topología de la red y el estado de los nodos vecinos. En este protocolo, los nodos reconocen la manera más eficiente de encaminar los paquetes, esto se debe a que estos, ya conocen la topología de la red y toma en cuenta la calidad de los enlaces. El inconveniente de crear una carga de tráfico en a red y una carga de procesamiento extra en los nodos se añade al crear las tablas de encaminamiento con el algoritmo de Dijkstra u *Open Path First*

En este protocolo, los nodos conocen la mejor manera de encaminar los paquetes ya que conocen completamente la topologia de la red y tiene en cuenta la calidad de los

enlaces, pero se añade el inconveniente de crear una carga de tráfico en la red y una carga de procesamiento extra en los nodos al crear las tablas de encaminamiento con el algoritmo de Dijkstra u *Open Path First* [60]

- B.A.T.M.A.N

B.A.T.M.A.N es un protocolo diseñado por Freifuck, una comunidad *wireless* sin fines de lucro, en Alemania principalmente. [61]

Este fue diseñado teniendo en cuenta algunas deficiencias del protocolo OLSR, como alto consumo de recursos; B.A.T.M.A.N es un protocolo de enrutamiento en donde ningún nodo tiene la información completa de la topología de la red.

Por el contrario, para construir su tabla de enrutamiento, este, almacena el enlace de un salto, el cual conecta un nodo con el otro, al cual se le denomina dirección. para cada enlace de un salto, se establecen paquetes que llegan a cada nodo. Y así genera mensajes en *broadcast* llamados *Originator Messages* (OGM) para informar a los nodos vecinos sobre su existencia. [62]

Este protocolo, mantiene la red informada de la existencia de los nodos vecinos, pero esta no tendrá la información topológica de la ruta para cada nodo.

Este protocolo determina el mejor enlace entre nodos, haciendo comparación de la cantidad de secuencias recibidas no repetidas de cada nodo, si este es mayor, sera el mejor enlace. De esta manera se comparte la topología de la red y cada nodo, almacena la mejor dirección para el destinatario. [63]

- Babel

Babel es un protocolo desarrollado por Juliusz Chroboczek, profesor de la Unidad de Formación e Investigación de la Universidad Diderot de París. Esta basado en el algoritmo de vector de distancias y diseñado para ser robusto y eficiente.

Utiliza una técnica basada en *Distance Vector*. Funciona con IPv4 e IPv6 y fue diseñado, en un principio, para redes inalámbricas, por lo que es un protocolo muy robusto en presencia de nodos móviles, impidiendo la formación de bucles sin fin y ofreciendo una rápida convergencia.[63]

Cuadro 5: Comparación entre los distintos protocolos de enrutamiento

Protocolo	Tipo de Protocolo	Alcance de Transmisiones	Métrica de ruteo	Uso Libre/Propietario
OLSR	Proactivo	Unicast	Camino mas corto	Libre
B.A.T.M.A.N	Reactivo	Multicast	Próximo mejor salto para cada destino	Libre
Babel	Reactivo	Multicast	Configurable. Por defecto: Calidad de enlace	Libre

Performance Evaluation of Mesh Protocols in Real Time Mesh Test Bed

Este documento tiene como objetivo el diseño de una malla en tiempo real, para comparar diferentes protocolos de enrutamiento y realizar pruebas detalladas sobre los mismos. Las métricas utilizadas incluyen la evaluación del rendimiento, la utilización del ancho de banda, relación de entrega de paquetes y latencia.[64]

Evaluation of mesh routing protocols for wireless community networks

En este artículo se evalúan los mecanismos de autogestión y escalabilidad de tres protocolos de enrutamiento para redes tipo maya, a través de experimentación por medio de un banco de pruebas en entornos controlados. Dentro de las métricas a evaluar se encuentran:

- Eficiencia de la red
- Adaptación de la red a cambios en la infraestructura
- Métricas de escalabilidad

[65]

Performance Evaluation and Optimization of B.A.T.M.A.N. V Routing for Aerial and Ground-based Mobile Ad-hoc Networks

En este artículo se evalúa el desempeño del protocolo de enrutamiento B.A.T.M.A.N 5 para la integración de una red tipo malla, exponiendo a una red de comunicación altamente dinámica. Este artículo presenta un modelo de simulación de código abierto, para realizar la evaluación. [62]

An Experimental Comparison of Routing Protocols in Multi Hop Ad Hoc Networks

En este artículo se presenta el rendimiento de tres protocolos de enrutamiento, basado en múltiples saltos dentro de una red tipo ad hoc, mostrando el alcance, de los siguientes protocolos: OLSR, BABEL, y B.A.T.M.A.N. [66]

5.17.3. Plan Estratégico

En esta sección se redacta la planificación que se realizará dentro de un periodo de tiempo, la estructura se debe construir con la participación de las personas que forman parte de la organización. Los elementos a incluir deben ser :

1. Historia de la organización.
2. Diagnostico de la organización.
3. Visión de la organización.
4. Principios y valores de la organización.
5. Rol.

Gestión de proyectos

Gestionar un proyecto se refiere a todos los tramites necesarios para que un proyecto tenga los recursos humanos, materiales y financieros necesarios para que este se lleve a cabo.

Funciones de los COCODE's Relacionados con proyectos

Los COCODES, están integrados por miembros de la comunidad elegidos de forma democrática. Su función principal es detectar las necesidades principales de la comunidad a la cual pertenecen. Su cometido consiste encontrar propuestas que solucionen dichas necesidades, que sean viables. El COCODE debe de estar familiarizado con "El Ciclo del Proyecto" lo cual significa que deben conocer los cuatro pasos básicos para poder proceder a analizar el problema, estos pasos son:

1. Identificación del problema.
2. Formulación de políticas, planes programas y proyectos de desarrollo de la comunidad.
3. Dar seguimiento a la ejecución de las políticas, planes, programas y proyectos de desarrollo comunitarios.
4. Evaluar la ejecución y eficacia de los programas y proyectos comunitarios.

Luego de localizados y ejecutados los cuatro puntos anteriores el COCODE debe informar a la comunidad sobre el desarrollo, progreso y finalización del proyecto.

Formulación, asignación y ejecución de recursos para las comunidades

El COCODE tiene el deber de priorizar las necesidades de la comunidad a partir de diagnósticos, para plantear soluciones y gestionar recursos y soluciones para los mismos.

Perfil de proyectos diagnósticos

El perfil del proyecto lo determina la comunidad a través del diagnostico del problema, atreves de asamblea puede plantear soluciones y luego la comunidad decide que necesidad tiene mayor urgencia.

5.17.4. Marco jurídico

A partir de 1985 se determina que las políticas públicas lleguen a la población en general, con el objetivo de satisfacer necesidades y servicios básicos. Para ello el estado de la República de Guatemala establece que un 10% de los ingresos del estado debe ser transferido a los municipios, destinados en su mayoría a proyectos de educación, salud, obras de infraestructura y servicios públicos que mejoren la condición de vida de la población.

Para poder diagnosticar las necesidades de la población y sus posibles soluciones debe de seguirse una guía jurídica que a continuación detallamos:

1. Identificación del proyecto.
 - Nombre del proyecto.
 - Análisis de la problemática.
 - Situación sin proyecto.
 - Situación con proyecto.
2. Beneficios directos e indirectos.
3. Objetivos y metas.
4. Estudio de mercado.
5. Estudio técnico.
6. Ingeniería del proyecto.
7. Estudio financiero.
8. Seguimiento de proyectos.

5.17.5. Proceso para la gestión de proyectos ante los Consejos de Desarrollo

Para poder presentar un proyecto ante el Concejo de Desarrollo es necesario seguir los siguientes pasos:

1. Densificación del proyecto.
2. Descripción de la organización o comité.
3. Antecedentes del proyecto.
4. Justificación del proyecto.
5. Diagnóstico y caracterización del área de influencia.
6. Aspectos geográficos.
7. Aspectos económicos.
 - Población.
 - Beneficiarios directos e indirectos.
 - Aspectos demográficos.
 - Características económicas.
8. Aspectos económicos cualitativos.

- Educación.
 - Salud.
 - Situación ambiental.
 - Servicios públicos existentes.
 - Medios de comunicación.
 - Cultura.
 - Descripción del proyecto.
9. Metas del proyecto: Consiste en dar a conocer los beneficios tanto sociales como económicos que la población va a obtener con el proyecto.
 10. Beneficiarios del proyecto: Se refiere directamente a la población a la cual va dirigido el proyecto.
 11. Cronograma de actividades: Consiste en detallar todas las actividades que se harán en el proyecto y los periodos de tiempo que estas duraran.
 12. Costo y financiamiento del proyecto: Consiste en detallar el valor del proyecto.
 13. Impacto del proyecto: Consiste en los logros que el proyecto impactara en la comunidad.
 14. Evaluación del proyecto: Consiste en evaluar los logros y obstáculos que puede tener el proyecto ya elaborado.

Elaborados estos pasos es necesario presentar los proyectos a la DTP y SEGEPLAN, los cuales son las entidades encargadas de elaborar presupuestos y definir recursos para alcanzar los objetivos propuestos.

5.18. Redes WLAN

Una red WLAN, por sus siglas en inglés *Wireless Local Area Network*, es un tipo de red que implementa un sistema de comunicación de datos que es utilizada para reemplazar las redes LAN cableadas. Este tipo de redes normalmente son implementadas en edificios o recintos universitarios. Este tipo de redes usa radiofrecuencias para transmitir y recibir datos utilizando el aire como medio, minimizando la necesidad de conexiones cableadas.[67]

Las WLAN se encuadran dentro de los estándares desarrollados por la IEEE. Un aspecto a destacar de las redes WLAN es su integración en entornos de redes móviles 3G. Las WLAN constituyen en la actualidad una solución tecnológica de gran interés en el sector de las comunicaciones inalámbricas. Estos sistemas se caracterizan por trabajar en bandas de frecuencia que están exentas de una licencia de operación. Sin embargo, esto obliga el desarrollo de un marco regulatorio que permita un uso eficiente y compartido dentro del espectro radioeléctrico disponible de dominio público. [68]

Las características de más se destacan son:

- **Movilidad:** Permite transmitir información en tiempo real en cualquier lugar de la organización o empresa del usuario. Esto supone mayor productividad y posibilidades de servicio.
- **Facilidad de instalación:** Al no usar cables, se evita tener cables encima de muros y techos, lo que mejora el aspecto y habitabilidad de las localidades y reduciendo el tiempo de instalación. También, permite el acceso a la red a usuarios temporales.
- **Flexibilidad:** Las redes WLAN pueden llegar a donde redes cableadas no pueden. Es muy útil en zonas en donde el cableado no es posible o es muy costoso.

5.19. Punto de acceso

Un punto de acceso, en inglés *access point* o AP, es un dispositivo que es utilizado para crear una red de área local o WLAN. Usualmente son utilizados en oficinas o edificios. Un punto de acceso se conecta a un Router, Switch o Hub y proyecta una señal de *Wi-Fi* en un área designada. [69]

Cuando se tienen muchos clientes con muchos dispositivos queriéndose conectar a la red inalámbrica un punto de acceso lo permite fácilmente. Un punto de acceso da la libertad de poder escalar el número de dispositivos que nuestra red soporta. Algunas de las ventajas del uso de puntos de acceso son:

- Añadir más puntos de acceso a la red es fácil.
- Algunos puntos de acceso utilizan el mismo cable Ethernet con el que están conectados a un router para alimentarse de energía eléctrica por lo que no se requiere de instalación extra de una línea eléctrica.
- La mayoría de puntos de acceso permiten el uso de portales captivos y de un ACL (*Access Control List*), para restringir el acceso a la red.

5.20. Punto de acceso no autorizado

Una de las principales preocupaciones que los profesionales de seguridad tienen en cuanto a WLAN son los puntos de acceso no autorizados o *Rogue AP* en inglés. Un *Rogue AP* o se refiere a un AP que está conectado a una LAN o WLAN sin autorización, o a un AP que no está conectado a la red pero que actúa como si lo estuviera, aceptando solicitudes de clientes que desean utilizar la red. Un *Rogue AP* incluso puede tratarse de un AP que posee un adaptador inalámbrico y de software especializado que le permite actuar como un AP.

Los *Rogue AP* no autorizados que están conectados a una red son una preocupación en seguridad porque puede que no estén aseguradas acorde a las políticas de seguridad de una organización, lo que se traduce en nuevas vulnerabilidades hacia la red. Los *Rogue AP* que no están conectados a la red pueden aceptar peticiones por parte de clientes lo que les permite denegar el acceso de los clientes. Estos dispositivos se pueden dividir en:

- Maliciosos
- No-maliciosos

La prevención y detección de los *Rogue AP* normalmente se realiza mediante la implementación de sistemas de prevención de intrusión que monitorea el espectro de radio en búsqueda de estos dispositivos.

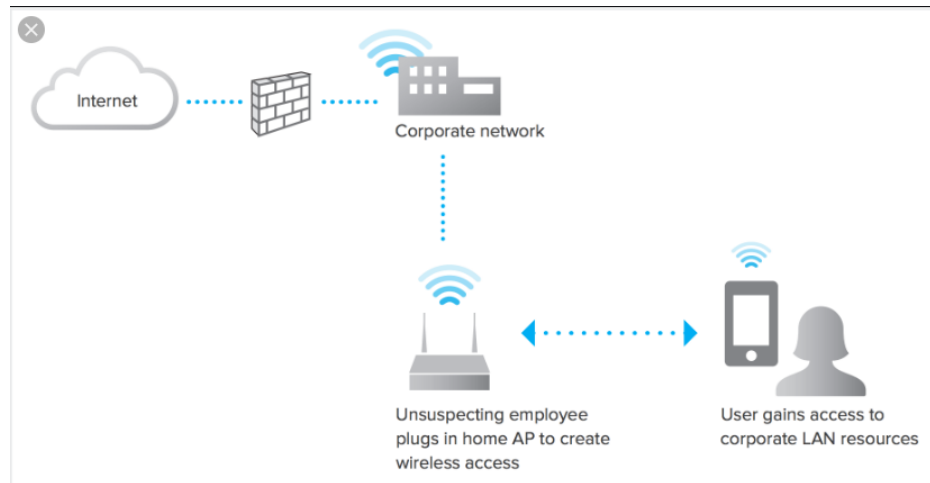


Figura 16: Ejemplo de compromiso accidental de LAN cableada en un entorno corporativo mediante el uso de un punto de acceso no autorizado. [70]

5.21. Portal captivo

Un portal captivo es una pantalla de autenticación que es desplegada cuando un usuario con un dispositivo inalámbrico no está autorizado para acceder a los recursos de la red. Esta página de autenticación es conocida como portal captivo.

Un portal captivo puede ser activado en el dispositivo de un cliente de dos maneras:

1. Redireccionamiento por DNS
2. *Splash page*

El redireccionamiento por DNS funciona al simplemente redireccionar todas las peticiones de DNS de un usuario y resolverlas hacia la pantalla de inicio de sesión del portal captivo. Lamentablemente, con este método se detectó que posee una tasa de éxito muy baja.

Por otro lado una *Splash page* utiliza un mecanismo un poco diferente. Esta alternativa también hace uso del redireccionamiento por DNS, pero responde a las solicitudes acc. a los sistemas operativos que engañan al sistema operativo al creer que existe un inicio de sesión por parte del portal captivo y forzar que el sistema operativo automáticamente despliegue

la página de inicio de sesión al usuario. Esto lo que significa es que si el dispositivo de un cliente se conecta a la red *Wi-Fi*, automáticamente se le desplegará la pantalla del portal captivo.

La única diferencia entre un portal captivo por redireccionamiento por DNS y un *splash page* es que este último despliega el portal captivo, mientras que el redireccionamiento DNS necesita que se haga primero una petición para desplegar el portal captivo. El problema con el redireccionamiento DNS radica en que si, por ejemplo, un usuario está haciendo uso de la red mediante una aplicación móvil no habría forma de que el usuario sepa si debe iniciar sesión.[71]

5.22. Disponibilidad de redes en TI

Alta disponibilidad es la capacidad de mantener operativo los servicios y operaciones de una red, eliminando los problemas comunes de los sistemas de información. En estos sistemas la información se encuentra consolidada y en caso de pérdida de los datos o problemas en el servicio, otra máquina entraría a reemplazarla y tomar el rol de la máquina principal. [72]

La alta disponibilidad se divide en dos tipos:

- Alta disponibilidad en *hardware*

Esto se traduce en redundancia de *hardware*. Si se produce un fallo de hardware en alguna de las máquinas, la configuración del equipo debería permitir cambiar el hardware en el instante sin tener que darle de baja a los servicios. A este proceso se le llama *HotSwap*.

- Alta disponibilidad en software

Para que exista la alta disponibilidad en software, en caso se produzca un fallo por parte de las aplicaciones, el software de alta disponibilidad es capaz de re iniciar automáticamente los servicios que han fallado. Al momento en el que la máquina en la que hubo fallos se recupera, las aplicaciones pueden migrar a la máquina original.

La alta disponibilidad es un tema de debate en los departamentos de TI debido a que balancear una alta disponibilidad contra los costos que implementarla representa es algo muy difícil de alcanzar.

5.23. Servidor web

Un servidor web puede hacer referencia a *hardware* o *software*, o ambos trabajando en conjunto.

En términos de *hardware*, un servidor web es una computadora que contiene el *software* de un servidor web y los componentes de un sitio web (como por ejemplo páginas HTML,

imágenes, hojas de estilo CSS y archivos *JavaScript*). Está conectado a una red de internet y soporta el intercambio de datos con otros dispositivos conectados a la misma red.

En términos de *software*, un servidor web incluye varias partes que controlan como los usuarios de una red acceden a archivos alojados, esto se ve comúnmente como un servidor HTTP como mínimo. Un servidor HTTP es un *software* que entiende dirección web como las URL y el protocolo HTTP. Este puede ser accesado mediante el uso de dominios (por ejemplo, google.com) de los sitios web que almacena, y entrega el contenido en el dispositivo del usuario final.

Normalmente, cuando un navegador solicita un archivo que está alojado en un servidor web, el navegador realiza esa solicitud mediante el protocolo HTTP. Cuando una solicitud alcanza el servidor web correcto, el servidor HTTP se encarga de aceptar la respuesta, encontrar el documento solicitado y devolverlo al navegador. Si el documento no existe o no puede encontrarlo el servidor devuelve una respuesta 404. [73]

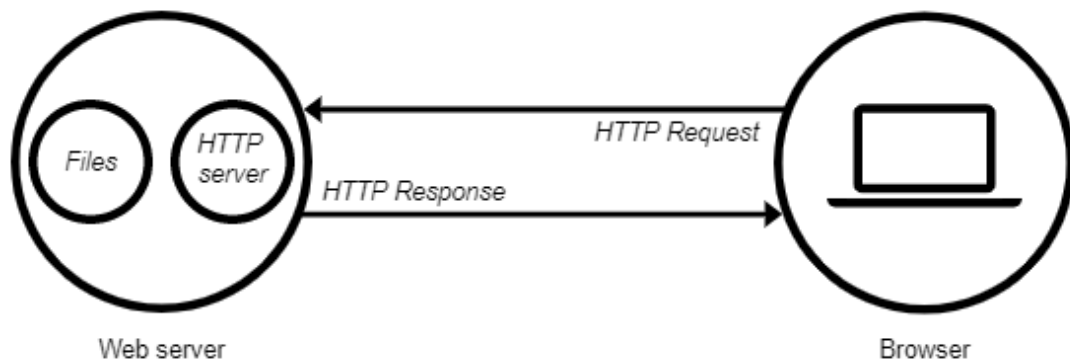


Figura 17: Comunicación entre un servidor web y un navegador. [73]

Los servidores web se dividen en dos tipos:

- Servidor estático: Se le denomina estático a un servidor que entrega al navegador el contenido tal y como es.
- Servidor dinámico: Se le denomina servidor dinámico a un servidor que entrega al navegador contenido que es cargado dinámicamente, es decir, actualizado mediante aplicaciones o llamadas a bases de datos.

5.24. NGINX

NGINX es un *software* de código abierto utilizado como servidor web, *reverse proxy*, balanceador de carga, trasmisor multimedia entre otros. Es un servidor web que está diseñado para el rendimiento y estabilidad máxima. Además de su funcionamiento como servidor HTTP, NGINX también funciona como servidor proxy para correos electrónicos y como *reverse proxy* y balanceador de carga para servidores HTTP, TCP y UDP.

NGINX fue creado por Igor Sysoev para resolver el problema C10K, el cual describía la dificultad para los servidores web de la época para manejar grandes cantidades de conexiones concurrentes. Debido a su diseño orientado a eventos y un arquitectura asincrona, NGINX revolucionó como los servidores operan en situación de necesidad de un alto rendimiento. Después de volver el proyecto en un proyecto de código abierto, el uso de NGINX creció exponencialmente. Luego a esto, Sysoev fundó NGINX, Inc. para darle soporte a un continuo desarrollo de NGINX y para ofrecer NGINX plus, el cual es un producto comercial que incluye características adicionales para los clientes empresariales. [74]

Algunas de las características más utilizadas de NGINX son:

- **Balancedador de carga:** NGINX hace que los balanceadores por medio de hardware luzcan obsoletos. Como un balanceador de carga NGINX es altamente configurable y está diseñado para manejar modernas arquitecturas en la nube. Además, ofrece integración con modernas herramientas para DevOps.
- *Reverse Proxy:* Mediante el uso de una arquitectura basada en escalabilidad, NGINX también es utilizado como *reverse proxy* para el manejo de tráfico entrante y distribuirlo a otros servidores.
- **NGINX WAF:** Debido a que es difícil crear aplicaciones y sistemas seguros, NGINX ofrece NGINX WAF. Esta característica consiste en un *Web Application Firewall* que protege el servidor web en contra de ataques sofisticados a nivel de aplicación. NGINX WAF se basa en un *software* de código abierto altamente conocido llamado ModSecurity. [75]

5.25. Protocolo de Seguridad

Un protocolo de seguridad es un protocolo que normalmente se ejecuta en un ambiente de poca confianza y que cumple con un objetivo de seguridad. Los protocolos de seguridad de red son un tipo de protocolo que garantiza la seguridad y la integridad de los datos en tránsito a través de una conexión de red. Los protocolos de seguridad de red definen los procesos y la metodología para asegurar los datos de red de cualquier intento ilegítimo de ver o extraer el contenido de los datos. [76]

5.26. Protocolos de cifrado *Wi-Fi*

Un protocolo de cifrado *Wi-Fi* consiste en un protocolo de seguridad para redes WLAN que implementa algoritmos de cifrado para el cifrado de la comunicación entre los dispositivos de la red. El uso de un protocolo de cifrado permite aumentar el nivel de privacidad de la información de un usuario dentro de la red y permite impedir que terceros puedan tener acceso a esta información. [77]

A lo largo de los años se han creado distintos protocolos de cifrado y distintas variantes para cada uno de ellos. Sin embargo, dichos protocolos pueden ser segmentados en tres grupos, los cuales se describen a continuación:

1. WEP

Es el primer protocolo de seguridad que fue implementado bajo el estándar de la IEEE 802.11, para el cifrado de datos a la hora de su transmisión. El objetivo de este protocolo es proporcionar un método de autenticación y confidencialidad en redes WLAN. Con el pasar de los años este protocolo entró bastante en polémica debido a que los analistas criptográficos encontraron en él varias vulnerabilidades, siendo hoy en día el protocolo de cifrado *Wi-Fi* más vulnerable y débil. Como consecuencia el protocolo WEP fue reemplazado por el protocolo WPA. [78]

Tipos de protocolo de cifrado WEP:

- WEP de 64 bits
- WEP de 128 bits

2. WPA

Wi-Fi Protected Access (WPA) es un protocolo de seguridad propuesto como reemplazo para el protocolo WEP. Al momento de propuesta, su propósito era mitigar las debilidades y vulnerabilidades del protocolo WEP. Al igual que dicho protocolo, el protocolo WPA está basado sobre el estándar de la IEEE 802.11. Este protocolo realiza la autenticación de usuarios mediante el uso de un servidor en donde se almacenan las credenciales y contraseñas de los usuarios en la red, aunque al igual que WEP puede utilizarse con una clave predefinida. A pesar de corregir las vulnerabilidades introducidas en WEP la mayor generación de inconvenientes para este protocolo fueron temas de retrocompatibilidad con hardware antiguo. Por lo tanto, la cobertura y campo de aplicación para este protocolo se vió reducida. Adicionalmente, alrededor del año 2008 se demostró que ataques a las redes con WPA eran factibles haciendo uso de parte de otros famosos ataques WEP como “Chopchop” [79]

Tipos de protocolo de cifrado WPA:

- WPA -PSK-KIP
- WPA -PSK-AES
- WPA -MGT-TKIP
- WPA -MGT-AES

3. WPA2

Los mismos creadores del protocolo WPA, debido al éxito de dicho protocolo y la popularidad del término, la alianza Wi-Fi decidió renombrar el protocolo 802.11i bajo el nombre de WPA2. Dicho protocolo mitiga las vulnerabilidades que tanto WEP como WPA presentaban por lo que rápidamente se volvió el protocolo líder en cuanto a cifrado Wi-Fi. Para ello, añadieron distintas mejoras como lo son la implementación de un nuevo algoritmo de cifrado, mejoras en la autenticación de usuarios y la integración de nuevos protocolos de seguridad como el protocolo CCMP. [79]

Tipos de protocolo de cifrado WPA2:

- WPA2 -PSK-KIP
- WPA2 -PSK-AES

- WPA2 -MGT-TKIP
- WPA2 -MGT-AES

En el siguiente cuadro se presenta un cuadro comparativo entre los distintos protocolos de cifrado *Wi-Fi*.

Sistema de encriptación	WEP	WPA	WPA2
Estándar	802.11b	802.11g	802.11i
Algoritmo	RC4	RC4TKIP	AES (Rijndael)
Características	Protección a redes inalámbricas vulnerables	IV extendido Llaves dinámicas (TKIP) Incluye MAC del emisor	Número algoritmo de mayor complejidad Tramas convertidas por operaciones matriciales
Longitud de claves	64 (40) o 128 (104) bits	128 a 256 bits	128 a 256 bits
Vulnerabilidad	IV muy corto Llaves estáticas Claves cortas Chequeo de integridad independiente de datos cifrados	Autenticación por handshake auditable. Claves en diccionario, o reconocibles por atacante	Claves conocidas Rondas cortas en información muy confidencial Uso de claves en diccionario o conocidas por atacante
Ataques conocidos	FMS, por estadística de IV, muy exitoso, obteniendo gran cantidad de tramas con IV	Por fuerza bruta comparando claves con handshake, éxito dependiente de tener la clave en el diccionario	Por fuerza bruta muy lenta comparando directamente con la red claves de diccionario, muy poco éxito en bastante tiempo de ataque

Cuadro 6: Evolución de sistemas de codificación inalámbrica. [80]

5.27. Niveles socioeconómicos en Guatemala

La sociedad guatemalteca puede dividirse en siete grupos, que son catalogados de acuerdo a su estándar de vida. Dichos grupos sociales se dividen de la siguiente manera, empezando desde el grupo social más alto:

- A
- B
- C1
- C2
- C3
- D1
- D2

Según este estudio el 62.8% de la población guatemalteca se encuentra en los dos niveles más bajos (D1 y D2). Dentro de este grupo hay distintos niveles de salario, pero el ingreso

mensual promedio se encuentra por debajo de los Q.7,200. El siguiente grupo lo conforma 35.4% de la población perteneciente a la clase media (grupos C1, C2 y C3). Las familias en este grupo cuentan con un ingreso promedio entre Q.11,000 y Q.25,000. Este último grupo genera el 56% del dinero del país. Seguido de este último, se encuentra el grupo B que corresponde a las personas con ingresos mensuales entre los Q.25,000 y Q60,000. Finalmente, el resto de la población corresponde a la clase alta con ingresos por encima de los Q.61,000. [81]

6.1. Diseño de la interfaz gráfica del sistema de comunicación

6.1.1. Prototipo en papel

El diseño e implementación de la interfaz gráfica se realizó durante los meses de junio a septiembre 2019. Se utilizó la metodología del diseño centrado en las personas siguiendo. Para la primera versión se realizaron prototipos estilo *paper mockup* o prototipo en papel, usando la herramienta *balsamiq*.

Con dicha herramienta se diseñaron las siguientes pantallas:

- la página de inicio o *landing page*, donde se coloca una portada con un texto que da una breve introducción al proyecto (Figura 18).
- pantalla de inicio de sesión, donde el usuario introduce su usuario y contraseña (Figura 19).
- página principal de la aplicación donde se ven los anuncios actuales (Figura 20).
- página del chat, donde se observan los diferentes grupos de conversaciones (Figura 21)
- página de la clase en vivo (Figura 22)

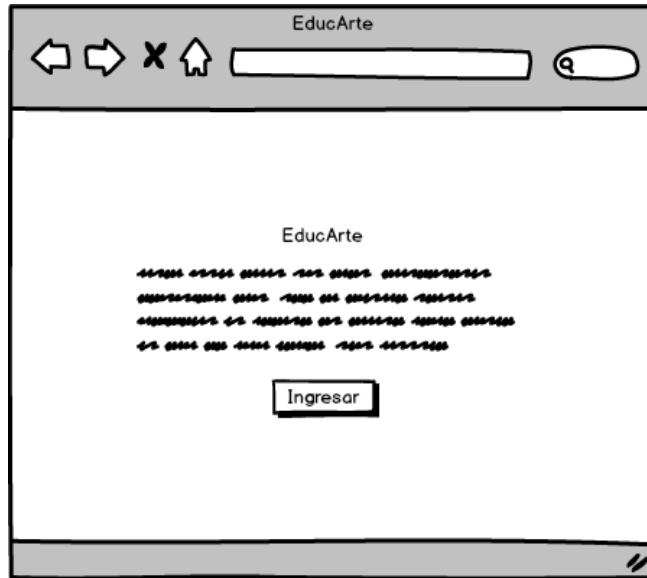


Figura 18: Página de inicio

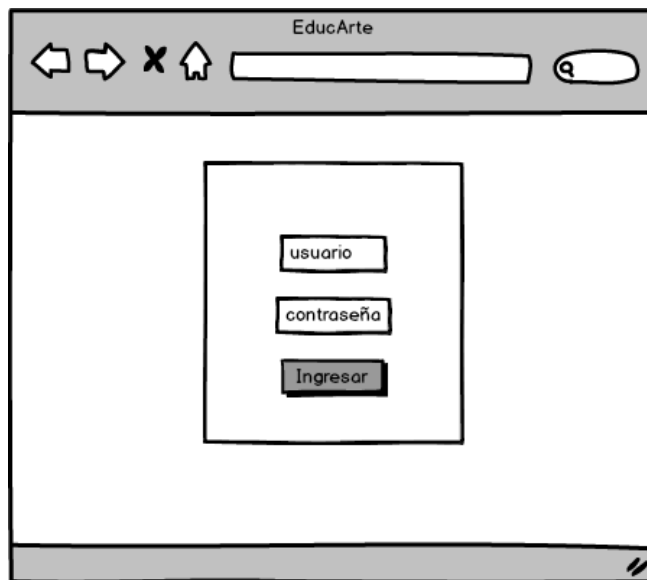


Figura 19: Pantalla de inicio de sesión

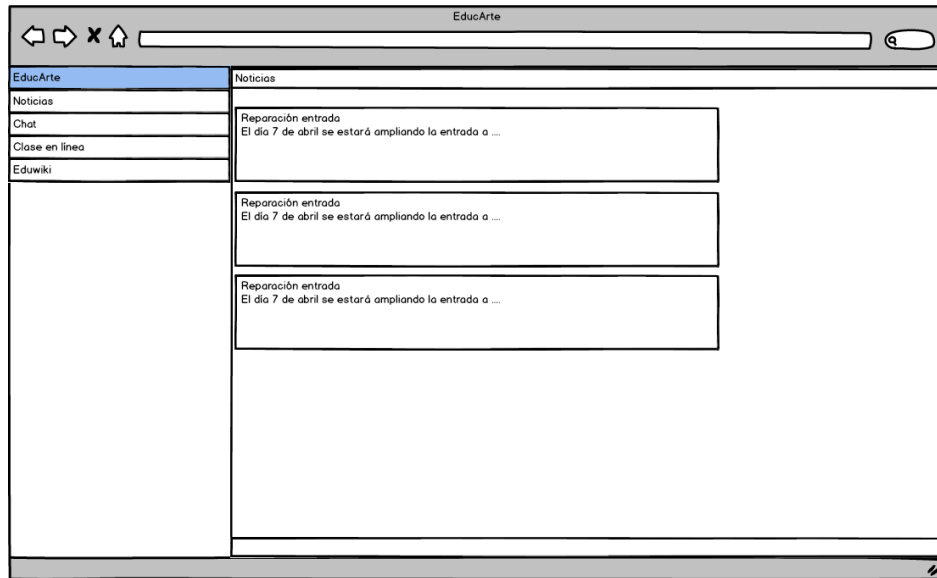


Figura 20: Página principal

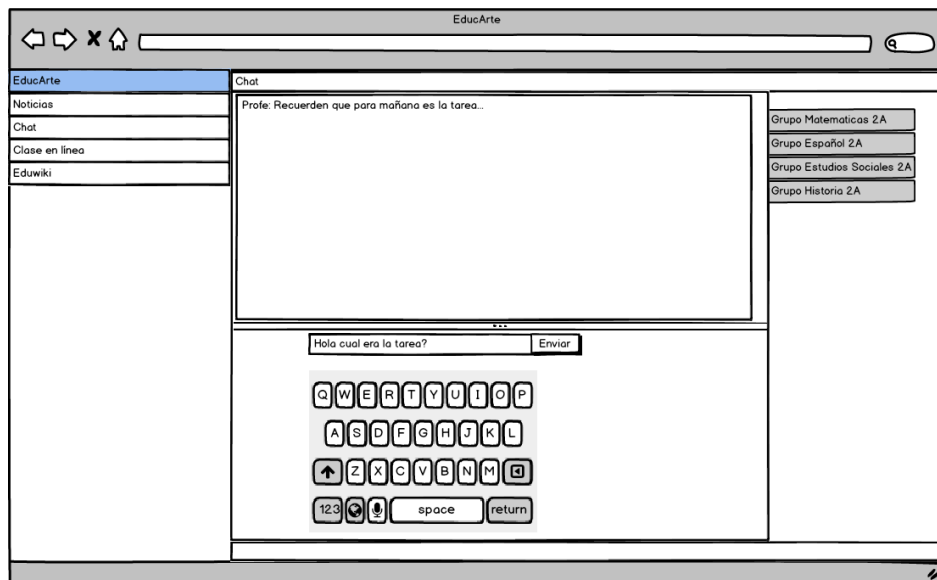


Figura 21: Pantalla de chat

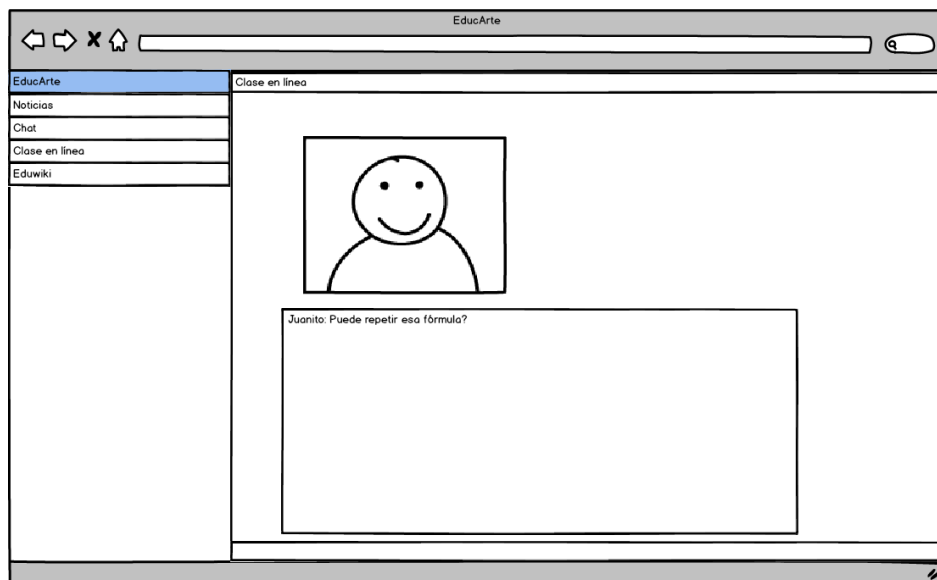


Figura 22: Pantalla de clase en vivo

El diseño de estas pantallas sirvió para tener una idea básica antes de empezar a diseñar el código, con esta versión se realizaron algunas pruebas dentro del equipo para organizar los componentes y luego se procedió a implementar este diseño formalmente.

6.1.2. Prototipo inicial

Se realizó el primer prototipo funcional usando el framework Vuejs. Se escogió *Vue*, ya que como se observa en 2, es el que menos espacio en memoria necesita, tiene la menor curva de aprendizaje, se adapta bien para proyectos medianos y a pesar de ser el más nuevo y no ser mantenido por una gran empresa, como se observa en 6, es el que mayor crecimiento tiene.

Para los estilos se añadieron las clases de CSS de Bulma.io, usando el comando `npm install bulma` y en el archivo `/src/themes/theme.scss` se importa el archivo `bulma.sass`. De esta manera, en cualquier parte del proyecto se puede acceder a las clases de *Bulma*, sin tener que importarlo en todos los componentes.

Se escogió *Bulma.io* por su modularidad, y porque a diferencia de las opciones tradicionales como *bootstrap* o *SemanticUI*, no obliga al desarrollador a usar *jQuery* ni incluye archivos `.js`, sino únicamente las clases CSS. Esto es una ventaja, ya que al usar un *framework* como *Vue*, no es necesario usar *jQuery* para manipular el DOM y se evita agregar más capas que pueden afectar el desempeño.

También es importante mencionar que para el trabajo en equipo de este proyecto se utilizó la metodología de Gitflow. Usar esta metodología ayudó a mantener el orden de las nuevas funciones que se fueron agregando a la aplicación, y mantener separado los cambios de cada módulo.

6.1.3. Implementación del diseño

En toda la interfaz predomina la paleta de colores que se observa en la Figura 23. Se usaron principalmente, los colores amarillo y azul, ya que el amarillo es un color que según la psicología del color, fomenta la felicidad y creatividad, mientras que el contraste con el azul que genera calma y relajación, y esta combinación es ideal para una aplicación educativa.

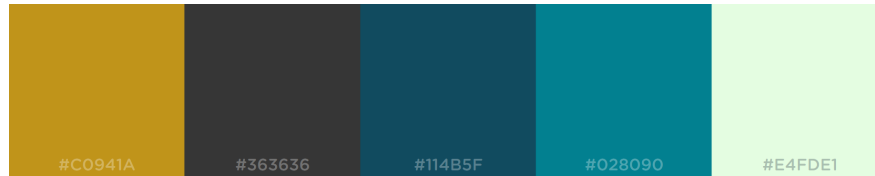


Figura 23: Paleta de colores para la interfaz

Luego de la revisión del prototipo en papel, se decidió agregar una pantalla de composición de anuncio, creación de grupo de chat y creación de clase en vivo, las cuales solo pueden ser vistas por los usuarios con el permiso de administrador/maestro.

Para estas pantallas, se había agregado primero una nueva página de composición usando *breadcrumbs* como se observa en la Figura 24, pero para que el contenido fuera más fluido se utilizaron ventanas modal, ver la Figura 25.

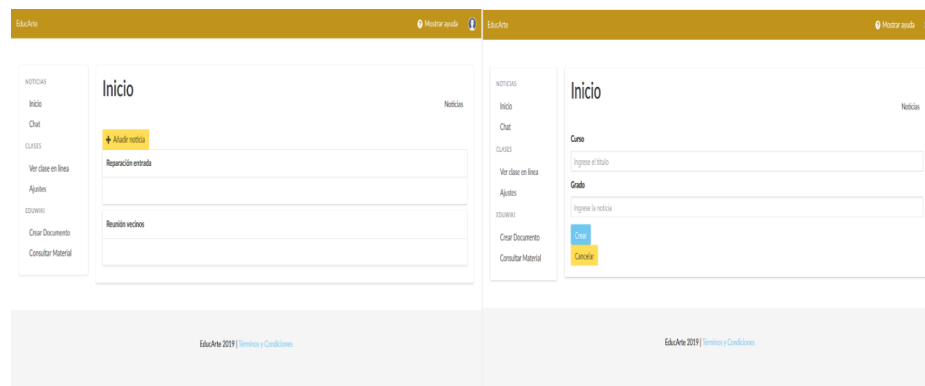


Figura 24: Izquierda pantalla anuncios, derecha pantalla creación de anuncios

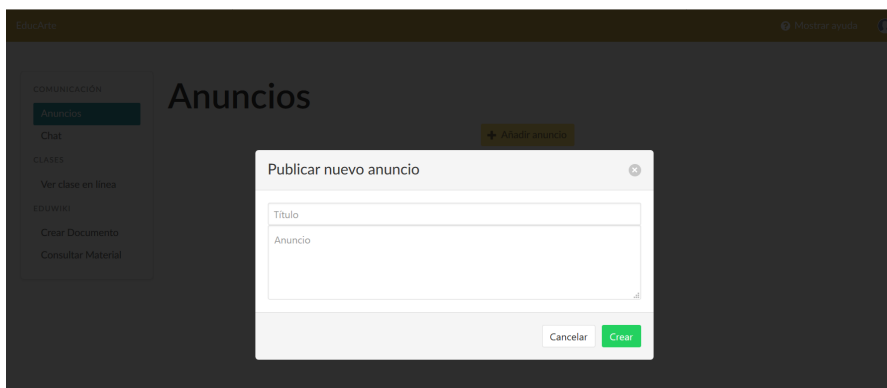


Figura 25: Creación de nuevo anuncio usando modal

6.1.4. Pruebas de usabilidad

Se diseñó la prueba de usabilidad con las siguientes características:

Alcance

Las pruebas no se realizaron sobre toda la aplicación, sino sobre estas tres funciones y tareas respectivas:

- Anuncios: (1) Iniciar sesión y leer anuncios, y (2) la creación, (3) edición y (4) eliminación de anuncios.
- Chat: (5) Acceder al chat, poder enviar y recibir mensajes
- Videollamada: (6) Poder ingresar a una videollamada
- Cerrar sesión (7)

Con estas siete tareas, se cubren las principales funciones de este módulo.

Reclutar usuarios

Los usuarios que participaron en las pruebas fueron personas de la Organización No Gubernamental ACUDE, Asociación para la Cultura y Desarrollo, el día viernes 25 de octubre de 2019.

Identificar objetivos

Objetivo principal: Demostrar la usabilidad de la aplicación.

Objetivos secundarios: Validar que no haya errores en el diseño de la aplicación y encontrar puntos de mejora para facilitar el uso.

Métricas

Se definieron las siguientes métricas:

- Tasa de éxito: Porcentaje de personas que logran completar cada tarea
- Cantidad de clicks: Cuántos clicks tiene que hacer cada persona para completar la tarea
- Tiempo en completar cada tarea

Ejecución de las pruebas

Se realizaron las pruebas con cuatro miembros de la ONG en una oficina en la zona 9 de Guatemala.

Análisis y elaboración de reporte

Finalmente, se recolectaron los resultados de las pruebas y se realizó un resumen que se presenta en el Cuadro 7.

Tarea	Tiempo (s)	Clicks	Tasa éxito %
Iniciar sesión	45.0	5.5	100
Crear anuncio	83.7	5.7	100
Editar anuncio	23.0	3.0	100
Borrar anuncio	24.0	4.5	100
Enviar mensaje en chat	34.0	5.5	100
Crear grupo chat	19.0	3.0	100
Ingresar clase en línea	22.0	5.0	100
Cerrar sesión	11.3	2.3	100

Cuadro 7: Resultados de las pruebas de usabilidad

6.1.5. Visualización final de la aplicación

Finalmente se aplicaron las modificaciones a la interfaz gráfica para mejorar la usabilidad y la visualización de la aplicación.

Interfaz antes de las pruebas

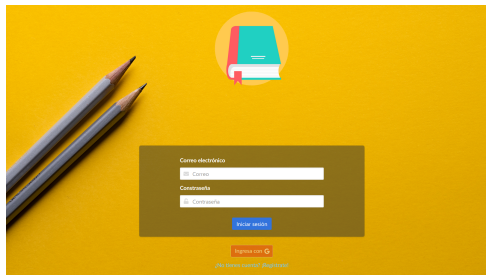


Figura 26: Pantalla de inicio de sesión

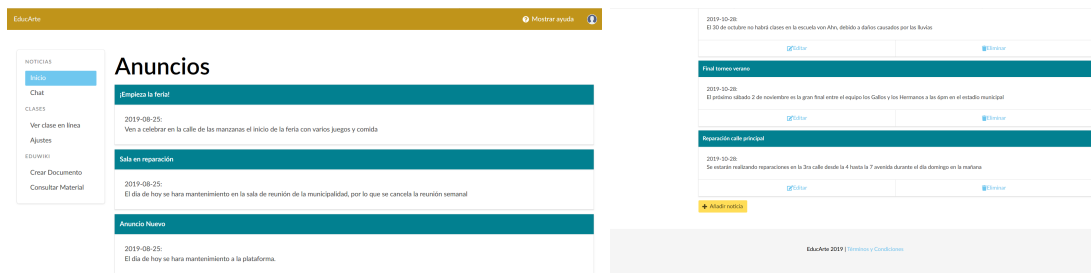


Figura 27: Pantalla de anuncios desde el rol de administrador

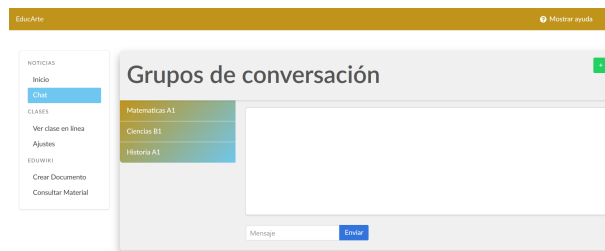


Figura 28: Pantalla de chat

Interfaz final después de las pruebas

En la Figura 29 se agregaron estos textos en el *slider*, de manera de dar información introductoria sobre la aplicación:

- ¿Qué es?: Es una aplicación que te permite acceder a material educativo, leer anuncios importantes de tu comunidad, chatear con las personas que utilizan la aplicación y ver clases en vivo. Todo esto sin necesitar internet.
- ¿Cómo funciona?: Solo necesitas registrarte en la siguiente página e ingresar. Es un servicio gratuito.
- ¿Para quién es esto?: Es para cualquier miembro de la comunidad, pero está enfocado principalmente a mejorar la comunicación entre maestros y estudiantes.

Luego en la Figura 30, se mantuvo el diseño original con el formulario de usuario y contraseña. Se había cambiado en el prototipo inicial a correo y contraseña, pero en las pruebas de usabilidad con la fundación Acude, se recibió como retroalimentación, que no todos los miembros de la comunidad tendrán correo.

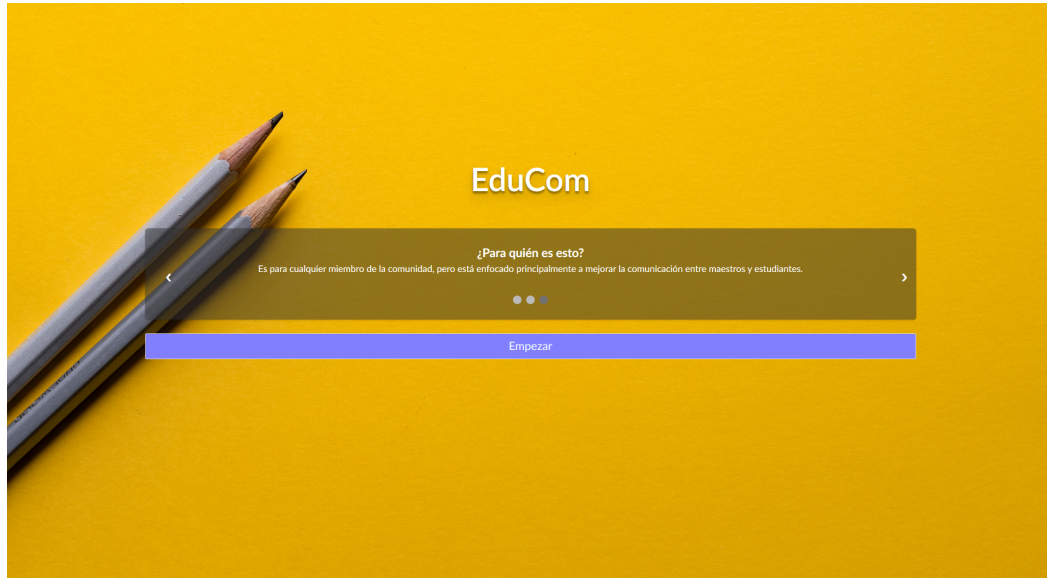


Figura 29: Vista final de la página de inicio

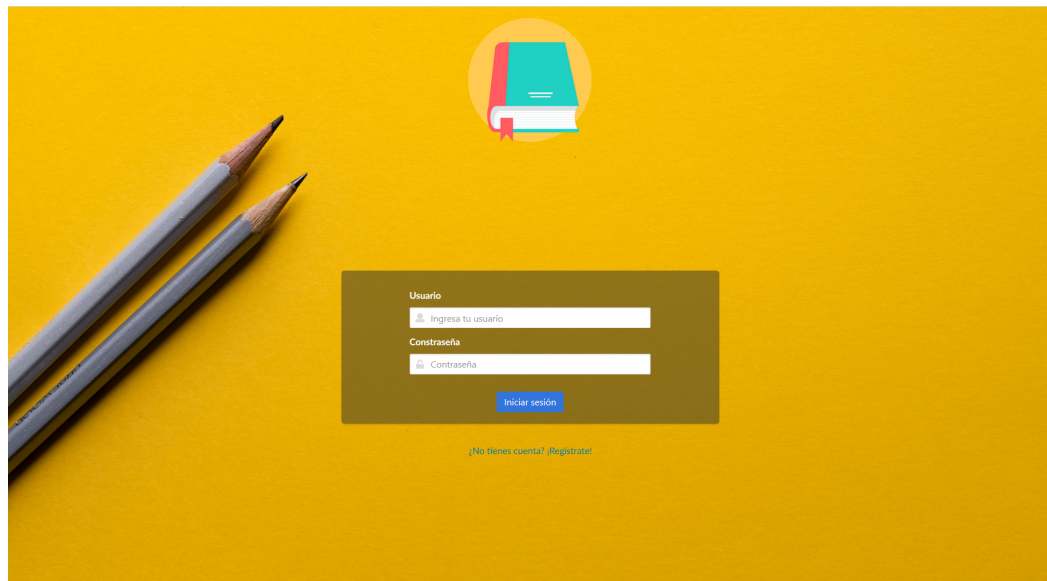


Figura 30: Vista final de la página de inicio de sesión

En la pantalla de anuncios en la Figura 31, se modificó del diseño original visto en la Figura 20 de tal forma que se tengan dos columnas de anuncios, ya que se observó en las pruebas de usabilidad que los anuncios que escriben los usuarios no son tan largos, por lo que se tiene mayor visibilidad al mostrarlos en una pantalla sin necesidad que el usuario tenga que arrastrar el mouse para ver todos los anuncios.

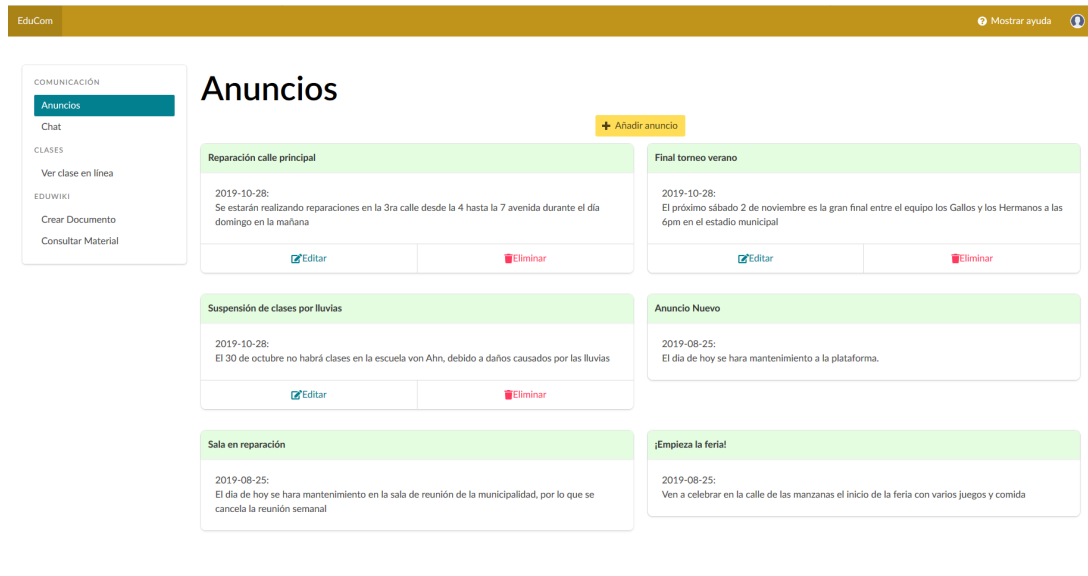


Figura 31: Vista final de la página de anuncios

Además, se cambió el color del botón de eliminar a rojo, para que sea más distinguible y el usuario preste más atención antes de presionarlo, de igual manera se movió el botón de 'Crear Anuncio' hasta arriba, ya que se observó que a los usuarios les toma mucho tiempo encontrarlo si tienen que ir hasta abajo a buscarlo.

Además, se reutilizó la ventana modal para crear y editar anuncios

En la pantalla del chat, se cambiaron los colores y se agregó una función que se marque el grupo de conversación que está activo para que el usuario sepa a dónde está enviando el mensaje. Esto se realizó porque se observó que el usuario presionaba varios clicks para asegurarse que esté en el grupo correcto.

6.2. Red tipo malla

6.2.1. Definición

En el mes de junio del año 2018, se decidió tomar el proyecto de realizar un prototipo de una red de tipo malla, presentado por la Universidad del Valle de Guatemala.

El proyecto consta de dos grandes fases, la primera es toda la investigación que se realizó previa a la ejecución del proyecto como tal y la segunda y fase final es el diseño de prototipo el cual se divide en dos grandes prototipos, la red en malla y la guía para presentar proyectos a comunidades indígenas.

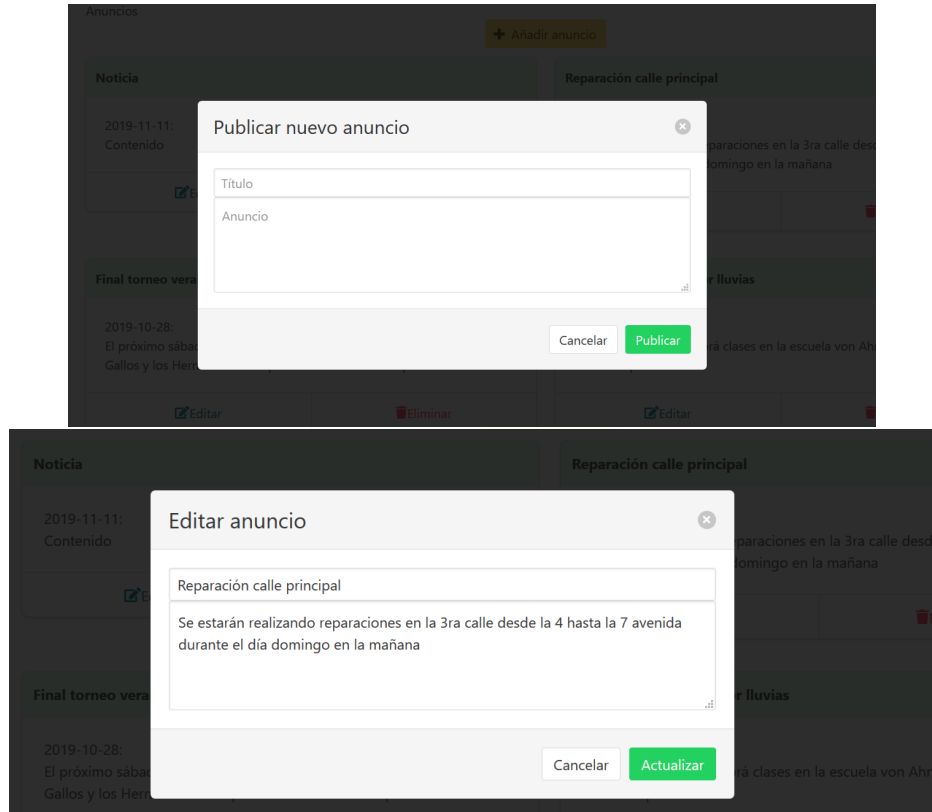


Figura 32: Arriba modal creación, abajo modal actualización

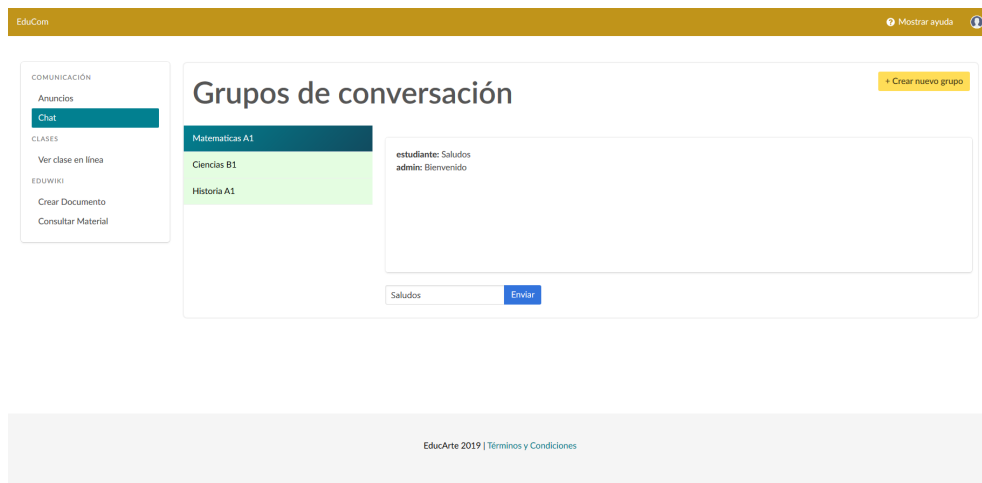


Figura 33: Vista final de la página del chat

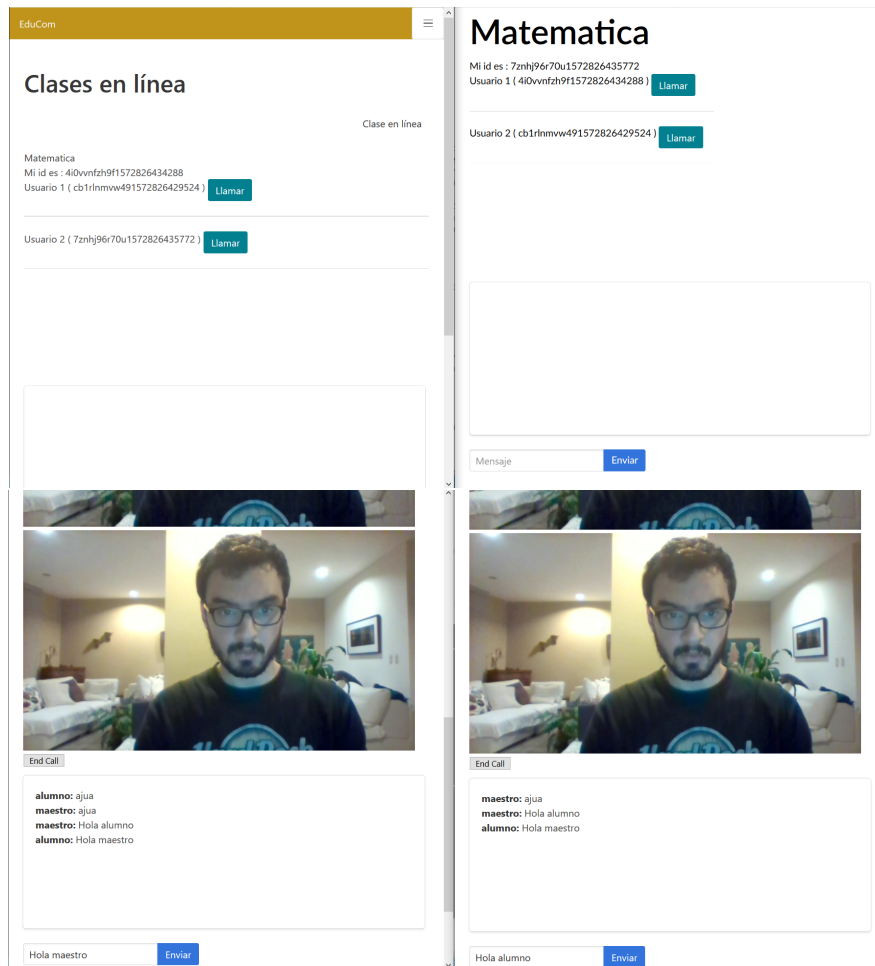


Figura 34: Izquierda persona que inicia la llamada, derecha persona recibiendo la llamada

6.2.2. Investigación

A partir de enero del año 2019 se inició la fase de investigación de *firmwares*, protocolos de enrutamiento, configuraciones de red y equipo para el funcionamiento de una red tipo malla. Se recolectó la información necesaria para las distintas etapas de diseño y prototipo de dicha red. Esta información se procedió a realizar un análisis comparativo entre las diferentes características de distintos *softwares* para el funcionamiento de la red.

Del 7 de enero al 25 de octubre se investigaron soluciones existentes relacionadas a las condiciones topográficas, accidentes geográficos, tipología de vegetación, condiciones de acceso y servicios básicos de la población en estudio.

Paralelo a la investigación anterior, se realizaron entrevistas a entidades municipales con el fin de conocer las necesidades primordiales en tecnología. El primer punto fue la definición de los perfiles de los usuarios de interés, con el fin de identificar las necesidades (5 de febrero). Los perfiles de estudio definidos fueron, personas del interior de Guatemala, las cuales cuentan con dificultades de conexión a internet.

El día 4 de marzo, se realizó una reunión con un funcionario municipal y miembros del COCODE, para establecer formas de comunicación entre los estudiantes y la comunidad objetivo con el fin de darles a conocer los alcances de la implementación de la tecnología de comunicación red en malla.

Durante el mes de octubre, se prepararon entrevistas con agentes de la universidad del valle y organizaciones no lucrativas con proyectos de tecnología para el progreso de pueblos y aldeas de escasos recursos. Estas instituciones cuentan con experiencia en el manejo de proyectos y su aceptación en las comunidades rurales. Tomando como base sus conocimientos sobre la rama de sociología en estas comunidades, se diseñó en conjunto, el prototipo de la guía de aceptación e intervención del proyecto. Esta guía consta de varios pasos, con el fin de que la información sea de fácil manejo para lograr la aceptación del proyecto red tipo malla, por parte de la población y lograr su integración al proyecto de estudio.

Es importante mencionar que a cada entrevistado se le preguntó al inicio, si era posible la grabación con cámara de vídeo o grabación de voz. Además todas estas entrevistas fueron realizadas dentro de las instalaciones correspondientes a cada persona, durante las entrevistas se realizó un proceso de observación, ya que en el momento de efectuar las entrevistas se observa el lugar, para lo cual se tomó nota de las observaciones para un análisis más profundo.

6.2.3. Diseño de prototipo

Prototipo red en malla

En el mes de febrero del año 2018 se solicitó la adquisición de nuevas antenas para el proyecto. El 15 de mayo, se eligió el modelo necesario entre las opciones presentadas por medio de la Universidad del Valle de Guatemala, siendo estas antenas de marca Ubiquity modelo UAP-AC-M.

Tras la entrega en 22 de agosto del año, 2019 se comenzó por realizar una prueba de

UAP-AC-M	
Cobertura	600ft - promedio 300ft
Consumo de energía	24V, 0.5A <i>Gigabit PoE Adapter*</i>
Consumo máximo de energía	8.5W
Conexión	6.5 Mbps a 867 Mbps hasta 100 conexiones

Cuadro 8: Tabla de especificación de antena Ubiquity UAP-AC-M



Figura 35: Antenas UAP-AC-M Ubiquity

conexión con las antenas, utilizando el software propietario de Ubiquiti, UniFi, donde se procedió a diseñar la arquitectura de la red, colocando cada antena como un repetidor y enrutador de la red, las cuales se comunican entre ellas y así generar una red de tipo malla.

Con dicha herramienta se diseño la red que se muestra en la Figura 5.

Una vez se comprobó el funcionamiento correcto de las antenas con el firmware original, se procedió a realizar la evaluación de diferentes firmwares de formato libre para reemplazar Ubiquiti Unifi y así poder integrar diferentes marcas de antenas a la red tipo malla, para lo cual se eligieron 3: OpenWRT, Tomatto y DD-WRT .

Para determinar el software a utilizar se investigo si estos tenían soporte para nuestras antenas, siendo Tomatto el único software que no tiene soporte para estas antenas en específico por lo cual se descartó, seguido se realizó un estudio de actualización y soporte de firmware, concluyendo que OpenWRT al tener una comunidad mayor, el soporte a diferentes antenas es mayor, así como la actualización de firmware para estas, otra métrica que se utilizo fue la libertad de poder configurar cada antena, siendo OpenWRT un firmware que otorga la mayor libertad, con la complicación que este requiere un conocimiento mas extenso del tema.

Tras determinar el firmware a utilizar en las antenas, se procedió el día 28 de agosto, a realizar la instalación del mismo, para lo cual primero se habilitó la opción de ingresar a cada

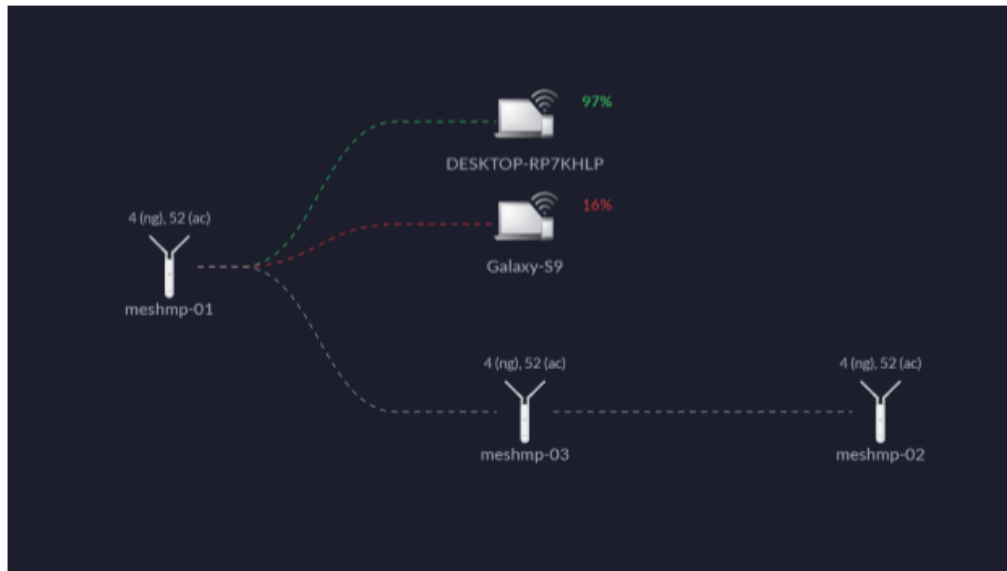


Figura 36: Estructura de red mesh UniFi

antena por medio de SSH, ya que Ubiquiti lo deshabilita para que los usuarios sin experiencia no puedan modificar nada del firmware, este proceso se realizó en las configuraciones de Unifi.

Tras cambiar la configuración de ubiquiti se procedió a establecer conexión física por medio de cable cat5 entre mi computador y una antena, se realizó la conexión por medio de SSH, para ingresar al firmware de la antena y se procedió a descargar el versión del firmware de OpenWRT necesaria para esta antena en específico, una vez se tiene el firmware, se procede a sustituir el *firmware* de la antena y así realizar la instalación de forma correcta.

Una vez se verificó que la instalación de OpenWRT fue exitosa, se procedió a configurar la red inalámbrica y las características de la antena, empezando el día 10 de septiembre , para lo cual se utilizo únicamente la terminal dentro de OpenWRT ya que este no cuenta con interfaz gráfica, al finalizar la configuración pertinente para la antena, se procede a reiniciar el equipo y así este pueda utilizar las nuevas configuraciones, se conecto a un router para poder tener convección a internet y se procedió a realizar la instalación de los protocolos de enrutamiento.

Para determinar qué protocolo de enrutamiento se procederá a utilizar según las necesidades del proyecto se presenta un cuadro comparativo con sus características y se realizan pruebas de conexión y reparación automática por nodos, para lo cual se utiliza el parámetro RTT (Round-Trip time) que determina el tiempo de ida y vuelta de un paquete a través de los nodos, para lo cual se utilizó la herramienta PING. Cabe mencionar que este proceso se realizó para cada antena de forma individual.

Prototipo de protocolo de comunicación e intervención para las comunidades objetivo

La gestión de proyectos puede tener innumerables metodologías para su desarrollo y dependerá de cada población de estudio, un análisis determinado. En el presente trabajo se utilizaron dos metodologías, la primera es una guía estructurada según la recopilación de información, por medio de entrevistas. Esta metodología tiene como base, parte del estudio que se realizó con ACUDE, la fundación Ramiro Castillo Love y parte de personal docente y estudiantes de la Universidad del Valle de Guatemala. La segunda opción tiene como base, parte de la investigación realizada en la tesis "Guía para la gestión de proyectos ante los consejos de desarrollo Urbano y Rural"[82], en el entendido de que cada comunidad necesita un análisis específico y técnicas específicas para ejecutar el consentimiento informado de cada proyecto por parte de la misma población.

Paralelo a todo el procedimiento de instalación, prueba de *firmware* y protocolos de enrutamiento, se realizaron reuniones con personal docente y alumnos de ciencias sociales de la Universidad del Valle de Guatemala, personeros de la municipalidad, miembros de la comunidad y miembros de organizaciones sociales sin fin de lucro, las cuales trabajan dentro de comunidades indígenas y de escasos recursos, para discutir sobre la estructura básica que cada proyecto debe establecer para la integración de comunidades rurales en Guatemala.

El día, 5 y 6 de septiembre, se estableció la primera reunión con miembros de la municipalidad en estudio y sus líderes comunitarios, para establecer puntos importantes en la aceptación de proyectos de tecnología dentro de la comunidad. Para estas reuniones se procedió a utilizar el método de lluvia de ideas por parte de los integrantes de la comunidad, para conocer sus puntos de vista y cuales podrían ser importantes para la aceptación del proyecto.

El día 30 de septiembre se realizaron reuniones con personeros de la Universidad del Valle de Guatemala, con conocimientos del tema social para establecer puntos importantes sobre el acercamiento y la integración de la comunidad de forma activa en el desarrollo de proyectos tecnológicos, dichas reuniones fueron de tipo foro donde cada integrante de forma separada expuso sus experiencias en cuanto la relación que se debe tener con los líderes comunitarios.

El día 19 de octubre, se realizó una reunión con agentes de diferentes organizaciones no lucrativas en el ámbito tecnológico, donde estos expresaron las experiencias que han obtenido a lo largo del tiempo, para lograr un nivel de aceptación alto, por parte de las comunidades rurales, hacia sus proyectos.

Para documentar las ideas obtenidas en las reuniones antes mencionadas se procedió a grabar las conversaciones y se documentaron las notas escritas.

Para poder entablar una comunicación guiada hacia la comunidad, se utilizó como una herramienta de trabajo la tesis "Guía para la gestión de proyectos ante los consejos de desarrollo Urbano y Rural"[82] elaborada por el arquitecto Otto Rene Rojas Miranda, de la Universidad de San Carlos de Guatemala, estableciendo los puntos más importantes para desarrollar un acercamiento más estructurado.

El día 28 de octubre, se realizó la última reunión con un estudiante y catedrático de la rama de Ciencias Sociales de la Universidad del Valle de Guatemala, para discutir la estructura final del prototipo de protocolo de comunicación e intervención para la comunidad objetivo. Así mismo se analizó y se estructuró la base para desarrollar el Consentimiento Informado, el cual se debe presentar ante los líderes respectivos como una carta introductoria del proyecto.

6.3. Diseño de la interfaz gráfica de la aplicación educativa

6.3.1. Enfoque y variables de investigación

El enfoque de este módulo es mixto (cualitativo y cuantitativo). Esto se debe a que se realizó el módulo de *frontend* y experiencia de usuario para la herramienta educativa que estará acoplada con la red *mesh*. Para poder tener un análisis de resultados efectivo, se realizaron pruebas de usabilidad con los potenciales usuarios de la aplicación. Estas pruebas pueden ser consideradas como un método tanto cualitativo como cuantitativo.

Entre las variables de investigación, se encuentran los indicadores de usabilidad de la aplicación educativa, entre los cuales se incluye el cumplimiento de requisitos funcionales a través de la realización de tareas por parte de los potenciales usuarios. El principal instrumento de recolección de datos fue la ejecución de las pruebas de usabilidad, la cuales fueron descritas en el marco teórico.

La población seleccionada fueron profesores escolares y estudiantes de nivel primario y nivel medio, ambos perfiles pertenecientes a una comunidad rural de Guatemala.

Las fases de trabajo para este módulo fueron las siguientes:

1. Investigación: se investigó acerca de principios de diseño y usabilidad, enfocados hacia la población estudiada.
2. Desarrollo: se implementó la solución al problema observado, basándose en los principios y la teoría investigada en la primera fase.
3. Evaluación: se hicieron pruebas de usabilidad con los sujetos de estudio para la solución creada, analizando qué detalles deben cambiar para una nueva iteración (pasos 2 y 3).
4. Conclusiones: se desarrollaron conclusiones con respecto al resultado obtenido en la última iteración, confirmando o rechazando la utilidad de la solución planteada.

6.3.2. Selección de tecnología

Entre los tres *frameworks* para desarrollo web que se explicaron en el marco teórico de este documento, se seleccionó VueJS como la tecnología principal a utilizar para el desarrollo de la interfaz de usuario para la aplicación educativa. Se tomaron en cuenta los siguientes aspectos entre los tres *frameworks* para esta elección:

- Angular fue descartado primero, puesto que tenía el mayor tamaño entre los tres *frameworks*, está orientado para proyectos grandes y no utiliza un DOM virtual (definido en la sección 4.6.1) como lo hacen ReactJS y VueJS, lo cual lo hace tener un rendimiento lento.
- El tamaño de VueJS es más pequeño que el de ReactJS, lo cual es clave en el ambiente en que se encontrará el servicio, puesto que se tendrán recursos limitados.
- VueJS es más recomendado para proyectos medianos/pequeños, a comparación de React que se encuentra más orientado a proyectos grandes/medianos, principalmente por la cantidad de código que debe desarrollarse desde un principio con React.
- La curva de aprendizaje de VueJS es la menor, lo cual se consideró que podía ayudar a desarrollar el proyecto con mayor agilidad, puesto que varios de los miembros del equipo no tenían conocimiento de algún *framework* para desarrollo web.

6.3.3. Prototipado

El primer paso para realizar el prototipado fue listar los requisitos funcionales que el módulo debía tener, en orden de prioridad. Los requisitos definidos fueron los siguientes:

1. Desarrollar una herramienta para crear *markdowns*, para que los encargados de publicar material educativo realicen esta tarea en dicha herramienta.
2. Permitir la previsualización del documento por parte de los encargados de publicar material educativo, previo a su publicación.
3. Crear vistas para navegar dentro de la aplicación y seleccionar grados, clases y documentos creados en un grado y clase en específico.
4. Desarrollar una vista para que tanto alumnos como los encargados de publicar material puedan visualizar los documentos publicados.
5. Crear una vista para que los encargados de publicar material puedan editar documentos que ya han sido publicados.
6. Integrar el prototipo con la capa de *backend* (desarrollada como parte de otro módulo) para obtener información de grados, clases, documentos creados y también para publicar y editar los documentos ya publicados.

6.3.4. Plan de evaluación

El primer proceso de las pruebas de usabilidad, como se mencionó en la sección 4.4.1, es crear un plan de pruebas. Primero se definió el alcance de la evaluación, donde se decidió evaluar la usabilidad de la aplicación educativa, tanto en la sección para crear material educativo como para acceder a este mismo y visualizarlo.

En un principio, se determinó que los sujetos de investigación debían ser personas que se encuentren realizando estudios a nivel primario o a nivel medio, y profesores también

de escuelas en estos mismos niveles educativos. Ambos perfiles debían pertenecer a una comunidad rural, la cual sería una potencial ubicación para la instalación de la red *mesh*. La unidad de análisis es el individuo estudiado.

Se realizó una reunión con personal de la Asociación para la Cultura y el Desarrollo (ACUDE), en conjunto con un miembro de la Fundación Ramiro Castillo Love del Banco Industrial. ACUDE tiene en desarrollo varios proyectos para mejorar las condiciones de vida de las poblaciones en el área rural, entre las cuales se encuentra el plan Educación Integral del Nuevo Milenio. Este proyecto, que se realiza en colaboración con la Fundación Ramiro Castillo Love, consiste en ejecutar un programa de alfabetización y post-alfabetización a niños y jóvenes en sobre edad escolar, en comunidades indígenas, cantones, aldeas y caseríos en los departamentos de Chiquimula, Quiché y Totonicapán.

Por motivo del tiempo disponible para el prototipado y la realización de pruebas, se consideró con el grupo de trabajo que era más factible realizar las pruebas de usabilidad con los miembros mencionados de ACUDE. Estos miembros de la asociación cumplían con el rol de capacitar a los educadores en cuanto al uso de los recursos computacionales brindados por la Fundación Ramiro Castillo Love y también en cuanto a la impartición de clases hacia los estudiantes.

La principal ventaja de este acercamiento es que las oficinas de ACUDE se encuentran dentro de la Ciudad de Guatemala, por lo que la ejecución de las pruebas se realizó de manera más fácil y rápida. Además, los miembros seleccionados de la asociación tenían experiencia capacitando en las áreas rurales y también tenían conocimiento de varias características de los potenciales usuarios y de su contexto, correspondientes a varios de los aspectos del *framework* ABCS mencionado en el marco teórico.

Luego de definir los sujetos de estudio, se definieron las tareas para las pruebas. Aquí se definen qué tareas específicas relacionadas al producto realizaron los usuarios evaluados. Se definieron tareas en el siguiente orden:

1. Comenzar a crear un nuevo documento para una clase en un grado en específico. Esto se hace para evaluar la navegación desde el menú principal hasta la pantalla mencionada.
2. Crear un nuevo documento en una clase distinta a la seleccionada. Esta tarea se realiza para evaluar el control del usuario y prevención de errores.
3. Agregar al documento un título y agregar también título e información a la primera sección. Se desea evaluar con esta tarea la relación entre el sistema y el mundo real.
4. Agregar una sección adicional al documento y colocarle tanto título como contenido.
5. Eliminar una de las secciones. También se realiza para evaluar control de usuario y prevención de errores.
6. Responder al evaluador en qué grado y clase se está creando el documento. Esta tarea ayuda a evaluar la visibilidad del estado del sistema.
7. Previsualizar el documento, para verlo desde la perspectiva del estudiante.
8. Salir de la vista previa y seguir editando el documento. Se usa para evaluar la consistencia y estándares.

9. Publicar el documento. También se usa para evaluar la consistencia y estándares.

6.3.5. Ejecución de las pruebas

Para agilizar el proceso de diseño y validación, se utilizó el modelo de prototipado horizontal para las primeras iteraciones, el cual también fue mencionado y explicado en el marco teórico del presente documento.

El prototipado horizontal hace posible evaluar la interfaz de usuario en su totalidad, puesto que toma menos tiempo realizarla que al utilizar un modelo de prototipado vertical, en el que se realizan funciones reales, pero únicamente se pueden evaluar individualmente. Realizar un prototipado horizontal permite reconocer potenciales problemas de usabilidad en el diseño desarrollado de manera temprana.

La ejecución de las pruebas consistió primero en explicar a los usuarios evaluados el giro de la aplicación, además del rol que ellos tendrían dentro del contexto de esta misma. Posterior a esto, se procedió a pedir al usuario que intentara ejecutar las tareas enumeradas en la sección 5.3. Se utilizaron principalmente preguntas como "¿Cómo realizaría usted 'X' tarea?", "¿Cuál sería el proceso para poder alcanzar 'X' objetivo o estado", etc.

En los anexos de este documento, se encuentra un guión introductorio que se utilizó con los usuarios previo a la prueba. El principal rol del evaluador consistió en observar y medir la ejecución de las pruebas, además de tomar apuntes de comentarios que el usuario brindó durante la sesión.

La cantidad total de usuarios con los que se realizaron pruebas de usabilidad fue de 4, cercano al número recomendado que se mencionó también en el marco teórico. Se tomaron las siguientes métricas durante y después de la evaluación, para medir la usabilidad:

- Tasa de completación de cada tarea.
- Tiempo que tomó realizar la tarea.
- Cantidad de pasos utilizados para ejecutar la tarea.

La ecuación 4 indica cómo calcular la Tasa de completación de cada tarea. En cuanto al tiempo y cantidad de pasos, se comparó este número contra el tiempo que tomó al desarrollador realizar la tarea. Se considera que un 70 % de éxito sobre este número obtenido es aceptable, además que este número puede ser extrapolado a otro tipo de métricas similares a esta [83].

$$\text{Tasa de completación} = \frac{\text{Número de personas que completaron la tarea}}{\text{Número total de personas evaluadas}} * 100 \% \quad (4)$$

El presente megaproyecto tiene como objetivo principal realizar un modelo de *Red Mesh*. A través de este modelo se introducirán a una comunidad distintos servicios, cuyo propósito es ampliar las capacidades de comunicación y el acceso a la información para sus usuarios.

Este módulo se centrará en el diseño de un modelo de infraestructura de la red, es decir, todo lo concerniente a configuraciones tanto de equipos como de la instalación física de la red, así como las configuraciones necesarias para aspectos como lo son protocolos de comunicación y de seguridad. Además, el diseño de la infraestructura de la red deberá adecuarse a las necesidades y capacidades existentes en la comunidad objetivo. Con esto nos referimos a que el modelo de red tiene que contemplar todas las limitantes que se puedan presentar como lo son: los problemas de conexión física, las vulnerabilidades de seguridad en la red y los problemas de alimentación de energía eléctrica para su continuo funcionamiento.

Por último, este módulo también contempla una propuesta de diseño de una guía de comunicación entre los miembros del equipo de diseñadores de la red y las personas líderes de la comunidad, que son el principal vínculo entre el equipo y los usuarios para los que se piensa introducir este modelo de red.

6.4. Diseño del Backend para el sistema de comunicación

El diseño del Backend para la plataforma se realizó durante los meses de Mayo 2019 a Septiembre de 2019. Durante el diseño se tomaron en cuenta las características que se previeron de la *Mesh Network* a implementar. Entre estas:

- Internet limitado, tanto en velocidad como en disponibilidad.
- Capacidad limitada de almacenamiento.
- Capacidad limitada de procesamiento.

Además, para que la aplicación cumpla con todas las tareas propuestas, el *backend* debe proveer las siguientes funcionalidades:

- Agregar, modificar y eliminar noticias.
- Agregar, modificar y eliminar salas independientes de mensajería instantánea.
- Agregar, modificar y eliminar salas con retransmisión en directo de audio y video con su propia sala de mensajería instantánea.
- La comunicación entre navegadores para establecer una conexión para la retransmisión en directo de audio y video.
- Envío de mensajes de texto entre todo aquel usuario que se encuentre dentro de una sala, independiente de las demás salas.

Dichos puntos mencionados sobre la capacidad de la red en la cual se implementaría la aplicación, y tomando en cuenta las funcionalidades a implementar en la misma, dieron pauta a la elección de la arquitectura de la aplicación y el servidor. Se decidió usar la arquitectura basada en servicios debido a la complejidad y extensión del módulo, permitiendo centralizar el manejo de la información, agregar una capa de abstracción entre el cliente

y la información; y permite que el sistema sea un servicio independiente, ayudando a su depuración y mantenimiento. Además, agregar nuevas funcionalidades en un futuro se vuelve un trabajo menos pesado. [84]

Para la retransmisión de video se decidió utilizar la tecnología WebRTC. Esta tecnología viene incrustada en los navegadores de red modernos, tales como Google Chrome, Mozilla Firefox y Microsoft Edge. Esta tecnología permite la transmisión de diferentes tipos de información entre navegadores, entre ellos audio y video. La razón detrás del uso de esta tecnología surge de la necesidad de evitar usar aplicaciones extras (tercerizadas) para el envío de información en forma de audio y video entre navegadores, creando una mejor experiencia para el usuario. Esta tecnología, sin embargo, requiere de un servidor que se conoce como Servidor de Señalización (*Signaling Server*, en inglés), el cual fue el implementado en este módulo.

En la retransmisión de audio y video, esta será unidireccional entre los usuarios ya que se tiene pensado que únicamente el catedrático transmitirá su audio y vídeo hacia el resto de los integrantes de la sala. Dicho esto, el diseño del Servidor de Señalización para la comunicación entre los navegadores se realizó de forma que funcionará con una topología de tipo estrella, donde el nodo central será el sistema del catedrático. Esto significa que cualquier cliente nuevo en la sala se conectará únicamente con el transmisor por medio de WebRTC. Cada uno de estos deberá iniciar automáticamente el proceso para establecer la conexión WebRTC una vez haya ingresado a la sala, para poder recibir la retransmisión de audio y video.

La definición de rutas para el uso del servidor por medio del cliente se fueron definiendo en conjunto con el frontend a lo largo de todo el proyecto. De igual forma, la definición de la comunicación de información se hizo en conjunto para tener en concreto la información que se requería de cada lado.

6.4.1. Implementación del Backend

Se tomaron en cuenta dichos puntos para buscar las herramientas adecuadas para el desarrollo del servidor y se consideraron las siguientes: Node.js, Python y Golang. Cada una tiene sus ventajas y desventajas, ofreciendo diferentes facilidades y dificultades a la hora de desarrollar la aplicación deseada, sin embargo, al final se eligió Node.js. La decisión de esta herramienta fue porque las funciones principales del servidor era intercambiar información entre los usuarios mediante mensajería instantánea y retransmisión de video. Node.js es una herramienta que sobresale en comunicación en tiempo real, dado a su naturaleza basada en eventos y la implementación de WebSockets por medio del framework Socket.io.

El primer prototipo de la aplicación se desarrolló en el mes de Septiembre, en dicho prototipo se desarrolló e implementó el servicio para la agregación, modificación y eliminación de anuncios. De igual forma, los servicios con las mismas acciones para las salas de mensajería instantánea y las salas para la retransmisión de audio y video, se desarrollaron e implementaron. Las primeras pruebas de usabilidad se realizaron sobre este primer prototipo, y se llevaron a cabo en el mes de Septiembre, en conjunto con el cliente desarrollado en otros módulos del proyecto.

En este prototipo se implementó una RESTful API capaz de responder a las solicitudes del lado del cliente para realizar las acciones de crear, modificar, eliminar y recuperar los anuncios. Para las salas de, tanto mensajería instantánea, como de retransmisión de audio y video, se implementaron las acciones de crear, eliminar y recuperar. Para las acciones, en sus respectivos servicios, se implementaron con los siguientes métodos:

- GET
- POST
- PUT
- DELETE

Para los métodos de POST y PUT, que requieren el envío de un cuerpo del lado del cliente, fue también definido en formato JSON. En todos los servicios, la respuesta del servidor fue en formato JSON. En toda respuesta del servidor se incluyó un estado para informar al cliente del resultado de su solicitud. Se utilizaron los siguientes estados, definidos por la RFC 7231[85]:

- Estado 200: OK - Solicitud atendida correctamente.
- Estado 404: ERROR - Se produjo un error al procesar la solicitud.

El segundo prototipo se realizó en el mes de octubre, además de tener los servicios antes mencionados, se implementó el servicio para tener una comunicación en tiempo real por medio de mensajes de texto. En conjunto, se implementó el Servidor de Señalización para la retransmisión de video y audio por medio de la tecnología WebRTC. Se realizaron las pruebas de usabilidad para este prototipo el mes de octubre en conjunto con el cliente y el servidor desarrollados en otros módulos del proyecto.

6.4.2. Funcionalidad del sistema de comunicación

Con el desarrollo de una RESTful API en mente, se espera que los clientes sean capaces de utilizar los servicios de forma dinámica, es decir, poder utilizarlos sin necesidad de preocuparse si hay más clientes usando el servicio. Una de las ventajas que implica el usar una API, como se mencionaba, es escalable. Si se requiere de un servidor con un mayor poder computacional, se puede adquirir un nuevo sistema al cual únicamente se tendría que desplegar la aplicación ya desarrollada. Esto significa que las capacidades de la aplicación dependen del sistema en donde se despliega.

Esto trae el beneficio que mucho del poder computacional de la aplicación recae sobre el servidor, permitiendo que más números de dispositivos no encuentren problemas al usar la aplicación por limitaciones de hardware. Al mismo tiempo, envuelve el manejo de la información en una caja negra, es decir, crea una capa de abstracción entre el usuario y la información. Esto aumenta la seguridad de la información, rebajando la posibilidad de brechas, corrupción de la *data*, entre otros. Además, la información se muestra de una forma más fiable y consistente.

El funcionamiento de la API es simple, el cliente debe mandar la solicitud que desea al servidor por medio de la API, a lo cual el servidor enviará una respuesta. En caso de haber ocurrido un error, se envía un mensaje que la solicitud falló, y de lo contrario, envía la información solicitada de una forma concisa y consistente.

Por otro lado, se tienen los sockets con el servidor, los cuales permiten que se esté mandando información de forma constante. Una vez se abre un socket, el servidor estará escuchando constantemente por señales emitidas, y al mismo tiempo, mandará señales a los clientes con información ya definida. Una vez abierto un socket, el servidor espera que señales ya definidas se envíen junto con la información esperada.

6.5. Diseño de la red

El diseño de una red no es un proceso algorítmico en el cual se ejecuten una serie de pasos. Normalmente, el diseño de una red está más orientado a las preguntas “¿Qué es lo que tengo?” vs “¿Qué quiero lograr”, o en otras palabras, “recursos” vs “objetivos”. Por lo tanto, este diseño de red está orientado a trabajar con los recursos existentes para llegar a los objetivos deseados.

Para ello es necesario responder las siguientes preguntas:

6.5.1. ¿Quiénes van a usar la red?

Los usuarios de la red son personas que residen en comunidades del área rural de Guatemala y que poseen un nivel socio económico muy bajo, para ser más específicos, parte del sector socio económico D2. [86] A estos usuarios se les dará acceso al uso de varios servicios diseñados específicamente con base en las necesidades de la comunidad.

Dichos usuarios en su mayoría son jóvenes, pero se contempla el hecho que los usuarios puedan ser de distintas edades incluyendo niños y adultos. Este tipo de usuarios normalmente poseen dispositivos móviles de gama media/baja. También, los usuarios son estudiantes o personal que hacen uso de los laboratorios de computación de escuelas públicas o de programas educativos ejecutados por ONG.

6.5.2. ¿Para qué se va a utilizar la red?

La red será utilizada por los usuarios para consumir distintos micro servicios que serán implementados en la misma. Para este proyecto los servicios propuestos son los siguientes:

1. Servicio de *Wikis*.

Es utilizado para la creación de páginas de tipo “*Wiki*” para presentar de manera digital el contenido del curso que imparten los catedráticos de la escuela de la comunidad o del programa educativo que esté siendo implementado en la comunidad. Con este servicio se agilizará, facilitará y estandariza la forma de presentar de manera digital

los contenidos educativos. Se espera que con este servicio los catedráticos se apoye a los catedráticos a preparar de una manera más ordenada y estructurada el contenido del curso que imparten. También se busca promover el uso de la tecnología de una manera más cotidiana en las nuevas generaciones de catedráticos y estudiantes en las escuelas y programas educativos del área rural.

Este servicio debe ser administrado por algún encargado designado, que puede ser un catedrático o director que se encargue de realizar la administración de roles y usuarios dentro del servicio.

2. Servicio de videochat / videoconferencia.

Varias ONG participan en la creación de laboratorios de computación para programas educativos o, en ciertas ocasiones, como donaciones para escuelas de las localidades. Este servicio es utilizado en dichos laboratorios de computación para el *Streaming* de videoconferencias entre catedráticos y estudiantes. Se espera que con este servicio los catedráticos o técnicos que imparten cursos en estos laboratorios utilicen este servicio para realizar cursos más interactivos y mejorar los procesos de comunicación entre ellos y sus estudiantes. De esta manera aumentar la participación e interacción entre los usuarios y el catedrático, y promover el uso de tecnologías de maneras innovadoras en los procesos educativos.

El sistema que se utilizará para la administración de usuarios es el mismo sistema utilizado en el servicio anterior, por lo tanto el gestor del sistema es el mismo encargado.

3. Servicio de mensajería instantánea

Este servicio es un servicio de mensajería instantánea que está pensado para ser utilizado en conjunto con el servicio de videochat/videoconferencia. Este servicio es utilizado para enviar mensajes de manera instantánea entre usuarios de la red. Se espera que con este servicio se incremente las capacidades de comunicación entre los usuarios de la red, y en el caso de su uso para programas educativos, mejorar el proceso de comunicación entre el catedrático de un curso y sus estudiantes.

El sistema que se utilizará para la administración de usuarios es el mismo sistema utilizado para los dos servicios mencionados anteriormente, por lo tanto el gestor del sistema es el mismo encargado

4. Servicio de anuncios / noticias.

Este servicio es utilizado para la visualización y publicación de noticias o anuncios de y para la comunidad. Se espera que este servicio permita que los usuarios que utilicen la red puedan tener completa disponibilidad de estos anuncios y noticias. A diferencia de los otros servicios mencionados anteriormente este servicio no depende del proceso de administración de usuarios que los demás servicios utilizan. El acceso a las noticias y anuncios de este servicio es completamente libre y fácil para los usuarios.

Los usuarios de la red consumen estos servicios a través del uso de la red inalámbrica, accediendo desde celulares, desde los equipos de computación de los laboratorios o cualquier otro dispositivo móvil. Por último, es necesario contemplar que la red tenga la capacidad de incrementar la cantidad de servicios que ofrece.

Todos los servicios mencionados anteriormente residirán en el servidor central de la red, especificado un poco más a detalle más adelante en el documento. Este servidor central es el encargado de servir las páginas web y los contenidos que los usuarios soliciten.

6.5.3. ¿Hasta qué punto se va a expandir la red?

La red está diseñada para que pueda crecer dinámicamente según el tamaño del área de la comunidad que se requiera cubrir. También, la red puede ser replicable en otras comunidades en donde se tengan necesidades y condiciones similares.

6.5.4. ¿Valor de los datos para establecer seguridad?

El valor de los datos para este proyecto es algo muy complejo calcular. Esto se debe a que el valor de los datos lo establece cada servicio particular que sea implementado dentro de la red. Claro está que no todos los servicios requieren y manejan la misma información, por lo que además de que no están definidos todos los servicios que la red gestionaría, evaluar cuál es el valor de los datos no es posible. Este proyecto se deja como abierto para futuras implementaciones de controles adicionales de seguridad y al estar en una etapa inicial, principalmente de diseño, se le da prioridad a otros aspectos como la disponibilidad y resiliencia de la red.

6.5.5. ¿Presupuesto?

A pesar de no tener un presupuesto definido, se tiene presente que la red debe ser realizada con bajos costos. En general, el equipo seleccionado para el diseño de esta red está orientado a presentar un alto grado de costo beneficio como lo son los casos de las antenas y el uso de Raspberry Pi.

6.5.6. ¿Disponibilidad de la red?

Debido a las condiciones en las que se tienen contempladas en cuanto a la implementación de la red y el enfoque de sus servicios, se requiere de alta disponibilidad.

6.5.7. ¿Qué topología?

Red en Malla. Esta topología fue seleccionada debido a que presenta las mejores características para formar una red resiliente y enfocada en alta disponibilidad.

6.6. Seguridad de la red

Las redes WLAN desprotegidas están expuestas a muchos peligros. Es imperativo tener en mente que cualquier usuario es un potencial riesgo. Debido a que es fácil controlar el tráfico en la redes inalámbricas, usuarios que puedan recolectar información privada o saturación intencional de la red son riesgos cada vez más comunes. Normalmente, todos estas amenazas son riesgos que ya están presentes y que tarde o temprano pueden ocurrir debido a las brechas de seguridad en la red. Si no se conoce qué es lo que se quiere proteger y porqué, aplicar medidas de seguridad es como caminar a ciegas. Es crítico para cualquier red identificar los recursos que se quieren proteger y qué impacto tiene que se dañen, pierdan o roben alguno de estos recursos.

A continuación, se describen los puntos de seguridad evaluados para esta implementación de *red mesh*

6.6.1. Identificación de recursos y requisitos de la red:

Recursos

Dispositivos:

- Servidor central (Raspberry PI).

Para las pruebas realizadas en este proyecto se utilizó una Raspberry Pi 2 Modelo B la cual tiene las siguientes especificaciones:

- Procesador BCM2836 ARMv7 multinúcleo (4 núcleos) de 32 bits con una velocidad de reloj de 900 Mhz
 - 1GB en RAM
 - Conector GPIO de 40 pines (conector extendido)
 - Salida de audio analógico / video analógico de 4 pines
 - 4 Puertos USB
 - Salida de video HDMI
 - Interfaz para cámara y pantalla táctil (venta por separado)
 - Capacidad de alimentar periféricos de alto consumo a través de USB
 - Controlador Ethernet 100 Mbps
 - Funciona a 5V – 1.5A (recomendado)
- Antenas UniFi 802.11 AC AP Mesh para exteriores.
 - Celulares de gama media / baja.
 - Computadoras de laboratorios en escuelas públicas.

Servicios

- Servicio de *Wikis* para escuelas y contenido educativo.
- Servicio videoconferencias
- Servicio de mensajería instantánea
- Servicio de noticias comunales

Información a transmitir

- Imágenes del contenido de los cursos (formato *JSON*)
- Noticias/Anuncios públicos
- Mensajes de grupos de clases
- *Stream* de video

Requisitos

- Conexión inalámbrica. Mantener continuidad en su funcionamiento / Alta disponibilidad.
- Capacidad de prevención contra fallos. (Principalmente eléctricos)
- Capacidad de operar sobre dispositivos de bajos recursos.
- Capacidad de soportar múltiples servicios concurrentemente.

6.6.2. Identificación de potenciales riesgos, vulnerabilidades y la propuesta de medidas correctivas

Risk	Impact	Raw probability	Raw impact	Raw risk rating	Treatment	Treatment cost	Treatment status	Treated probability	Treated impact	Target risk rating	Current risk rating
Robo del equipo físico de las antenas y/o sistemas de suministro de energía	La funcionalidad de la red se pierde total o parcialmente	70%	100%	70%	Poner el equipo dentro de un compartimiento con candado y a una altura fuera del alcance de las personas	\$215	100%	1%	80%	1%	1%
Equipo dañado por tormentas eléctricas	Daño total o parcial del equipo	50%	80%	40%	Proteger el equipo con una cobertura que lo proteja de los fuertes climas.	\$35	100%	10%	16%	2%	2%
Access points falsos / Rogue Aps	Robo o acceso a información de usuarios de la red sin autorización o consentimiento	5%	60%	3%	Implementación de white list entre Ip's de las antenas y escaneos de red constantes.	\$0	100%	1%	16%	0%	0%
Ataque al servidor central	Todos los servicios podrían estar de baja	65%	90%	59%	La implementación de un WAF en el servidor web.	\$1,250	100%	25%	16%	4%	4%
DoS a la red inalámbrica	Alta congestión de tráfico en la red causando denegación de servicio para otros usuarios	5%	10%	1%	Implementación de un balanceador de cargas	\$1,250	100%	10%	0%	0%	0%
Uso inapropiado de la red	Generación de tráfico no deseado para la red	85%	10%	9%	Redireccionamiento de tráfico o bloqueo del mismo en base a whitelist. Definir políticas de uso de la red.	\$0	100%	0%	0%	0%	0%

Cuadro 9: Evaluación de riesgos - IRR

Para mejor evaluación sobre los potenciales riesgos de seguridad y vulnerabilidades de esta red mesh se realizó una tabla de riesgos, impactos y tratamientos también conocido como IRR. En dicha tabla se listaron algunas de los potenciales riesgos identificados son cuales son descritos a continuación:

1. Robo del equipo físico y/o sistemas de suministro de energía

Este riesgo es de suma importancia porque su impacto de increíblemente alto. El riesgo de perder alguno de los dispositivos de la red (APs, servidor o sistemas de suministros de energía) tendría como impacto que la red pudiera quedar parcial o totalmente inhabilitada. El modelo de *red mesh* permite un grado de resiliencia en dado caso algún nodo fallara, pero la pérdida del equipo es un riesgo que no solo incurre en fallas del funcionamiento de la red, sino también incurre en costos extras al reemplazar dichos dispositivos.

Por el contexto en el que se vive actualmente en Guatemala, incurrir en casos de robo o pérdida del equipo es un riesgo con una probabilidad bastante alta. Por esa misma razón, se propuso una alternativa como tratamiento para este riesgo. El tratamiento o mitigación que se propuso para este riesgo consiste en dos partes:

- Colocar el equipo dentro de un compartimiento bajo llave.

- Colocar el equipo a una altura fuera del alcance de las personas.

Como propuesta de un compartimiento, se realizó un modelo de demostración de cómo podría ser un compartimiento de este tipo el cual se puede observar en las figuras 8, 9, 10 y 11 de la sección de anexos de este documento. En el Cuadro 2: Evaluación de riesgos - IRR, se puede observar el costo estimado de la implementación de esta medida correctiva

2. Equipo dañado por tormentas eléctricas

Como sabemos Guatemala se encuentra en una zona sub-tropical, por lo tanto el clima fluctúa bastante entre fuertes soles, lluvias y tormentas. Según un estudio realizado por el sitio Weather Spark, se encontró que en los meses de mayo a octubre la probabilidad diaria de precipitación está por encima del 40 %. [87]



Figura 37: Probabilidad de precipitación en Ciudad de Guatemala

Teniendo esta situación en mente, se consideró el riesgo de que el equipo pudiera dañarse a causa de lluvias o tormentas eléctricas. Por esa misma razón, se propuso que el equipo fuera resguardado dentro de un compartimiento. Dicho compartimiento es el mismo mencionado en el riesgo anterior por el cual el costo que se ve en el Cuadro 2, es un costo compartido.

3. Puntos de acceso falsos o no autorizados

Este riesgo plantea la situación en la cual un usuario o agente malicioso instalará un punto de acceso falso o no autorizado dentro de la red. Esto lo que causa es que clientes que se desean conectar a la verdadera red, terminaran conectándose hacia un punto de

acceso el cual no forma parte de la red. Esto lo que significa es que ese punto de acceso podría ser explotado para obtener información sin autorización o consentimiento de los usuarios.

Para mitigar este riesgo se propuso la implementación y configuración de una *white list* a nivel de dispositivos de la red. Esto lo que permitiría es que la red únicamente considere como nodos los dispositivos especificados en esa lista.

4. Ataques al servidor central

Un riesgo que se tomó en cuenta es riesgo de que el servidor central de la red, que es el encargado de servir las aplicaciones hacia los usuarios, fuera atacado. Este riesgo tendría un grave impacto sobre los servicios de la red porque podría implicar que el servidor quedara deshabilitado.

Para mitigar este riesgo se propuso la implementación de un WAF a nivel del servidor web. Esto lo que permitiría es que los ataques que son realizados a nivel de aplicación sean prevenidos. NGINX ofrece el servicio de un WAF con la adquisición de su versión plus. Por esa razón, el costo de este tratamiento es el costo de la adquisición de una licencia de NGINX Plus. Dicho costo también es compartido con la propuesta de mitigación del siguiente riesgo.

Si se disponen de los recursos monetarios se recomienda utilizar la herramienta propuesta anteriormente, pero en caso no se disponga de estos recursos también existen otras alternativas *open source*. Estas alternativas son gratis pero podrían ser un buen complemento o solución para el riesgo identificado. Algunos de los WAF *open source* que se recomiendan son los siguientes:

- ModSecurity
- IronBee
- NAXSI
- WebKnight

Es importante mencionar que todos los WAF mencionados anteriormente son compatibles con un servidor web NGINX.

5. DoS a la red inalámbrica

Este es un riesgo que está presente en todas las implementaciones de una red. Prácticamente consiste en la saturación de la red mediante el envío de peticiones a una escala que la red por si sola no puede manejar. Este riesgo podría ser causado de manera consciente, es decir un atacante que crea solicitudes en exceso con el fin de saturar la red, como puede ser de manera inconsciente, es decir un alto flujo de usuarios dentro de la red lo que provocaría una saturación en el tráfico de la misma.

Para mitigar este riesgo lo que se propuso fue la implementación de un balanceador de carga. Estos balanceadores de carga pueden implementarse tanto a nivel de software como hardware. En esta propuesta, se recomendó utilizar el balanceador de carga que está incorporado como una de las características que ofrece NGINX Plus. Por esa razón, el costo de implementar un balanceador de carga fue compartido con el riesgo anterior.

Al igual que con el riesgo anterior, a continuación se mencionan algunos balanceadores de carga *open source* en caso no se disponga del recurso monetario para la adquisición del balanceador de carga de NGINX. La contraparte de esto es que el proceso de implementación de estos balanceadores de carga es más complicada y más elaborada. Algunos balanceadores de carga *open source* son:

- Seesaw (<https://opensource.google/projects/seesaw>)
Es un servidor virtual de Linux desarrollado en GO y utilizado por Google para obtener una plataforma robusta y escalable que pueda manejar el tráfico de direcciones IP virtuales. Además, puede realizar chequeos de sanidad contra *backends* y es fácil de manejar. Su código se puede encontrar en <https://github.com/google/seesaw>.
- Neutrino (<https://neutrinoslb.github.io/index.html>)
Es un balanceador de carga reconocido por reputación, al ser utilizado por eBay y con fama de ser eficiente y rápido en términos de procesador y utilización de memoria. Fue desarrollado en Scala con una *pipeline* altamente escalable para que nuevos módulos puedan ser agregados sin trabajo extra. <https://github.com/eBay/Neutrino>
- Balance (<https://www.inlab.net/balance/>)
Es un balanceador de carga desarrollado con su propio stack de IP's, optimizado para balanceador de carga y que puede ser utilizado para cualquier sistema operativo. Utiliza la tecnología de *multithreading* y controladores de entrega de aplicaciones (ADC, por sus siglas en inglés) para crear un concepto de módulo en la aplicación de balanceador de carga.

6. Uso inapropiado de la red

Finalmente, el último riesgo planteado en el IRR es el uso inapropiado de la red. Este riesgo se refiere a que los usuarios de la red utilicen la red para propósitos que no son los planeados. Esto incurriría en una generación de tráfico no deseado a lo largo de la red, así como el consumo de recursos como el ancho de banda de la misma.

Para mitigar este riesgo se propuso que se empleara el redireccionamiento o bloqueo de sitios web mediante otra white list a nivel de red. Esto lo que significa es que se le limitaría al usuario los sitios a los cuales puede acceder. De esta manera se puede asegurar que el usuario no utilice otros sitios más que los permitidos, que en este caso serían los sitios de los servicios montados en la red. Además, como solución complementaria para este riesgo se propuso la definición de las políticas de uso de la red. Estas políticas se le presentan al usuario al momento de que desea conectarse mediante un portal captivo. Esto significa que si un usuario desea conectarse a la red debe aceptar las políticas de uso. Esto se traduce a que el usuario fue informado sobre estas políticas y que en caso contrario podría restringirse el acceso a la red desde ese dispositivo.

De la misma manera, es importante que dentro de las políticas de uso de la red se le den indicaciones al usuario sobre la buena utilización de la red. Es decir, mantenerlo informado y proporcionarle algunos consejos como lo son el no compartir datos personales, no ingresar información sensible dentro de aplicaciones o servicios que no lo requieren, leer las instrucciones antes de utilizar algún servicio, etc.

6.6.3. Cifrado *Wi-Fi*

Uno de los primeros acercamientos que comúnmente se hace hacia las redes WLAN en cuanto a seguridad es el cifrado *Wi-Fi*. Esto lo que permite es que los usuarios que desean conectarse a la red requieran de una contraseña para poder asociar su dispositivo con la red. Además, este mecanismo hace que los paquetes que son enviados dentro de la red estén cifrados bajo un mismo protocolo.

Los protocolos más comunes para este tipo de cifrado son:

- WEP
- WPA
- WPA2

Debido a que esta *red mesh* se trata de una red comunitaria se necesita que la misma sea de fácil acceso. Por lo tanto, se considera que el cifrado *Wi-Fi* no es una alternativa viable porque presenta las siguientes dificultades:

- Se necesita de una contraseña para la asociación de un dispositivo móvil con la red.
- Hacer que cada usuario nuevo a la red ingrese una contraseña volvería el proceso tedioso.
- El distribuir o hacerle llegar la contraseña hacia los demás usuarios implicaría que la misma debe ser pública, por lo tanto, restringir el acceso con este método resulta redundante.

Debido a las dificultades mencionadas anteriormente este acercamiento sobre cifrado *Wi-Fi* no fue tomado en cuenta para el diseño de la red.

6.6.4. Portal captivo

Debido a que el acercamiento sobre cifrado *Wi-Fi* resultaba redundante de implementar, se buscó una alternativa distinta para manejar el acceso a la red. Debido a ello se consideró la opción de implementar un portal captivo. Un portal captivo es utilizado por dos razones:

- Darle trazabilidad a las acciones de un usuario en la red. Esto significa que si por ejemplo el usuario ingresa como credenciales el nombre “Juan” y el apellido “Pérez”, el sistema utilizará esta información para autocompletar campos en donde se requiera del nombre y apellido del usuario. Se tiene presente que un usuario conectándose a la red podría ingresar cualquier información en estos campos, pero eso solo sería un contra beneficio para el usuario debido a que no podría aprovechar la funcionalidad que le ofrece el uso correcto del portal captivo.

- Presentarle al usuario las políticas de uso de la red. Para poder acceder mediante el portal captivo el usuario debe aceptar que leyó y entiende las políticas de uso de la red. Esto es de beneficio porque que de lo contrario podría llegar a restringirse el acceso a la red desde ese dispositivo. Con esto también se deja claro que aunque un usuario no respete las políticas de uso de la red, él fue informado desde el inicio que comenzó a utilizarla.

6.7. Sistema de alimentación de energía

Según los hallazgos realizados en entrevistas con usuarios y técnicos de campo que ejecutan proyectos en comunidades rurales se detectó que el tema de la energía eléctrica es algo bastante complicado. La energía eléctrica en comunidades rurales tiene muchas fluctuaciones y fácilmente puede estar deshabilitada por un par de horas como puede estarlo de 1 a 3 días consecutivos. Por lo tanto, se buscó una manera de suplir esa necesidad. Normalmente, lo primero que se busca para este tipo de problemas es un Sistema de alimentación ininterrumpida, como por ejemplo los UPS. Lamentablemente, la solución de utilizar equipos UPS para las antenas de la red sigue dejando el espacio en blanco en cuanto a cómo obtener la energía eléctrica. Además, un UPS después de que perdió conexión eléctrica normalmente puede continuar funcionando por 2 o 3 horas máximo, lo cual sería un problema en los casos en los que la energía eléctrica se encuentra deshabilitada por varios días. Debido a esto se buscaron alternativas que fueran autosostenibles y que no se vieran afectadas por este problema en particular. Teniendo esto en mente, se buscaron alternativas que fueran basadas en paneles solares. A continuación, se muestra un diagrama sobre el funcionamiento de los paneles solares:

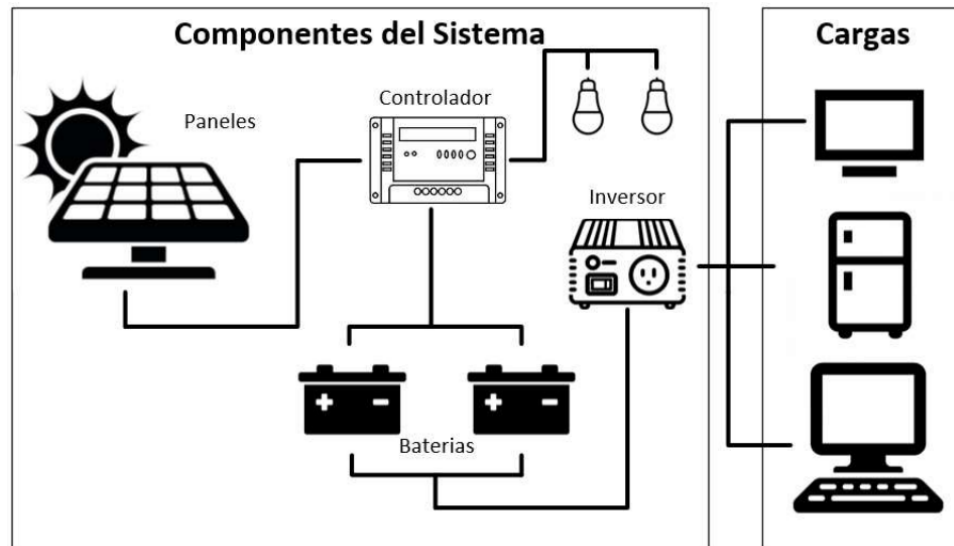


Figura 38: Diagrama de funcionamiento de paneles solares

Se cotizaron 3 distintas alternativas sobre paneles solares con empresas locales con el fin de poder tomar una decisión sobre qué proveedor escoger para este proyecto. Dichas alternativas se describen a continuación:

6.7.1. Solar Guatemala

Se realizó una cotización de un panel solar que soportara tres antenas. La cotización realizada fue de Q.5,750 quetzales por un sistema que tuviera las siguientes especificaciones:

- Panel de 320 Watts
- 4 baterías
- 6m de cable fotovoltaico
- 1 Kit de controlador e inversor

Ventajas

- Kits de fácil instalación, tipo "*plug and play*".
- Soporte técnico para la instalación de las antenas.
- Comunicación fácil y directa con la empresa.

Desventajas

- No hay variedad de tamaño de los paneles.
- Requiere de una inversión inicial alta.

6.7.2. SolarGuat

Al igual que la empresa anterior se cotizó un panel solar que supliera tres antenas y que permitiera una disponibilidad 24/7. La cotización tuvo un precio de Q. 2,825. Dicha cotización incluye lo siguiente:

- Panel de 130 Watts
- 2 baterías
- Controlador de carga de 20A
- 20m de cable fotovoltaico
- Conector MC4 Simple

Ventajas

- Menor costo que la alternativa anterior

- Ofrecen variedad en el tamaño de los paneles solares

Desventajas

- Falta de soporte en cuanto a instalación.
- Únicamente se cuenta con el manual de instalación.

6.7.3. Kingo Energy

Kingo es una alternativa bastante distinta a las mencionadas anteriormente. Es una empresa guatemalteca que ha llevado energía eléctrica a diversos lugares de la región norte del país mediante el uso de paneles solares. Según la página web de Kingo una familia en promedio gasta \$15 en velas y fósforos, con Kingo ese costo se reduce a \$8. [88]

La diferencia con esta empresa radica en que su modelo de negocio es totalmente distinto ya que no venden paneles solares. A diferencia, el servicio es prepago, es decir, el usuario paga solamente lo que consume. Los usuarios de Kingo pagan por el servicio contrario a comprar y adquirir el equipo de los paneles solares. Aunque los usuarios no adquieren el equipo esto permite que el costo de adquirir el servicio sea más accesible. Con esta alternativa lo que se paga es el tiempo que se desea utilizar el sistema. El costo del kit depende de las necesidades que se busque, pero por ejemplo, el kit de Kingo de 15 watts cuesta Q. 110 por mes. Aunque la instalación del equipo no tiene un costo como tal, en ocasiones es necesario dejar un depósito como garantía del equipo. Una vez halla finalizado el periodo de uso del equipo se devuelve y se obtiene el depósito de vuelta.

Para contratar Kingo, dependiendo de la dimensión del proyecto que se quiere implementar, puede ser mediante un contacto directo con Kingo o de lo contrario se debe solicitar en la tienda de la comunidad y adquirir el servicio mediante la compra de saldo en dicha tienda.

Ventajas

- Bajo costo al ser una alternativa "*as a service*"
- No requiere de una inversión inicial muy grande.
- Permite añadir más antenas al sistema de una manera más económica

Desventajas

- Contactarlos fue bastante difícil.
- Inversión continua al no adquirir el equipo permanente.
- Requiere de un administrador que se encargue de realizar los pagos mensuales.

6.8. Acercamiento a las comunidades rurales

El enfoque de este proyecto no es únicamente tecnológico. Uno de los retos a resolver es el acercamiento a las comunidades rurales. Las dificultades que se tienen al llevar un proyecto a una comunidad rural surgen alrededor de varias preguntas que fueron discutidas en un estudio que se realizó con ACUDE, que es la asociación para la cultura y el desarrollo. ACUDE junto con la Fundación Ramiro Castillo Love, participan en la ejecución de programas de alfabetización en comunidades rurales de Guatemala. Tienen más de 20 años de experiencia en la ejecución y planeación de este tipo de proyectos. Tuvimos la oportunidad de conversar con estas representantes de estas dos organizaciones para que nos transmitieran su experiencia en este ámbito.

La técnica de estudio utilizada fue del tipo grupo focal, en el cual participaron técnicos del campo de ACUDE, personal del área de gestión de proyectos de la Fundación Ramiro Castillo Love e integrantes del grupo de megaproyecto. La siguiente guía fue planteada teniendo como base las vivencias y experiencias que nos fueron compartidas.

6.8.1. Identificación de poder dentro de la comunidad.

El primer paso para llevar un proyecto a una comunidad rural comienza con la identificación de un poder o autoridad dentro de esta misma. Normalmente, estas autoridades o representantes comunitarios se encuentran en alguno de los siguiente grupos que se mencionan a continuación:

1. Concejos municipales
2. Alcaldes
3. COCODE
4. Asociación de vecinos
5. Ancianos comunitarios

Generalmente, el primer acercamiento es hacia la municipalidad de la comunidad. Se hacen solicitudes a las respectivas autoridades para obtener el permiso de implementar un proyecto. Esto significa que la organización interesada en implementarlo presenta a dichas autoridades el plan que se quiere llevar a cabo.

En otros casos, el contacto es directo con vecinos representantes de las comunidades o hacia los COCODE. Una gran ventana para el acercamiento hacia estos grupos son las alianzas con otras organizaciones que ya estén realizando proyectos en esa comunidad. Las alianzas con otras organizaciones permiten que, en comunidades en donde ya se ha logrado llegar hacia las personas, el acercamiento sea mucho más fácil. También, facilitan algunos procesos de logística, como lo es el transporte de personal, transporte de mobiliario, seguridad o el propio contacto con la comunidad. Los proyectos en donde no se ha logrado ningún contacto o acercamiento previo, son los más difíciles debido a que requieren de todo el proceso de convencimiento de los posibles participantes. En caso contrario, las comunidades en

donde ya se ha ejecutado un proyecto anteriormente son más fáciles de llevar a cabo. Si una organización se ha armado de prestigio, los mismos habitantes son los que buscan que dichas organizaciones lleven sus proyectos hacia su localidad.

Es con este primer contacto con quienes se debe conversar todos los temas logísticos que requiera el proyecto. Como lo son la ubicación de mobiliario y equipo y la prestación de servicios de seguridad.

Un punto importante que hay que dejar claro con estos representantes son los alcances y limitantes del proyecto. Los beneficios que la comunidad podría obtener con dicha implementación deben de quedar sumamente claros. Una vez se cuente con el permiso o consentimiento por parte de estos líderes comunitarios se puede proceder con el siguiente punto.

6.8.2. Integración del proyecto con la participación de la comunidad.

Para que la ejecución del proyecto sea realizada con mayor facilidad, se debe tomar en cuenta la participación misma de la comunidad para su implementación. En las experiencias previas que ACUDE ha tenido con proyectos educativos, requieren de contratar personal que sea el punto de contacto entre ellos y la comunidad. Este punto es de suma importancia porque es a través de este contacto que se obtendrá retroalimentación sobre la ejecución del proyecto.

Este representante debe ser parte de la región o de la comunidad misma. También, dicho representante es conocedor del área y por lo tanto domina el o los idiomas que se hablen en dicho lugar. Este representante es clave para la ejecución del proyecto porque también es el que funcionará como gestor y monitor del progreso del proyecto. Con este representante es con quien se harán reuniones de evaluación y monitoreo. Otra de las atribuciones de este representante es ser el motivador de los participantes.

6.8.3. Involucramiento y retención de los participantes del proyecto.

Proyectos anteriores han demostrado que el interés de las personas por la tecnología es cada vez más evidente. Los participantes de proyectos de alfabetización han demostrado un genuino interés por aprender sobre temas de tecnología. Esto se debe a que la tecnología presenta un gran atractivo para las comunidades rurales por lo que ellos mismos muestran interés e iniciativa en participar.

Algo sumamente importante de tomar en cuenta es que es más probable que un usuario del proyecto continúe su participación si logra identificar una oportunidad de mejora o beneficio del proyecto. A la implementación de actividades que sean de beneficio para el participante se le conoce como un beneficio económico-productivo. Esto significa que un usuario continuará participando si ve que su participación en el proyecto le genera algún tipo de valor. Este tipo de participación del usuario es de suma importancia para la retención de los usuarios.

Es importante, que el proceso de participación e involucramiento por parte de los parti-

participantes sea un proceso natural. Es decir, aprovechar los recursos que se tienen a la mano y tomar en cuenta las limitantes que podrían encontrar los participantes del proyecto. Un ejemplo de estas limitantes es la distancia. Si un proyecto está siendo ejecutado en una ubicación en donde los participantes tengan que trasladarse grandes distancias puede generar abandono por parte de dichos participantes. Por lo tanto, es importante que, en la medida de lo posible, se eliminen la mayoría de estas barreras.

En proyectos de tecnología, se debe recordar que un punto de atracción para los participantes es que la facilidad de estar más conectado genera aceptación entre las personas, pero que depende del contexto de la comunidad. La tecnología no es para todos en las comunidades rurales, pero quienes aprovechan la oportunidad pueden realizar un aprovechamiento muy grande.

7.1. Sistema de comunicación

Se realizó un servicio de comunicación de dos canales:

- Anuncios publicados por un usuario con rol de administrador, que pueden ser vistos por cualquier persona 31
- Servicio de mensajería instantánea o chat, con diferentes grupos de conversación, implementado con la librería de *web sockets* llamada *Socket.io*. 33

La función de anuncios se comunica con el servidor mediante solicitudes HTTP, mientras que el servicio del chat funciona a través de eventos con *Web Sockets*.

7.1.1. Diseño e implementación de la interfaz gráfica

Para cumplir con el objetivo 2, se buscaba no solo que la interfaz funcionara, sino que esta fuera usable. La usabilidad de la aplicación se demostró mediante pruebas de usabilidad, en las que se destaca que todas las tareas tuvieron una tasa de éxito del 100 %, ya que todos los usuarios pudieron realizar las tareas sin solicitar ayuda. Además, se tuvieron tiempos relativamente bajos para lograr las acciones, con excepción de la tarea 'Crear anuncio', para la cual se tomaron medidas correctivas para facilitar el cumplimiento. De igual forma, en promedio se necesitaron pocos clicks para cumplir.

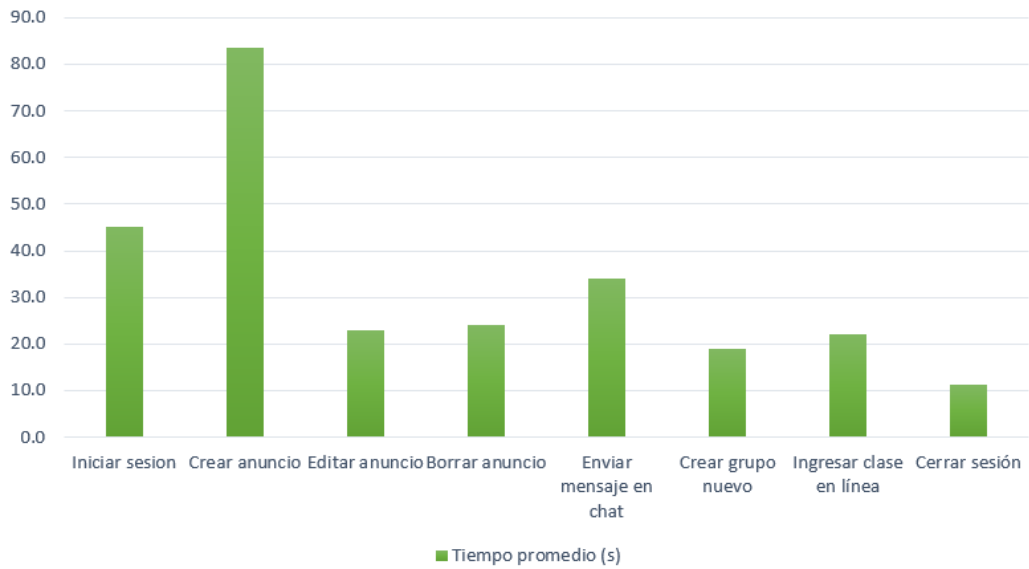


Figura 39: Tiempos promedio para realizar las tareas

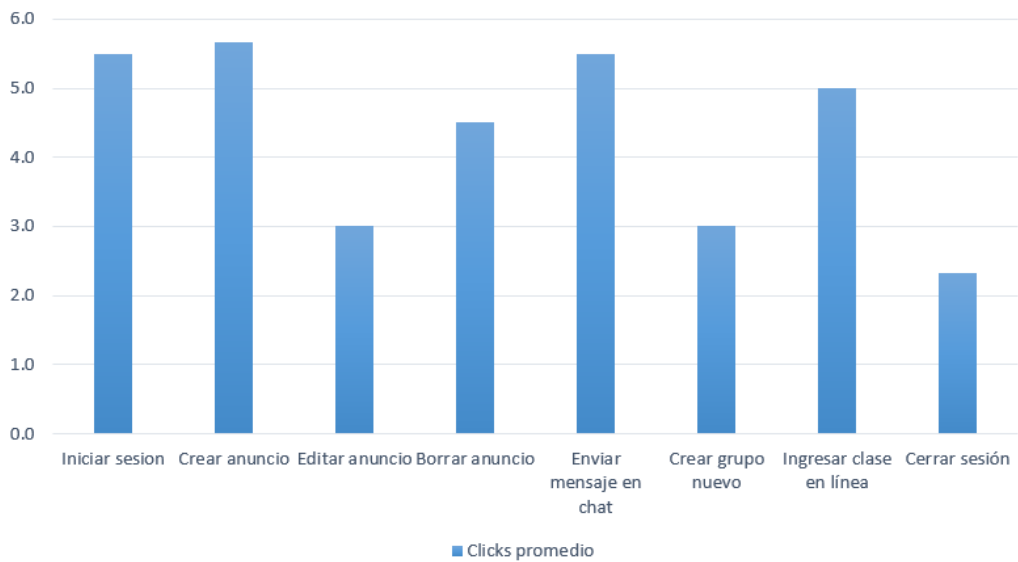


Figura 40: Cantidad de clics promedio para realizar las tareas

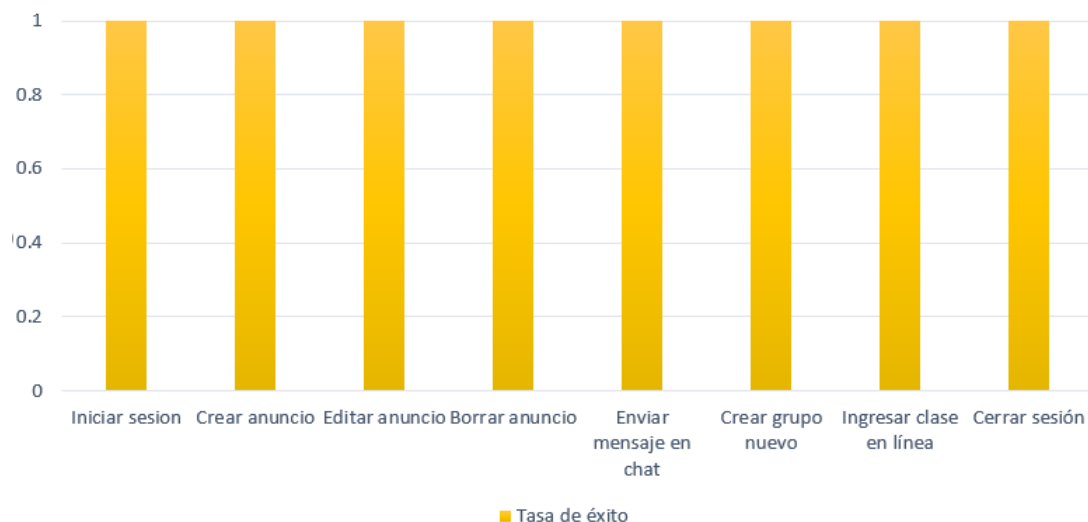


Figura 41: Tasa de éxito para realizar las tareas sin requerir asistencia

7.1.2. Implementación del foro de educación activa

Para el objetivo 3, se buscaba implementar un foro de educación activa, para lo cual se implementó un tercer canal de comunicación: videollamadas. Como se observa en 34, en esta sección se tiene un *stream* de video, en donde, por ejemplo un maestro puede estar dando una clase personal a un alumno, y además se agrega un componente de chat en la misma pantalla para que el alumno pueda escribir sus dudas por ahí también.

Según [12], se logra un foro de aprendizaje activo al incentivar la participación en un ambiente donde el alumno se sienta cómodo, implementando actividades uno a uno y con diferentes formas de comunicación, por lo que al poder acceder a una clase en línea desde donde el alumno prefiera, teniendo una conferencia a solas con el maestro/a y pudiendo escribir sus dudas también por el chat, se logra implementar un canal para usar esta metodología.

7.2. Diseño de protocolo

Los resultados de este proyecto, describen la función de una red tipo malla, este tipo de redes permiten la comunicación constante entre varios nodos, los cuales pueden a su vez establecer comunicación con los usuarios conectados a estos. La red tipo malla, muestra la ventaja de poder darle mantenimiento y sostenibilidad a los nodos que presentan algún problema y por lo cual la hace más efectiva y versátil que una red común.

El objetivo de esta investigación es darle una alternativa de bajo mantenimiento y gran versatilidad, ya que no provee de internet como tal, sin embargo esta garantiza conectividad entre varios usuarios dentro de la comunidad objetivo.

Al mismo tiempo es necesario, contar con un prototipo de protocolo de comunicación

e intervención para las comunidades objetivo, con este protocolo se persigue orientar al investigador con una guía para gestionar un proyecto ante los líderes de las comunidades así como las autoridades políticas y civiles, para obtener una mejor aceptación e interés por parte de estas entidades.

Al mismo tiempo es necesario elaborar un consentimiento informado, el cual consiste en un informe de menor complejidad, el cual es utilizado para fomentar la aceptación del proyecto dentro de la comunidad.

7.2.1. Prototipo red tipo malla

Firmware

Tras realizar la investigación pertinente, se procedió a efectuar una comparación de tres puntos importantes:

1. Compatibilidad de equipo Ubiquity UAP-AC-M
2. Compatibilidad y soporte con diferentes modelos y marcas
3. Libertad de configuración a través de herramientas disponibles

A continuación se muestra una tabla comparativa entre los firmwares 10.

Firmware	Compatibilidad UAP-AC-M	Soporte Genérico	Libertad
Tomatto	No compatible	Poco (29 equipos)	Poca
DD-WRT	Compatible	Medio (814 equipos)	Media
OpenWRT	Compatible	Alto (1545 equipos)	Alta

Cuadro 10: Tabla de comparación entre firmware y características

El soporte genérico de cada *firmware* indica la cantidad de equipos que tienen compatibilidad cada uno de estos, mientras el firmware soporte una mayor cantidad de equipos, es más fácil la integración de diferentes nodos, dentro de la red, aumentando la escalabilidad del proyecto.

En cuanto a libertad, esto indica la facilidad de operar paquetes que no son propios del firmware, utilizado en este proyecto para la instalación y configuración de los diferentes protocolos de enrutamiento.

En el Cuadro 11, se presentan las compatibilidades entre los diferentes *firmwares* y los protocolos de enrutamiento, con esto nos referimos a la libertad del *firmware*.

Con lo cual podemos observar que dentro de los *firmwares* seleccionados para esta investigación, OpenWRT es el firmware con mayor libertad en base a compatibilidad en general.

Firmware	OLSR	B.A.T.M.A.N	Babel
Tomatto	Compatible	Compatible	Compatible
DD-WRT	Compatible	No compatible	No compatible
OpenWRT	Compatible	Compatible	Compatible

Cuadro 11: Cuadro de compatibilidad entre firmware y protocolo de enrutamiento

```

root@OpenWrt1:~# batctl n
[B.A.T.M.A.N. adv openwrt-2019.3-0, MainIF/MAC: mesh0/18:e8:29:a3:7b:d3 (bat0/7a:e8:b3:92:37:
52 BATMAN_IV)]
IF      Neighbor      last-seen
mesh0   18:e8:29:a3:7c:6f  0.800s

```

Figura 42: Se muestran los nodos vecinos dentro de la red malla

Protocolos de Enrutamiento

Se presenta un cuadro con las mediciones de prueba PING en cada protocolo de enrutamiento, entre dos nodos a 100 iteraciones donde se muestran los resultados del tiempo que tarda en responder una antena a un *request* dentro de la red *mesh* Cuadro 12.

Protocolo	Min	Max	Media	Promedio	Paquetes perdidos (%)
OLSR	0.512 ms	2.038 ms	1.324 ms	1.406 ms	0
B.A.T.M.A.N	0.424 ms	1.878 ms	0.468 ms	0.524 ms	0
Babel	0.416 ms	1.404 ms	0.442 ms	0.486 ms	0

Cuadro 12: Tabla comparativa de tiempo de respuesta

En esta prueba demostramos las velocidades de respuesta, en los cuales observamos que el mas rápido en responder y por ende establecer una conexión entre los nodos, es babel seguid de B.A.T.M.A.N y por ultimo OLSR.

Utilizando los análisis de los siguientes artículos científicos se muestra una

7.2.2. Red malla final

Para verificar que la red en malla si esta establecida con el firmware OpenWRT y el protocolo de enrutamiento B.A.T.M.A.N se procedió a utilizar la función `batctl n` en la terminal para mostrar los nodos vecinos dentro de la red malla generada con B.A.T.M.A.N. como se muestra en la Figura 42, esto indica que el nodo en el cual estamos si establece una comunicación con sus nodos vecinos.

De igual forma se puede revisar por medio de la interfaz gráfica Lucí, la cual se instalo previo a la integración de B.A.T.M.A.N. Figura 97.

7.2.3. Acercamiento a las comunidades rurales

El enfoque de este proyecto no es únicamente tecnológico. Uno de los retos a resolver es el acercamiento a las comunidades rurales. Las dificultades que se tienen al llevar un proyecto a una comunidad rural surgen al rededor de varias preguntas que fueron discutidas en un estudio que se realizó con ACUDE, que es la asociación para la cultura y el desarrollo. ACUDE junto con la Fundación Ramiro Castillo Love, participan en la ejecución de programas de alfabetización en comunidades rurales de Guatemala. Tienen más de 20 años de experiencia en la ejecución y planeación de este tipo de proyectos. Tuvimos la oportunidad de conversar con estas dos organizaciones para que nos transmitieran su experiencia en este ámbito.

La técnica de estudio utilizada fue del tipo grupo focal, en el cual participaron técnicos del campo de ACUDE, personal del área de gestión de proyectos de la Fundación Ramiro Castillo Love e integrantes del grupo de megaproyecto. La siguiente guía fue planteada teniendo como base las vivencias y experiencias que nos fueron compartidas.

7.2.4. Prototipo de protocolo de comunicación e intervención para la comunidad objetivo

Tras la investigación pertinente y las reuniones con personal calificado en distintas ONG y personal docente y estudiantes de la Facultad de Ciencias Sociales de la Universidad del Valle de Guatemala, se determino oportuno usar la siguiente estructura de requerimientos para la gestión de proyectos, basada en la tesis "Guía para la gestión de proyectos ante los Consejos de Desarrollo Urbano y Rural"[82] por Otto Rene Rojas Miranda, de la Universidad de San Carlos de Guatemala. Cabe mencionar que cada caso tiene necesidades particulares y podrían necesitar una estructura diferente a la que se plantea a continuación.

1. Marco Jurídico. Corresponde a todas las normativas legales que pudieran afectar el proyecto.
 - a) Marco legal nacional.

Se debe investigar permisos y licencias para el manejo de ondas de transmisión por radio, tanto en la comunidad como en el ministerio de comunicaciones, esto debido al uso de antenas y radio frecuencias que formaran parte de nuestro proyecto, así mismo se debe investigar si existen normativas dentro de la comunidad objetivo.
2. Plan Estratégico. Corresponde a la planificación del desarrollo del proyecto dentro de la comunidad
 - Gestión de proyectos.

Se debe realizar los tramites necesarios para que el proyecto tenga todos los recursos humanos, materiales y financieros para su desarrollo. Estos tramites incluyen:

 - Licencias
 - Permisos
 - Normas municipales

- Entrevistas
- Estudio de la topografía del sector para determinar rangos óptimos para las antenas
- Alimentación eléctrica
- Factibilidad de mantenimiento

3. Identificación de poder dentro de la comunidad objetivo.

Para que el proyecto sea viable es necesaria la comunicación con los líderes políticos, sociales, económicos y religiosos de la comunidad, por lo cual se hace necesario identificar las siguientes autoridades:

- a) Municipio
 - Concejo Municipal
 - Alcalde
- b) Consejos de desarrollo Urbano y Rural
 - COCODDE
 - COMUDE
- c) Participación comunitaria
 - Asociación de vecinos
 - Alcaldía Indígena
 - Consejos asesores indígenas
 - Alcaldía Comunitaria o auxiliar
 - Cabildo Abierto

4. Proceso para la gestión de proyectos ante los consejos de desarrollo .

La gestión económica es necesaria para realizar los proyectos de desarrollo, en este caso, dicho apoyo economic no es necesario, sin embargo como una guía de trabajo se establecen las siguientes normativas para la presentación de proyecto.

- a) Identificación del proyecto
- b) Descripción del proyecto
- c) Justificación del proyecto
- d) Objetivos del proyecto
- e) Diagnostico y caracterización del área de influencia
- f) Aspectos geográficos
- g) Aspectos socioeconómico
 - Población
 - Beneficiarios directos e indirectos
 - Aspectos demográficos
 - Características económicas
- h) Aspectos socioeconómico cualitativos
 - Educación
 - Salud

- Servicios públicos existentes
 - Medios de comunicación
 - Cultura
 - i)* Objetivos y metas
 - j)* Estudio de mercado
 - k)* Estudio Técnico
 - localización del proyecto
 - Área de influencia
 - Tecnología
 - Presupuesto detallado
 - Cronograma de ejecución
 - l)* Seguimiento de proyecto
5. Presentación del proyecto hacia los líderes comunitarios.

La presentación del proyecto hacia los líderes comunitarios es de suma importancia, debido a que estas personas son influyentes dentro de la comunidad, se facilita la aceptación del proyecto por parte de la misma. Con el apoyo de estos líderes y de la comunidad en general es más fácil proceder a su ejecución y a fomentar el uso del servicio. Para lograr una mejor introducción se presenta la siguiente normativa:

- a)* Consentimiento informado.
- Consiste en una carta de introducción del proyecto, la cual debe elaborarse de forma concisa, la cual consiste en la aceptación por escrito de la ejecución y uso del proyecto por parte de la comunidad. Esta carta debe ser firmada por la persona o entidad que otorga el consentimiento. A continuación se presentan los lineamientos de la propuesta (ver anexo 103, 104, 105, 106).
- Propuesta de participación conjunta
 - Propósitos
 - Criterio de selección
 - Duración
 - Técnicas y procedimientos
 - Posibles riesgos e inconvenientes
 - Beneficios
 - Compensación
 - Contactos esenciales

7.3. Diseño de la interfaz gráfica de la aplicación educativa

7.3.1. Resultados del prototipado

A continuación, se explica a detalle el desarrollo de cada uno de los requisitos que fueron definidos en el capítulo de Metodología y cómo fueron cambiando, paralelo a la aplicación de principios de interacción humano-computador y previo a la ejecución de las pruebas de usabilidad.

Herramienta para crear *markdowns*

La Figura 43 muestra la primera versión de la herramienta para crear archivos *markdown*. En un principio, se había creado un único editor de texto en el que se ingresaba toda la información, incluyendo la sintaxis de *markdown*, cuya sintaxis se encuentra descrita en el Marco Teórico.

Luego de consultar el libro de Ritter que detalla el diseño centrado en las personas [43], se tomó en cuenta el aspecto cognitivo del *framework* ABCS, en cuanto a la cantidad y complejidad de funciones mostradas. Se consideró que los botones que permiten ingresar signos de numeral en una parte del editor podrían ser confusos para el usuario, puesto que al ser estudiantes de escuelas existe una alta probabilidad que no hayan tenido contacto con algún editor de *markdown* anteriormente, por lo que esta sintaxis para los títulos y subtítulos podría llegar a ser contraproducente.

Esto se resolvió retirando dichas funciones y rediseñando el editor en varios editores más pequeños, los cuales representan distintas secciones del documento. En la parte superior de la vista, se puede ingresar el título principal del documento (ver Figura 44), el cual es sustituido por su equivalente en la sintaxis de *markdown* (#). Abajo de este título, aparece un bloque con las secciones del documento, en las cuales se puede ingresar un título para la sección y su contenido en un cuadro de texto. El título es también sustituido por su equivalente en la sintaxis de *markdown* (#). También se permite agregar secciones nuevas y eliminar secciones ya existentes (ver Figura 45).

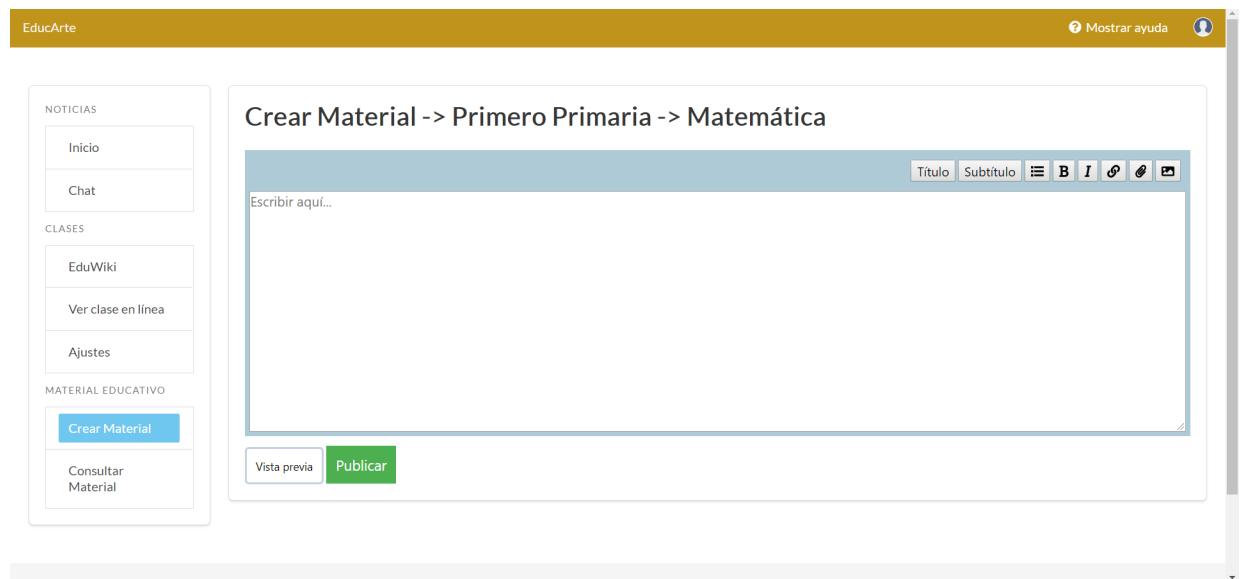


Figura 43: Primera versión del editor de documentos *markdown*

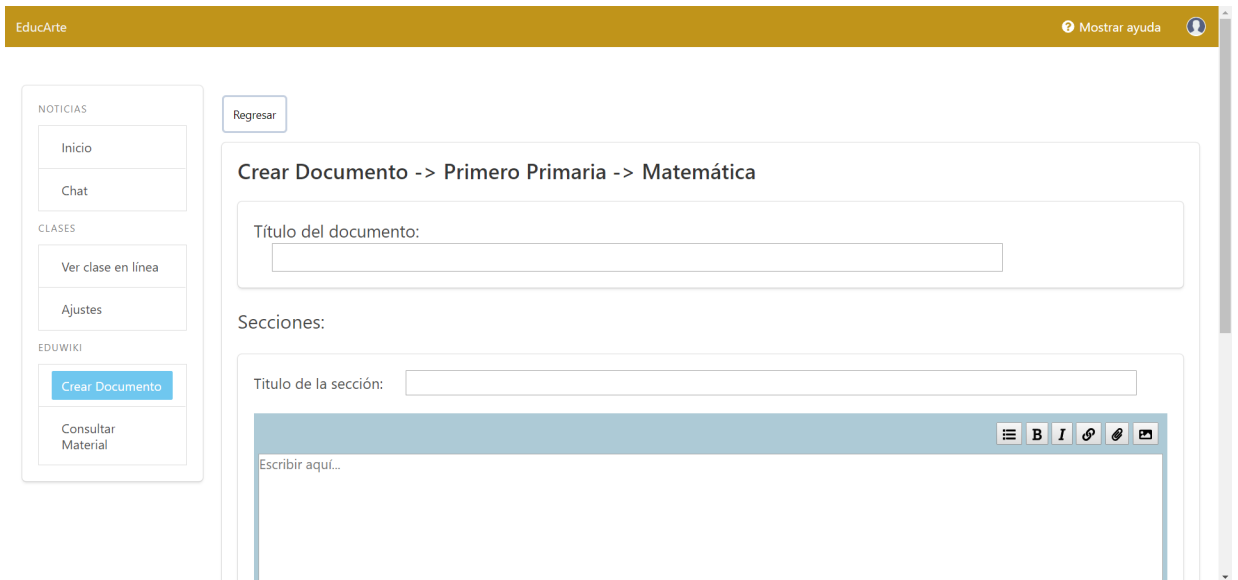


Figura 44: Segunda versión del editor de documentos *markdown*: título del documento

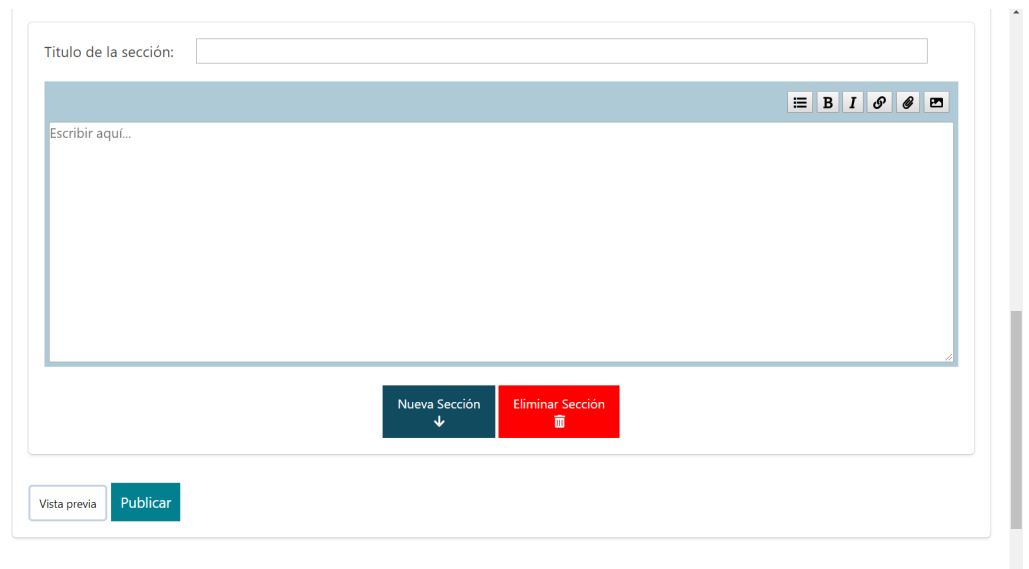


Figura 45: Segunda versión del editor de documentos *markdown*: secciones del documento

Bajo el mismo aspecto cognitivo del *framework* ABCS, se consideró que la cantidad de funciones mostradas en el editor seguían siendo demasiadas para lo que el usuario realmente requería para cumplir con el objetivo de la herramienta: crear documentos educativos. Se retiraron primero los botones para ingresar texto en negrita e itálica y el botón para ingresar listados, con el objetivo de reducir complejidad en la interfaz. Por último, se retiró el botón para ingresar hipervínculos a páginas externas, puesto que se descubrió durante el desarrollo de los otros componentes que la librería utilizada para parsear los documentos *markdown* sustituye los hipervínculos por su equivalente en sintaxis *markdown* (llaves para el texto y paréntesis para la URL). Estos cambios pueden ser vistos en la Figura 46.

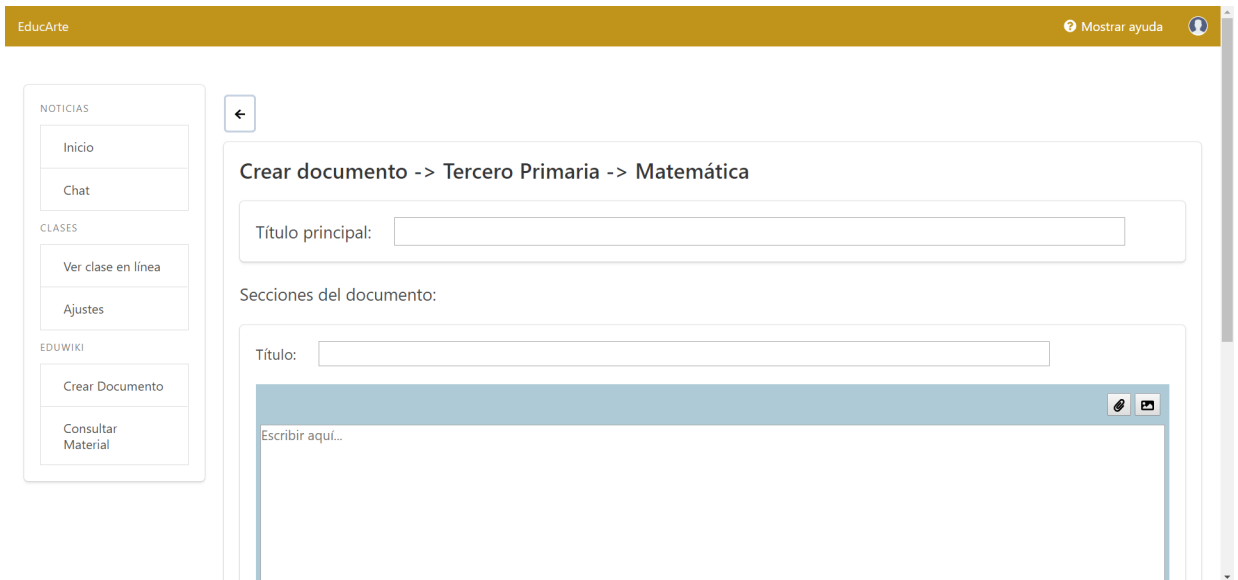


Figura 46: Tercera versión del editor de documentos *markdown*

Visualización del documento previo a su publicación

Al hacer click en el botón "Vista Previa", en la parte inferior del editor de *markdown*, se puede ver cómo se presentará el documento cuando ya esté publicado. La primera versión de esta vista previa puede verse en la Figura 47.

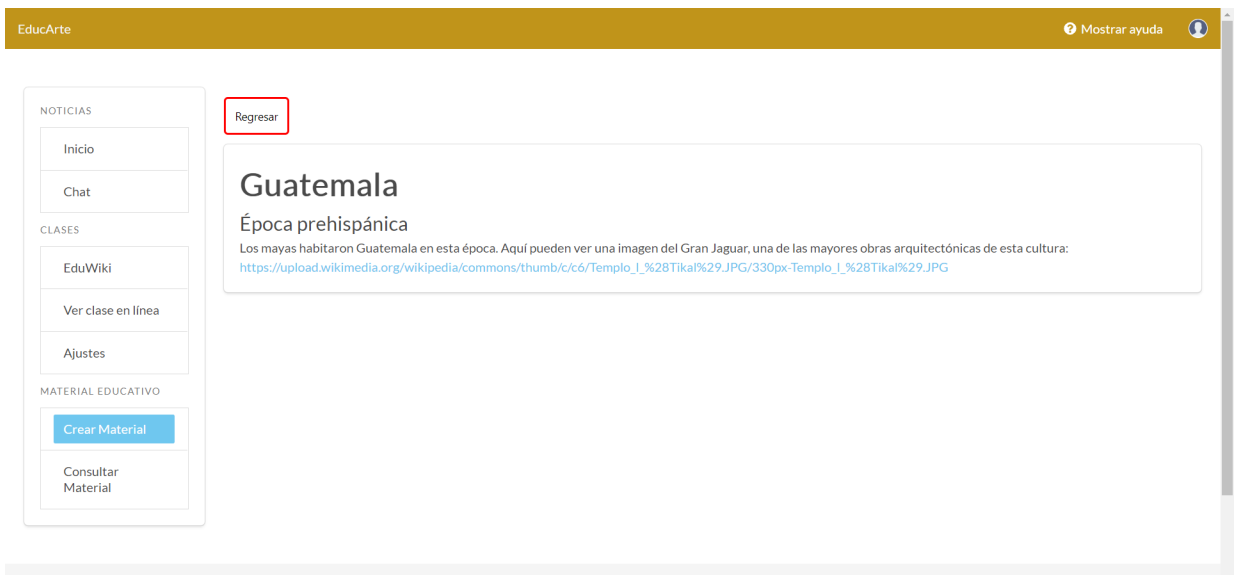


Figura 47: Primera versión de la vista previa de documentos

Para la segunda versión, se tomó en cuenta el aspecto de comportamiento del *framework* de ABCS para implementar un cambio pequeño en el diseño de esta sección. Se insertó automáticamente un separador entre el título principal y la primera sección, y también entre las mismas secciones. Esto facilita al lector del documento a diferenciar entre las distintas

partes del documento, y que no lo vea como una única sección. Estos cambios pueden verse en la Figura 48.

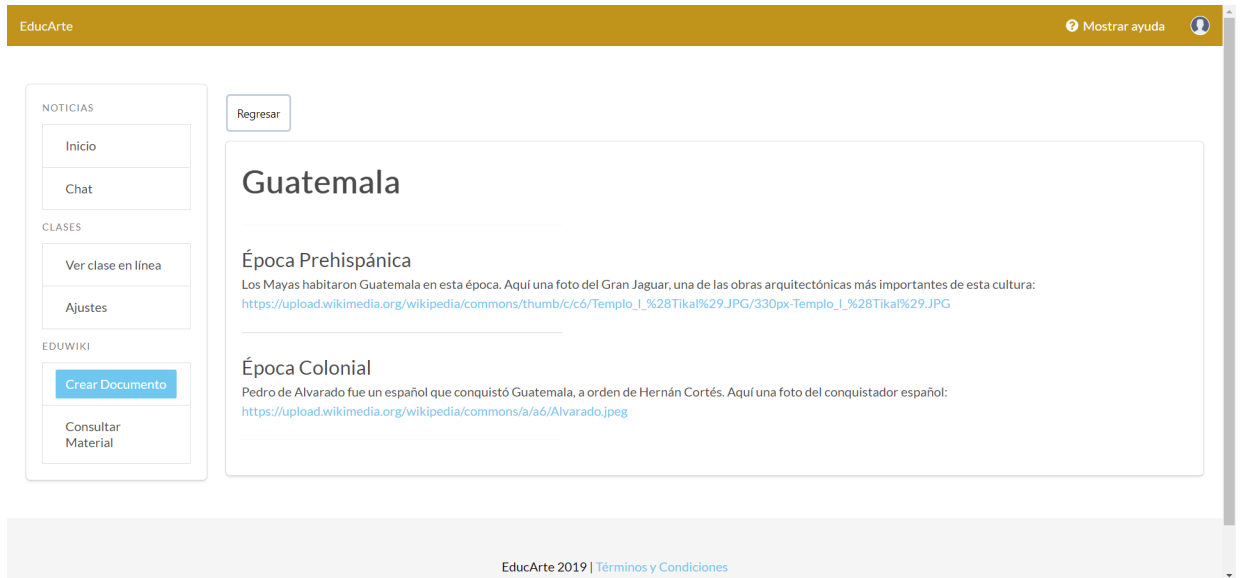


Figura 48: Segunda versión de la vista previa de documentos

Para la tercera versión, el primer cambio que se realizó tuvo una carga funcional y no tan relacionada a la usabilidad del prototipo. Cuando se ingresa un hipervínculo hacia una imagen externa, este es sustituido por su equivalente en *markdown* (corchetes para la descripción y paréntesis para la URL, precedidos por un signo de exclamación '!'), esto para que la imagen pueda ser visualizada en esta sección. Por otro lado, se cambió el texto del botón que se utiliza para regresar al editor *markdown* de 'Regresar' a 'Seguir editando', puesto que el primer texto es el mismo que se tiene en el editor *markdown*, lo cual ayuda a reforzar la cuarta heurística descrita en el marco teórico: Consistencia y estándares. Los cambios pueden verse en la Figura 49.

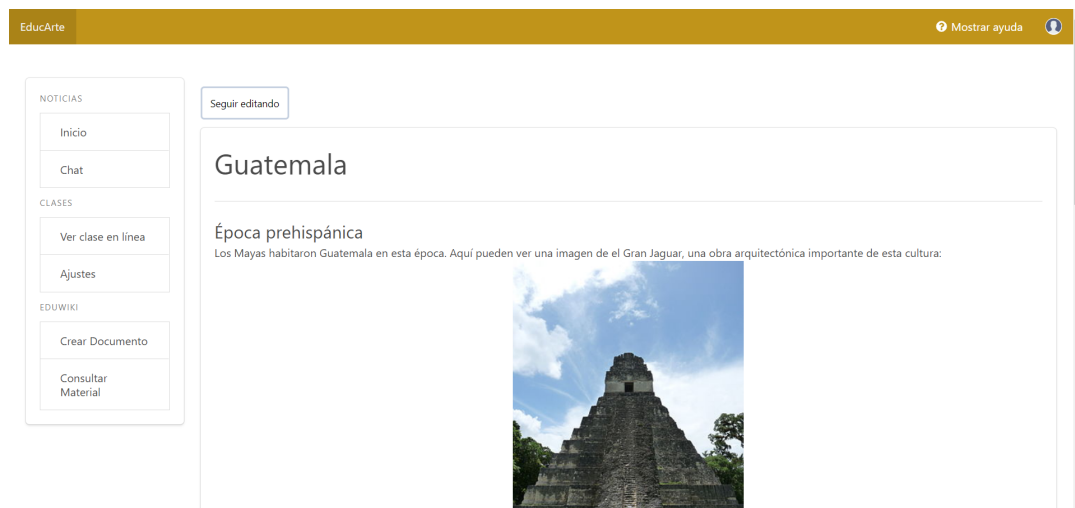


Figura 49: Tercera versión de la vista previa de documentos

Navegación en la wiki

La primera versión del menú para seleccionar un grado y clase para generar un documento nuevo puede verse en la Figura 50.

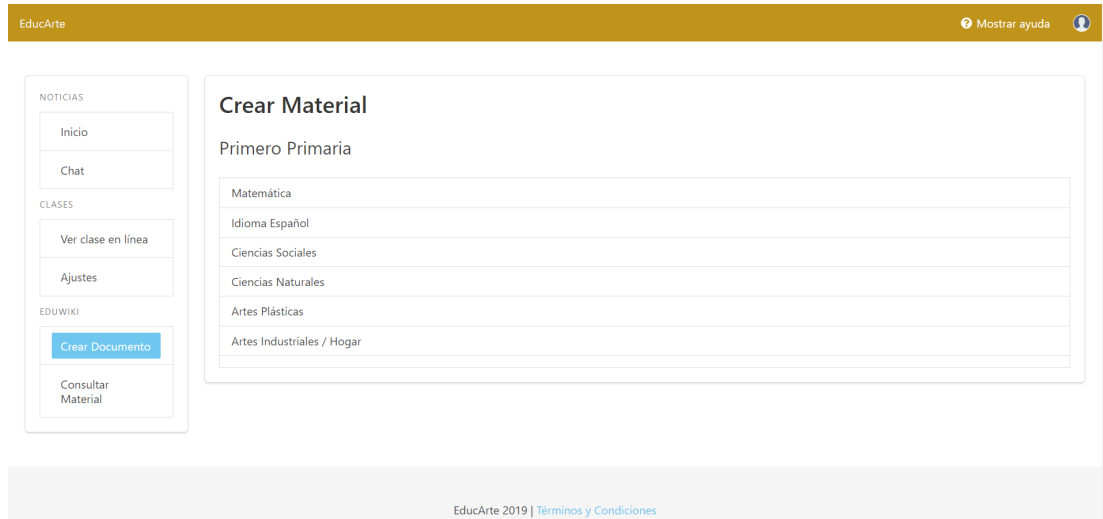


Figura 50: Primera versión del menú principal de la Wiki

Conforme se agregaban más grados, se percató que para el usuario final sería desplazarse hacia abajo para seleccionar los grados que se encontraban en la parte inferior. Tomando en cuenta el aspecto de comportamiento del framework ABCS, se sustituyó este listado de bloques por una caja para seleccionar el grado, y que únicamente las clases que pertenecen a este grado se desplieguen en ún solo bloque. Esto se hizo para representar el mismo orden por medio del cual se navegaba antes para escoger un grado y clase, colocando la caja para seleccionar grado al principio de la vista. Este cambio puede verse en las Figuras 51 y 52.

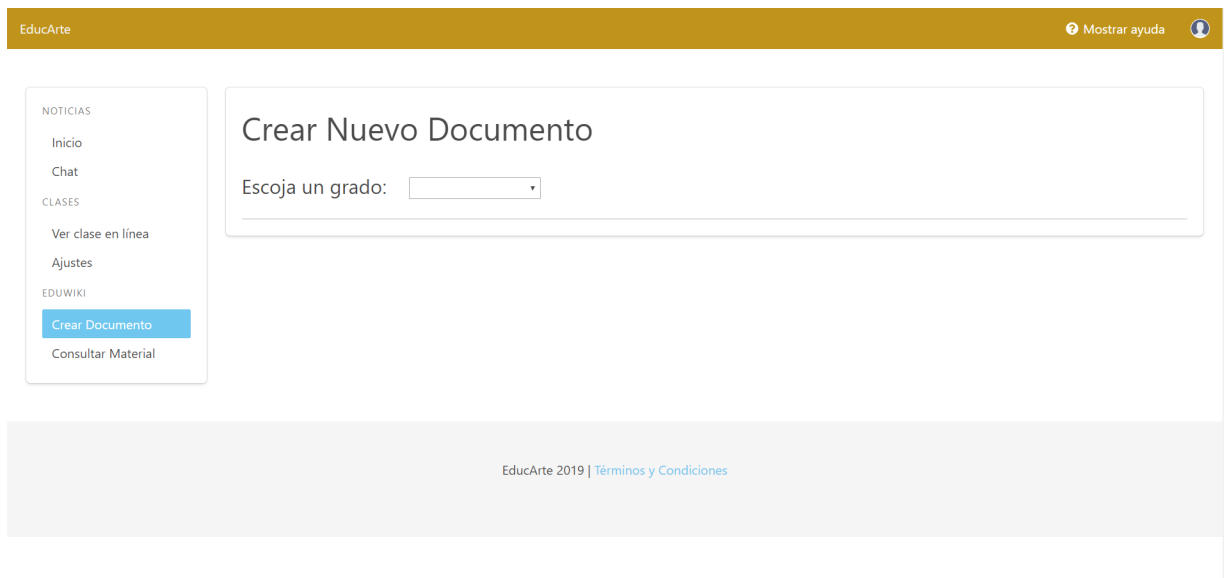


Figura 51: Segunda versión del menú principal de la Wiki

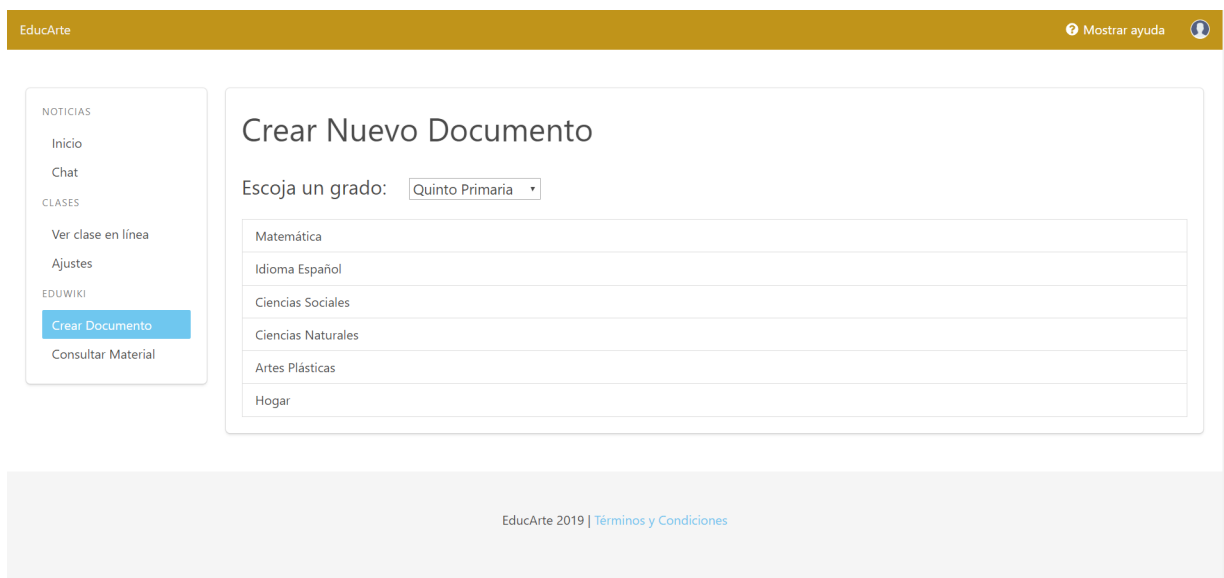


Figura 52: Segunda versión del menú principal de la Wiki

Al seleccionar una de las clases que pueden verse en la Figura 52, se accede a otra vista en la que se puede seleccionar uno de los documentos ya publicados. Como se puede ver en la Figura 53, aparece el listado con un botón grande para visualizar el documento, y un botón para editar el documento en la parte derecha del listado. El botón para editar únicamente debe aparecer si el usuario que está viendo el menú tiene permisos de edición, lo cual es parte de la integración con la capa de *backend*.

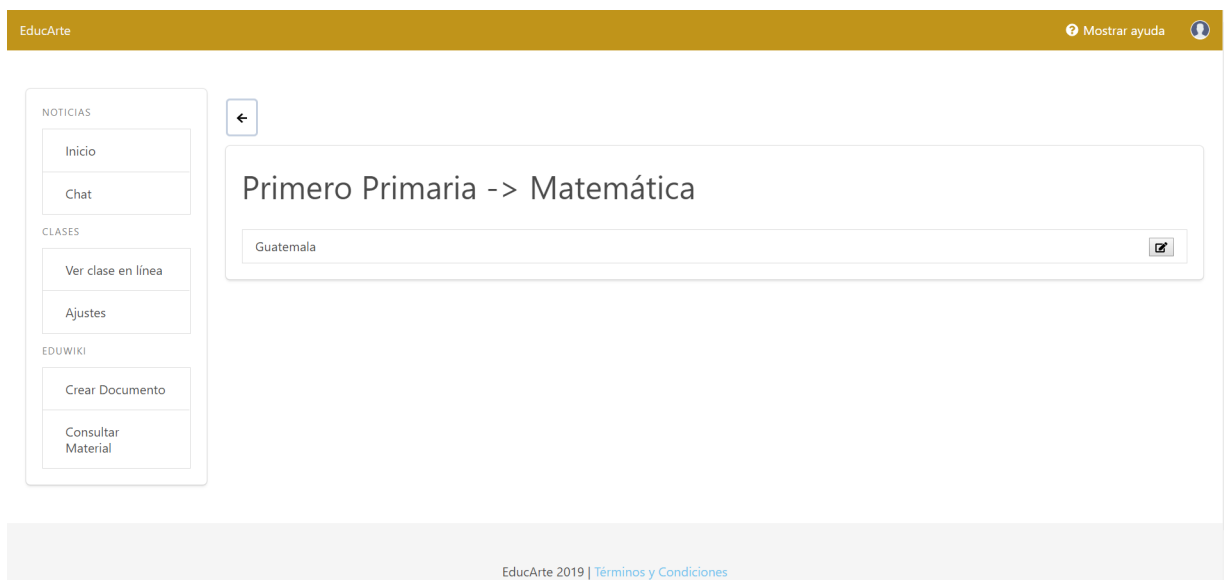


Figura 53: Menú para seleccionar un documento para su visualización o edición

Las vistas para visualizar y editar los documentos tienen el mismo formato que las vistas que se muestran en las Figuras 46 y 49, las cuales corresponden al editor de documentos *markdown* y a la vista previa de documentos, correspondientemente.

7.3.2. Resultados de las pruebas de usabilidad

Los datos que se muestran a continuación resumen el resultado de las pruebas realizadas a todos los usuarios. Si se desea ver a mayor detalle el resultado para cada usuario evaluado, se puede consultar los anexos de este documento.

Cuadro 13: Completación de las tareas

Tarea	Tasa de completación
1	100 %
2	100 %
3	100 %
4	100 %
5	100 %
6	100 %
7	75 %
8	100 %
9	100 %

El cuadro 11 muestra que la tasa de completación fue del 100 % para todas las tareas, exceptuando para la tarea número 7, que consistía en acceder a la vista previa del documento. La tasa para esta tarea fue del 75 %, lo que significa que un usuario no fue capaz de completar esta tarea. El principal problema aquí se produjo porque el usuario no logró identificar en

dónde se encontraba el botón de vista previa al estar en la parte superior del editor, puesto que el botón se encuentra al final del editor, al lado del botón para publicar el documento.

Cuadro 14: Estadísticas del tiempo invertido por los usuarios para realizar cada tarea (segundos)

Tarea	Promedio	Desviación estándar
1	10.50	1
2	11.25	4.71699057
3	40.25	13.8894444
4	27.75	5.12347538
5	6.25	1.70782513
6	7.75	2.62995564
7	10.50	15.0222058
8	3.00	1.15470054
9	3.25	1.25830574

Cuadro 15: Comparación de tiempos (segundos) entre usuarios y el desarrollador

Tarea	Promedio	Tiempo que tomó al desarrollador	% de diferencia
1	10.50	8	31 %
2	11.25	7	61 %
3	40.25	28	44 %
4	27.75	24	16 %
5	6.25	5	25 %
6	7.75	5	55 %
7	10.50	3	250 %
8	3.00	2	50 %
9	3.25	3	8 %

El cuadro 12 muestra el promedio y desviación estándar del tiempo que tomó completar cada tarea. Llama la atención las desviaciones de las tareas 3 y 4, que fueron significativamente altas (arriba de 5), sin embargo se concluyó que estas diferencias de tiempos se debían a los distintos niveles de experiencia en mecanografía que los usuarios manejaban. Esto debido a que las tareas 3 y 4 consistían en redactar títulos y oraciones, y el tiempo que tomó esta redacción, en todos los casos, abarcó la mayoría del tiempo que duró la tarea. El valor alto de la desviación para la tarea 7 se debe al dato atípico provocado por el usuario que no logró completar esta tarea, lo cual fue explicado arriba.

El cuadro 13 muestra una comparación de estos tiempos con el tiempo que le tomó al desarrollador del prototipo realizar las tareas. Como se mencionó en la sección anterior, se considera que un 70 % de éxito es aceptable para una métrica. Se consideró una tarea como exitosa si la diferencia entre ambos tiempos era menor a este número. La única tarea que no cumplió con esta condición fue la tarea 7, de igual manera, porque no pudo ser completada por un usuario. La segunda tarea con mayor porcentaje de diferencia fue la tarea 2, que consistió en cambiar el grado en el que se estaba creando el documento. Los usuarios lograron completar la tarea, pero tardaron en identificar el botón para regresar a la vista anterior al editor.

Cuadro 16: Cantidad de pasos que tomó cada tarea

Tarea	Promedio	Pasos que tomó el desarrollador
1	4.00	4
2	4.00	4
3	6.00	6
4	5.00	5
5	2.00	2
6	1.00	1
7	1.75	1
8	1.00	1
9	1.00	1

Por ultimo, el cuadro 14 muestra una comparación entre la cantidad de pasos que le tomó a los usuarios realizar las tareas y la cantidad de pasos que le tomó al desarrollador. Con estos datos, se logra ver que no existe una diferencia significativa entre ambos números, a excepción de la tarea 7 que, como ya se explicó anteriormente, difiere por el usuario que no logró completarla.

Adicional a estos datos cuantitativos, también se escuchó a los comentarios que algunos de los usuarios que participaron en las pruebas brindaron al terminar de realizar las pruebas. En general, los comentarios recibidos fueron positivos: los usuarios indicaron que la idea detrás de la aplicación les resultaba útil para las comunidades que atendían, además que sentían que el prototipo era entendible y usable.

Únicamente se obtuvo una crítica en cuanto a la vista previa de documentos: muchas de las personas que participan en el programa educativo tienen una edad avanzada, y por esto mismo en muchas ocasiones tienen problemas visuales. El usuario recomendó que se ampliara el tamaño de la letra de esta vista y de la vista en la que se visualizarán documentos ya publicados, puesto que consideraba que el tamaño no era lo suficientemente grande para que no entrara en conflicto con la condición visual de estos usuarios.

7.4. Integración del sistema de comunicación con la interfaz gráfica de la aplicación educativa

7.4.1. Creación, modificación, eliminación y recuperación de anuncios

Una de las partes del sistema de comunicación es un sistema de anuncios en donde se puedan compartir anuncios de relevancia para la comunidad. A través de la RESTful API, el cliente es capaz de crear, modificar, eliminar y recuperar los anuncios mediante las siguientes rutas:

```

1      {
2          "status": 200,
3          "data": {
4              "news": [
5                  {
6                      "id": "ID de la noticia",
7                      "title": "Título de la noticia",
8                      "content": "Contenido de la noticia",
9                      "userId": "ID de usuario que creó la noticia",
10                     "createdAt": "Fecha de creación de la noticia",
11                     "lastUpdated": "Fecha de última modificación"
12                 },
13                 {},
14                 ...
15             ]
16         }
17     }

```

Figura 54: Respuesta de servidor a la solicitud GET de anuncios.

```

1      {
2          "data": {
3              "title": "Titulo de la noticia",
4              "content": "El contenido de la noticia",
5              "userId": "El ID del usuario que creo la
6                  noticia"
7          }
8      }

```

Figura 55: Cuerpo de la solicitud POST de anuncios.

HTTP GET

Con esta solicitud se busca recuperar todos los anuncios a desplegar en la pantalla respectiva de anuncios. Por ser una solicitud de tipo GET, el cuerpo va vacío y la respuesta del servidor se ve reflejada en la Figura 54.

HTTP POST

Esta solicitud permite al cliente poder agregar un nuevo anuncio. Para esta solicitud es necesario que el cliente envíe un cuerpo con los campos demostrados en la Figura 55.

En el caso que la solicitud se haya procesado de manera correcta y no hubiera ocurrido error alguno, la respuesta del servidor será como en la Figura 56. En caso que un error ocurra, la respuesta del servidor será como en la Figura 57.

```
1      {
2          "status": 200,
3          "message": "OK"
4      }
```

Figura 56: Respuesta de servidor a las solicitudes POST, PUT y DELETE de anuncios en caso de haberse procesado exitosamente.

```
1      {
2          "status": 404,
3          "message": "ERROR"
4      }
```

Figura 57: Respuesta de servidor a las solicitudes POST, PUT y DELETE de anuncios en caso de haber ocurrido un error.

HTTP PUT

La solicitud al método PUT permite al cliente modificar un anuncio basado en su ID único, proporcionado al momento de hacer una solicitud del método GET. El ID del anuncio se requiere como parámetro en el URL. Al igual que la solicitud del método POST, el método PUT requiere un cuerpo, necesario para referenciar los campos a modificar. El cuerpo del método está plasmado en la Figura 58.

En caso que la solicitud se lleve a cabo exitosamente, la respuesta del servidor será como en la Figura 56. Si hubo un error al procesarse la solicitud, la respuesta del servidor será como en la Figura 57.

HTTP DELETE

La solicitud del método DELETE permite al cliente borrar un anuncio basado en su ID único, el cual se proporciona al momento de realizar una solicitud del método GET. El ID del anuncio se requiere como parámetro en el URL. Este método no requiere de un cuerpo al momento de realizar la solicitud y la respuesta del servidor, en caso de ser exitosa, se muestra en la Figura 56. En caso de haberse producido un error al momento de procesarla, el mensaje del servidor será como en la Figura 57.

```
1      {
2          "data": {
3              "title": "Titulo de la noticia",
4              "content": "El contenido de la noticia"
5          }
6      }
```

Figura 58: Cuerpo de la solicitud PUT de anuncios.

```

1      {
2          "status": 200,
3          "data": {
4              "chatrooms": [
5                  {
6                      "name": "Nombre de la sala",
7                      "namespace": "ID de la sala"
8                  },
9                  {},
10                 ...
11             ]
12         }
13     }

```

Figura 59: Respuesta del servidor a la solicitud GET de salas de mensajería instantánea.

```

1      {
2          "data": {
3              "name": "Nombre de la sala"
4          }
5      }

```

Figura 60: Cuerpo de la solicitud POST de salas de mensajería instantánea.

7.4.2. Creación, eliminación y recuperación de salas para mensajería instantánea

Para que los usuarios sean capaces de comunicarse en tiempo real, se implementó un espacio en la aplicación que tuviera salas de mensajería instantánea. A través de la RESTful API, el cliente es capaz de crear y eliminar salas, además de recuperar todas las salas ya existentes. Dichas acciones se logran con los siguientes métodos:

HTTP GET

Con esta solicitud, el cliente es capaz de recuperar todas las salas de mensajería instantánea ya existentes. La respuesta del servidor se ve en la Figura 59.

HTTP POST

De igual forma, el cliente es capaz de crear nuevas salas de mensajería instantánea por medio de una solicitud del método POST. La solicitud debe llevar un cuerpo como en la Figura 60, únicamente especificando el nombre a mostrar de la sala. El servidor responderá con un mensaje reflejando que fue realizada con éxito, Figura 61, o que se produjo un error, Figura 62.

```
1      {
2          "status": 200,
3          "message": "OK"
4      }
```

Figura 61: Respuesta de éxito del servidor a las solicitudes POST y DELETE de salas de mensajería instantánea.

```
1      {
2          "status": 404,
3          "message": "ERROR"
4      }
```

Figura 62: Respuesta de error del servidor a las solicitudes POST y DELETE de salas de mensajería instantánea.

HTTP DELETE

El cliente es capaz de eliminar una sala, sabiendo su ID. Es necesario que el cliente sepa el ID de la sala y lo introduzca como parámetro en el URL para poder llevar a cabo la acción. El ID de cada sala se obtiene al realizar una solicitud por medio del método GET. La respuesta de éxito y error del servidor se pueden ver en la Figura 61 y Figura 62, respectivamente.

7.4.3. Creación, eliminación y recuperación de salas para retransmisión de audio y video

Las salas para la retransmisión de audio y video están separadas de las salas de únicamente mensajería instantáneas, sin embargo, las salas de retransmisión de audio y video también contienen su propio ambiente para mensajería instantánea. De nuevo, el cliente es capaz de recuperar estas salas, además de crear y eliminar por medio de la RESTful API. Los métodos implementados fueron los siguientes:

HTTP GET

A través de esta solicitud el cliente es capaz de obtener todas las salas de retransmisión de audio y video ya existentes. La respuesta del servidor se muestra en la Figura 63.

HTTP POST

El cliente es capaz de crear nuevas salas por medio de una solicitud del método POST. Dicha solicitud debe llevar un cuerpo indicando el nombre de la sala, como se puede observar en la Figura 64. De haber realizado la solicitud con éxito, el servidor responderá de la

```

1      {
2          "status": 200,
3          "data": {
4              "videorooms": [
5                  {
6                      "name": "Nombre de la sala",
7                      "namespace": "ID de la sala"
8                  },
9                  {},
10                 ...
11             ]
12         }
13     }

```

Figura 63: Respuesta del servidor a la solicitud GET de salas de retransmisión de audio y video.

```

1      {
2          "data": {
3              "name": "Nombre de la sala"
4          }
5      }

```

Figura 64: Cuerpo de la solicitud POST de salas de retransmisión de audio y video.

siguiente forma, Figura 65, de lo contrario, en caso se hubiera producido un error, la respuesta será de la siguiente manera Figura 66.

HTTP DELETE

De igual forma, el cliente puede eliminar una sala de retransmisión de audio y video por medio de una solicitud del método DELETE. Para esto es necesario que el cliente sepa el ID de la sala y enviar dicho ID como parámetro en el URL de la solicitud. En caso que la sala se haya borrado de manera exitosa, el servidor dara la siguiente respuesta Figura 65, de lo contrario, el servidor devolverá un mensaje de error como se muestra en la Figura 66.

```

1      {
2          "status": 200,
3          "message": "OK"
4      }

```

Figura 65: Respuesta de éxito del servidor a las solicitudes POST y DELETE de salas de retransmisión de audio y video.

```

1      {
2          "status": 404,
3          "message": "ERROR"
4      }

```

Figura 66: Respuesta de error del servidor a las solicitudes POST y DELETE de salas de retransmisión de audio y video.

```

const sala = io.of('/chatroom')
sala.on('connection',
    (socket) => {
        // Usuario conectado
    });

```

Figura 67: Evento de conexión a una sala en específico.

7.4.4. Funcionalidad del sistema de mensajería instantánea

Para la implementación del sistema de mensajería instantánea se utilizó la tecnología de WebSockets, este protocolo permite una comunicación bidireccional a través de un socket TCP. Esta tecnología de comunicación, junto con la arquitectura de Node.js basada en eventos, permite que la mensajería entre usuarios aparente ser casi instantánea, es decir, la latencia entre el envío de un mensaje de un usuario y la captura del mismo por otro usuario, es baja.

Esta tecnología se logró a través de la librería Socket.io, la cual se implementó tanto del lado del cliente como del lado del servidor. Esta permite crear eventos que escuchen por señales que se envíen del otro lado del socket. Esto significa que tanto del lado del servidor como del lado del cliente, se implementan eventos que escuchen por señales, como el envío de un mensaje nuevo; y al mismo tiempo, se implementa el envío de señales.

Al mismo tiempo, esta librería permite la creación de salas o, como ellos lo definen, *namespaces*; que permiten la emisión de señales a todos los usuarios asignados a una de ellas. Esto facilita con la interacción entre usuarios pertenecientes a una misma sala. Los eventos desarrollados del lado del servidor son escuchar nuevas conexiones a las salas, Figura 67, y de escuchar nuevos mensajes de los clientes, Figura 68. Al escuchar nuevos mensajes, el servidor automáticamente emite una señal a toda la sala sobre el nuevo mensaje.

```

socket.on('mensaje', (data) => {
    io.of('/chatroom').emit('NuevoMensaje', data)
})

```

Figura 68: Evento para escuchar nuevos mensajes y la señalización de este al resto de la sala de mensajería instantánea.

```

const sala = io.of('/videoroom')
sala.on('connection', (socket) => {
  if (primerClienteEnLaSala()) {
    socket.emit('streamer', {isStreamer: true})
  } else {
    socket.emit('streamer', {isStreamer: false})
  }
});

```

Figura 69: Evento para detectar nuevos clientes y su respectiva señalización si es o no el transmisor.

```

socket.on('mensaje', (data) => {
  io.of('/videoroom').emit('NuevoMensaje', data)
})

```

Figura 70: Evento para escuchar nuevos mensajes y la señalización de este al resto de la sala de retransmisión de audio y video.

7.4.5. Funcionalidad del sistema de retransmisión por audio y video

Para el sistema de retransmisión de audio y video se utilizó la tecnología de WebRTC. Esta tecnología permite la conexión directa entre navegadores para una comunicación directa de audio y video entre estos, sin la necesidad de utilizar una aplicación tercera. Para realizar la conexión entre navegadores, estos deben conocer cierta información del otro para poder hacer la conexión de WebRTC. Más específicamente, el navegador necesita saber el SDP (por sus siglas en inglés, *Session Description Protocol*) del otro para poder establecer la conexión de WebRTC.

Dado que ambos navegadores necesitan saber la información del otro antes de poder iniciar una conexión, se deben “conocer“ antes de poder iniciar dicha negociación. En esta etapa es donde es necesario el Servidor de Señalización, para que ambos navegadores tengan el conocimiento de con quién deben conectarse, y sobre todo, saber la información del otro para verificar que dicha conexión es posible.

En estas salas, cualquier cliente nuevo únicamente recibirá la retransmisión de audio y video de un solo cliente. Por lo tanto, se sabe que todo cliente nuevo debe empezar una nueva negociación con el cliente transmisor automáticamente, para poder establecer la conexión por medio de WebRTC y recibir la retransmisión de audio y video. Por medio de la librería Socket.io, al igual que en el sistema de mensajería instantánea, se crean las salas

```

socket.on('client-sdp', (SDPClienteNuevo) => {
  io.of('/videoroom')
    .to('transmisor').emit('client-sdp', SDPClienteNuevo)
})

```

Figura 71: Evento para escuchar el SDP del cliente y su señalización al transmisor.

```

socket.on('client-answer', (SDPTransmisor) => {
  io.of('videoroom')
    .to('clienteNuevo').emit('client-answer', SDPTransmisor)
})

```

Figura 72: Evento para escuchar el SDP del transmisor y respectiva señalización al cliente nuevo.

```

socket.on('ice-candidate', (data) => {
  io.of('videoroom')
    .to('clienteNuevo/Transmisor').emit('ice-candidate', data)
})

```

Figura 73: Evento para escuchar y señalizar el evento *ICECandidate* del cliente nuevo al transmisor y viceversa.

independientes.

Al ser un prototipo, se decidió tomar al primer cliente que entra a la sala como el transmisor de audio y video de esta. Al momento de conectarse un cliente a la sala, se le enviará una señal indicándole si es o no el transmisor de esta, por medio de una señal, Figura 69. Además, para cada sala se crearon sus propios eventos y señalizaciones para escuchar por mensajes nuevos y retransmitirlos a los demás clientes dentro de esta, Figura 70.

Para la negociación entre clientes para su conexión mediante WebRTC se crearon eventos y señalización para mandar sus respectivos SDP, además del envío de sus eventos para *ICECandidate*. El primer paso de la negociación es que el cliente nuevo envíe su SDP, emitiendo una señal al servidor, entonces el servidor escuchará por esta señal y transmitirá una señal al transmisor de la sala, enviándole el SDP del cliente nuevo, Figura 71. Entonces el transmisor empezará su propio proceso para inicializar una nueva conexión de WebRTC y emitirá una señal al servidor, enviando su SDP. El servidor escuchará por esta señal y emitirá una señal al cliente nuevo para redijirle el SDP del transmisor, Figura 72.

Para finalizar la negociación, el cliente y el transmisor deben compartirse sus eventos de *ICECandidate*. Siguiendo el orden de los pasos anteriores, una vez el cliente nuevo recibe el SDP del transmisor, este inmediatamente emite una señal, enviando su evento de *ICECandidate*. El servidor escucha esta señal y crea una señal para el transmisor, enviándole la información del cliente nuevo. Por último, se realiza este último proceso en viceversa, la información del transmisor se envía al cliente nuevo, Figura 73. Finalmente, tanto el cliente como el transmisor tienen toda la información necesaria para establecer una conexión de WebRTC entre ellos y comenzar la retransmisión de audio y video.

7.4.6. Integración del Frontend y el Backend

Para lograr la integración del servidor y del cliente se realizó la definición de protocolos en conjunto para tener en concreto de qué forma se iba a realizar la comunicación de información. Finalmente, una vez hechas las rutas de la API y los eventos de sockets hechos, se realizaron pruebas para comprobar la funcionalidad con ambas aplicaciones.

El funcionamiento del sistema de anuncios se puede observar en la Figura 74. A través de la API, el cliente es capaz de recuperar los anuncios guardados, además de agregar, modificar y eliminar. Por otro lado, el funcionamiento de la aplicación en mensajería instantánea se puede observar en la Figura 75, se muestra lado a lado dos clientes que están en una misma sala manteniendo una conversación.



Figura 74: Funcionamiento del sistema de anuncios en la webapp.

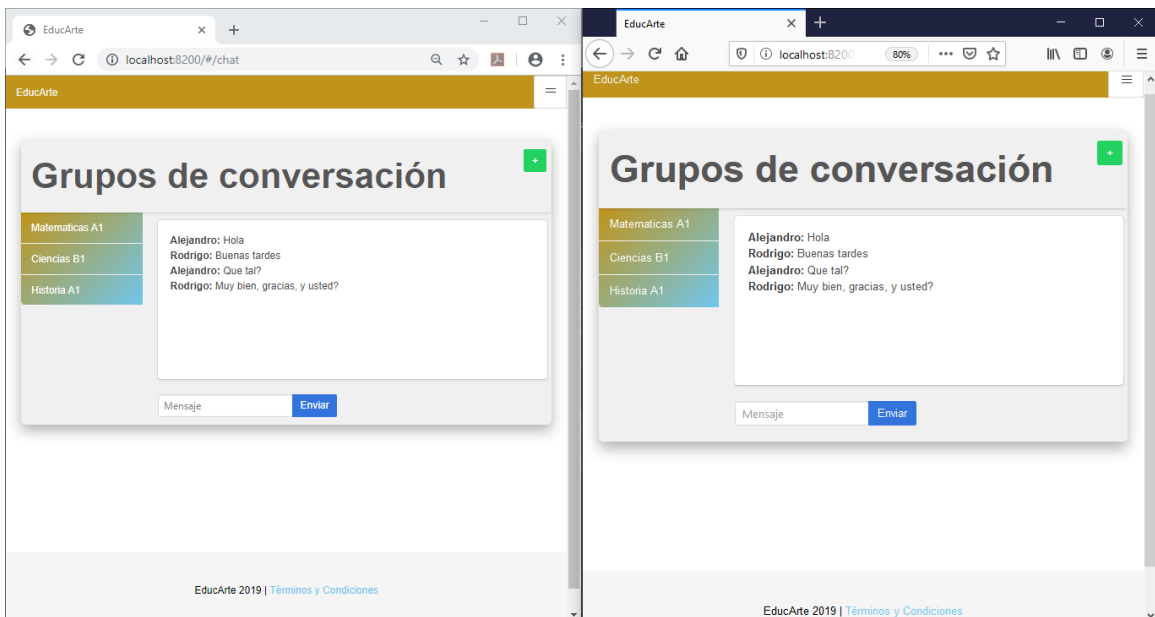


Figura 75: Funcionamiento del sistema de mensajería instantánea en la webapp.

7.5. Diseño e implementación del replicador de archivos

Para la realización del replicador de archivos, el lenguaje escogido fue *Go* también conocido *Golang*. Debido a que el lenguaje fue diseñado cuidadosamente alrededor de un modelo de concurrencia nuevo, que tiene la peculiaridad de integrarse muy bien con la sintaxis del lenguaje y que además es bastante ligero a comparación de los modelos utilizados por lenguajes como Java [89]. ¿Por qué el modelo de concurrencia fue el factor principal? Esto se explicará en las siguientes secciones.

7.5.1. Arquitectura del sistema objetivo

Para la arquitectura del sistema de replicación se tuvo en mente que iba a ser utilizado en un sistema distribuido que consiste de un nodo maestro y varios nodos réplica, cada nodo réplica representa uno de los servidores que alojan servicios educativos. Mientras que el nodo maestro es un servidor el cual siempre tendrá conexión a Internet y se encuentra fuera de la red en malla. En la Figura 76 se puede observar como se vería un sistema distribuido de cuatro comunidades para el cual fue diseñado el replicador.

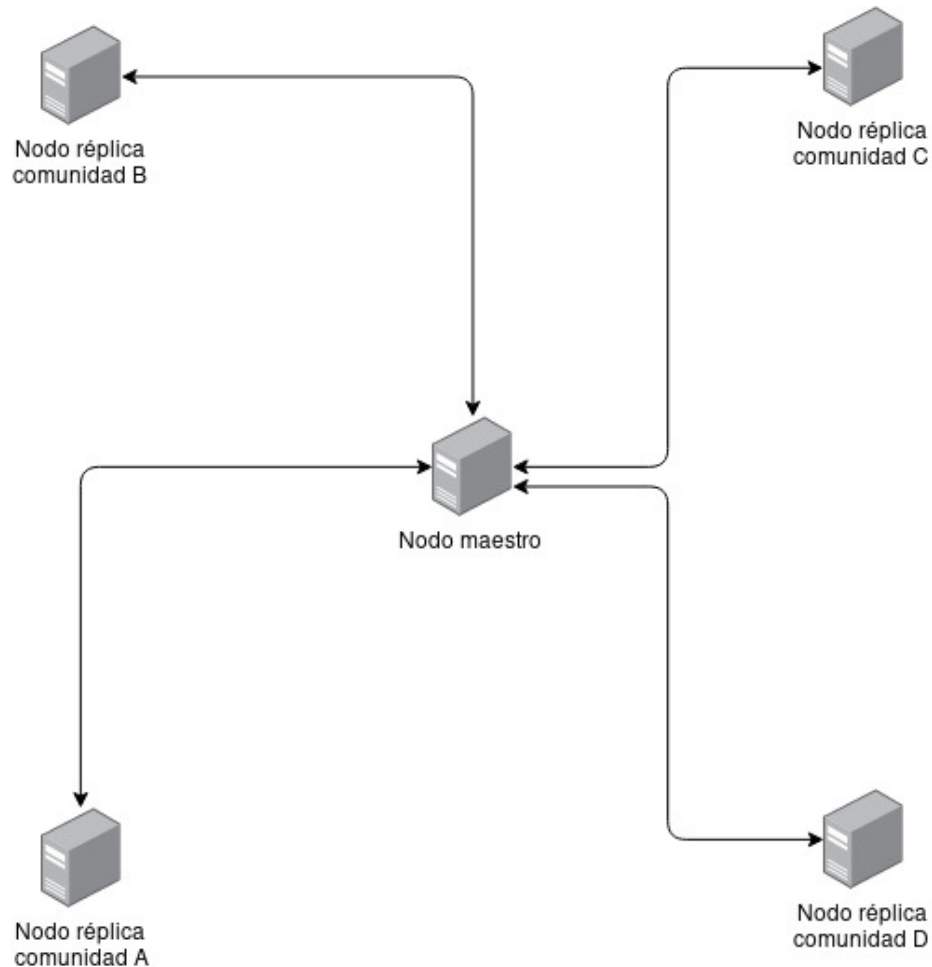


Figura 76: Arquitectura de los sistemas

Un **nodo maestro** es un servidor el cual su rol es únicamente almacenar la información que le transfieren uno o más nodos réplica, es decir es el encargado de tener las versiones oficiales de la base de datos y archivos de los servicios educativos registrados para luego distribuir los cambios a los nodos réplica que no se han sincronizado.

Un **nodo réplica** representa como se mencionó anteriormente a uno de los servidores que alojan un servicio educativo, con servicio educativo se refiere a una, varias o todas las siguientes instancias:

- Instancia de la interfaz de programación de aplicaciones de la enciclopedia en línea.
- Instancia del cliente de la enciclopedia en línea.
- Instancia de la interfaz de programación de aplicaciones de la plataforma de aprendizaje mediante foros activos.
- Instancia del cliente de la plataforma de aprendizaje mediante foros activos

Cada nodo réplica tiene, además de las características ya mencionadas, la peculiaridad de que se encuentran dentro de la red en malla de pocos recursos. Por lo que son elementos que se asumen que para funcionar no necesitan estar conectados constantemente al nodo maestro.

7.5.2. Funcionamiento del replicador

Con esta arquitectura en mente, se puede deducir que debe existir una instancia del replicador por cada directorio utilizado por un nodo réplica. Ya que para el alcance de este proyecto una instancia del replicador únicamente puede sincronizar un directorio a la vez.

Dado que el replicador debía ser bidireccional, es decir que los cambios en nodo réplica se deben de reflejar en un nodo maestro y los cambios en un nodo maestro se deben de reflejar en un nodo réplica, se debe tener tres hilos de ejecución, uno para detectar cambios en un directorio específico y encolarlos en una cola de eventos, otro para procesar dicha cola y ejecutar acciones de sincronización. Y un tercer hilo para mantenerse sincronizando nuevos cambios provenientes del nodo maestro.

Básicamente el funcionamiento del replicador de archivos se resume en el siguiente diagrama de flujo:

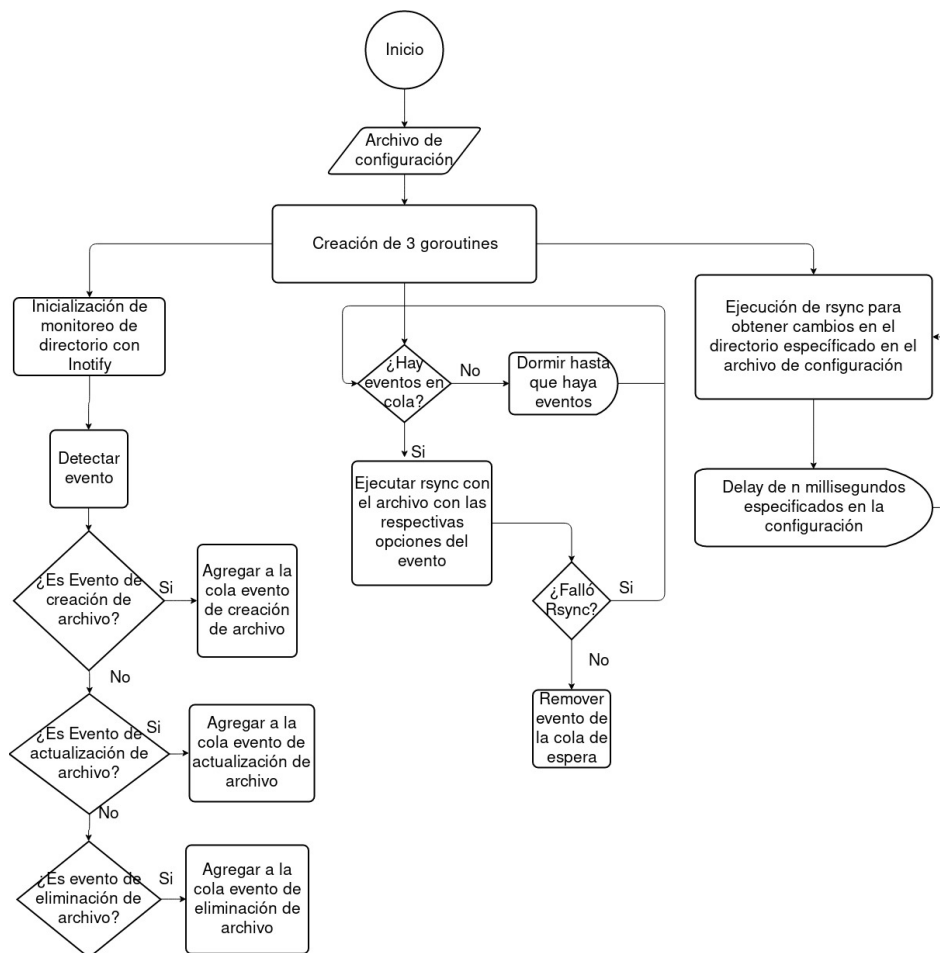


Figura 77: Diagrama de flujo del replicador

7.5.3. Dependencias del replicador y sus usos

Para el desarrollo del replicador de archivos se requirió de varias librerías y programas que facilitaron considerablemente la implementación de las funcionalidades, se tomó en cuenta que era bastante probable que el replicador fuera ejecutados en servidores con pocos recursos, por lo que se trató de utilizar la menor cantidad de dependencias posibles, y de ser posible la dependencia más ligera posible. Estas dependencias fueron:

- La herramienta *rsync* presente en muchos sistemas operativos Linux
- La librería para notificaciones de sistema de archivos *fsnotify*¹
- La librería para lectura de archivos de configuración en Go, *Viper*²
- La librería más ligera de Go para generación de registros desarrollada por Uber, *zap*³.

¹<https://github.com/fsnotify/fsnotify>

²<https://github.com/spf13/viper>

³<https://github.com/uber-go/zap>

Las dependencias pueden ser vistas con los enlaces puestos en las notas de pie de página.

rsync La herramienta *rsync* es la dependencia más importante para el funcionamiento del replicador, ya que es la que se encarga de realizar la diferencias entre los archivos del nodo réplica y el nodo maestro, comunicar estas diferencias y luego transferirlas. Así también se utiliza para actualizar todos los cambios que el nodo maestro ha recibido y que el nodo réplica no posee.

fsnotify Esta librería se utiliza para el manejo de eventos en un directorio específico del sistema de archivos, utiliza por debajo la interfaz de programación de aplicaciones de los sistemas Linux para monitoreo de sistemas de archivos, *inotify*. Los eventos de la librería que se utilizan al momento de detectar un cambio en el directorio que se le especifica al replicador con sus efectos son:

- **Creación de un archivo/directorio:** Al momento en que un archivo o directorio se crea dentro del directorio que se le especifica al replicador, se envía una señal a la herramienta *rsync* para que transfiera el nuevo archivo al nodo maestro.
- **Modificación de un archivo/directorio:** Al momento en que un archivo es modificado o movido, se envía una señal al nodo maestro con *rsync* para actualizar el archivo.
- **Eliminación de un archivo:** Al momento en que un archivo es eliminado, se envía una señal al nodo maestro de eliminar dicho archivo.

Viper El replicador necesita de un archivo de configuración, el cual establece la IP del servidor maestro y el módulo de *rsync* al cual se tiene que sincronizar, el archivo de configuración está escrito en formato TOML. La librería de *Viper* se encarga de analizar gramáticamente el archivo y obtener los datos necesarios para que el replicador funcione.

Zap Para el registro de eventos del replicador se decidió utilizar el registrador de Uber, Zap. Debido a que permitía mostrar información semi-estructurada y estructurada de la manera más eficiente según su documentación. [90]

7.5.4. Configuración del replicador

Como se mencionó en la sección 7.5.3 el replicador obtiene los parámetros de ejecución en un archivo en formato TOML el cual posee la siguiente estructura:

```

[options]
directory = "sync"
module = "wiki"
polling-rate = 10000

[server]
address = "127.0.0.1"

```

Figura 78: Ejemplo de archivo de configuración del replicador

Donde:

- *directory*: Directorio el cual se estará sincronizando
- *module*: Módulo de rsync del nodo maestro (si se tiene)
- *polling-rate*: Frecuencia en milisegundos para sincronizar cambios con el nodo maestro.
- *address*: Dirección IP del nodo maestro

Cabe destacar que para que el replicador esté correctamente configurado, en el nodo maestro se debe tener configurado un demonio de rsync con los módulos definidos:

```

lock file = /var/run/rsync.lock
log file = /var/log/rsyncd.log
pid file = /var/run/rsync.pid

[wiki]
path = /home/wiki
comment = Wiki directory
read only = no
host allow = 0.0.0.0/0

```

Figura 79: Ejemplo de archivo de configuración del proceso de *rsync* en el nodo maestro que funciona con el archivo de configuración de la Figura 78

Si se desea ver las configuraciones del demonio de *rsync* se puede referir a la documentación de dicha herramienta en el pie de página⁴.

7.6. Diseño e implementación de la interfaz de programación de aplicaciones de la enciclopedia en línea

Luego de tener ya diseñado e implementado el replicador de archivos, hacía falta cumplir con el objetivo de realizar una interfaz de programación de aplicaciones del servicio de

⁴<https://linux.die.net/man/5/rsyncd.conf>

enciclopedia en línea. Para diseñar dicha interfaz, se tomó en consideración la presencia del replicador, la forma en que este funciona y la arquitectura para la cual fue desarrollado.

La interfaz fue pensada para funcionar en un entorno con recursos restringidos como una *Raspberry Pi*, por lo que se decidió utilizar SQLite como mecanismo de persistencia, debido a que no requiere de ningún servidor para funcionar, lo extremadamente ligera que es la librería y lo rápidas que son las consultas a la base de datos incluso en entornos restringidos como en una *Raspberry Pi*. [40]

Debido a la forma en que se maneja las escrituras en SQLite y para mantener las propiedades de atomicidad, consistencia, aislamiento y durabilidad de la información, no fue posible utilizar el replicador para sincronizar la base de datos con el nodo maestro. Ya que si *rsync* se encuentra transfiriendo datos en medio de una transacción de la base de datos, el archivo de la base de datos almacenado en el nodo maestro puede ser corrompido fácilmente. Como solución a este problema se decidió utilizar la herramienta *SymmetricDS* que está específicamente diseñada para replicación unidireccional y bidireccional de múltiples tipos bases de datos, entre ellas SQLite.

7.6.1. Diseño de los esquemas de la base de datos

La enciclopedia en línea se pensó para ser utilizada únicamente por usuarios que estén registrados en el sistema, por lo que debía de existir un mecanismo de autenticación en esta. Para esto se creó una tabla *user* que se encargara de almacenar la información de los usuarios dentro del sistema.

Los atributos de la tabla *user* fueron los siguientes:

- *id* Llave foránea de la tabla de usuario
- *username* Nombre de usuario para autenticación
- *first_name* Primer nombre de la persona
- *last_name* Apellido de la persona
- *password* Contraseña ofuscada con algún mecanismo de *hashing*

user [table]	
id	-
username	
first_name	
last_name	
password	
1 row	1 >

Figura 80: Atributos de la tabla de usuarios

Se consideró que aunque el servicio fuera una enciclopedia en línea, los estudiantes no pueden editar las clases textuales que los docentes preparan para ellos, por lo que se definió una tabla de rol que únicamente será utilizada por la interfaz de programación de aplicaciones para decidir si proveer acceso a recursos y a ciertas acciones. Esta tabla tiene la funcionalidad de describir los posibles roles del sistema.

La tabla se denominó *role* y los atributos de la tabla son los siguientes:

- *name*: Nombre del rol
- *description*: Descripción del rol

role [table]	
🔑	id
	name
	description
2 rows	1 >

Figura 81: Atributos de la tabla de roles

Dado que el esquema se pensó para que en un futuro la enciclopedia pudiera tener más funcionalidades, se determinó que la relación entre un usuario y su rol, tiene que ser de uno a muchos ya que probablemente en un futuro la enciclopedia necesite que un usuario pueda tener varios roles para realizar determinadas acciones específicas. Para esto se creó una tabla intermedia *user_role*, con la única funcionalidad de mapear un usuario a varios roles. Aunque para este trabajo se permitió que únicamente se pueda mapear un usuario a un único rol.

Los atributos de la tabla *user_role* son los siguientes:

- *user_id*: Identificador del usuario
- *role_id* Identificador del role a mapear del usuario

user_role [table]	
🔑	user_id
🔑	role_id
< 2	1 row

Figura 82: Atributos de la tabla usuarios - roles

Como la enciclopedia en línea estuvo ideada para la educación en las comunidades, únicamente se contemplaron los grados del sistema educativo de Guatemala que van de

primero primaria hasta quinto bachillerato. Para esto se creó una tabla *grade* que representa lo anterior.

Los atributos de la tabla *grade* son los siguientes:

- *id*: Llave foránea de la tabla.
- *name*: Nombre del grado
- *description*: Descripción del grado

grade [table]	
id	
name	
description	
1 row	1 >

Figura 83: Atributos de la tabla de grados

Puesto que un grado tiene múltiples cursos, por lo que se decidió seccionar los grados por cursos, creando así una tabla denominada *course*. Formando una relación de un grado a muchos cursos

Los atributos de la tabla *course* son los siguientes:

- *id*: Identificador del curso
- *grade_id*: Identificador del grado al que pertenece el curso.
- *name*: Nombre del curso
- *description*: Descripción del curso

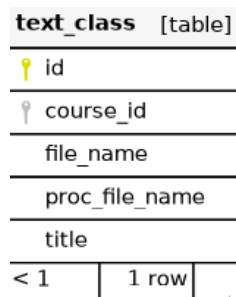
course [table]	
id	
grade_id	
name	
description	
< 1	1 row 1 >

Figura 84: Atributos de la tabla de curso

La última entidad en la base de datos es la de *text_class* que se encarga de representar el recurso de una clase textual de la enciclopedia en línea.

Los atributos de la tabla son los siguientes:

- *id*: Identificador de la clase textual/página de la enciclopedia
- *course_id*: Identificador del curso al que pertenece la clase textual
- *file_name*: Nombre del archivo que contiene la clase textual en formato Markdown
- *proc_file_name*: Nombre del archivo procesado (se explicará en la siguiente sección).
- *title*: Título de la clase textual.



The image shows a screenshot of a database table structure for 'text_class'. The table has the following columns: 'id' (primary key), 'course_id' (foreign key), 'file_name', 'proc_file_name', and 'title'. The table is currently empty, showing 1 row.

text_class [table]	
id	
course_id	
file_name	
proc_file_name	
title	
< 1	1 row

Figura 85: Atributos de la tabla de clases textuales

Luego de haber descrito cada una de las tablas presentes en la interfaz de programación de la aplicaciones, ya es posible observar holísticamente como éstas se relacionan entre sí. La siguiente figura lo demuestra:

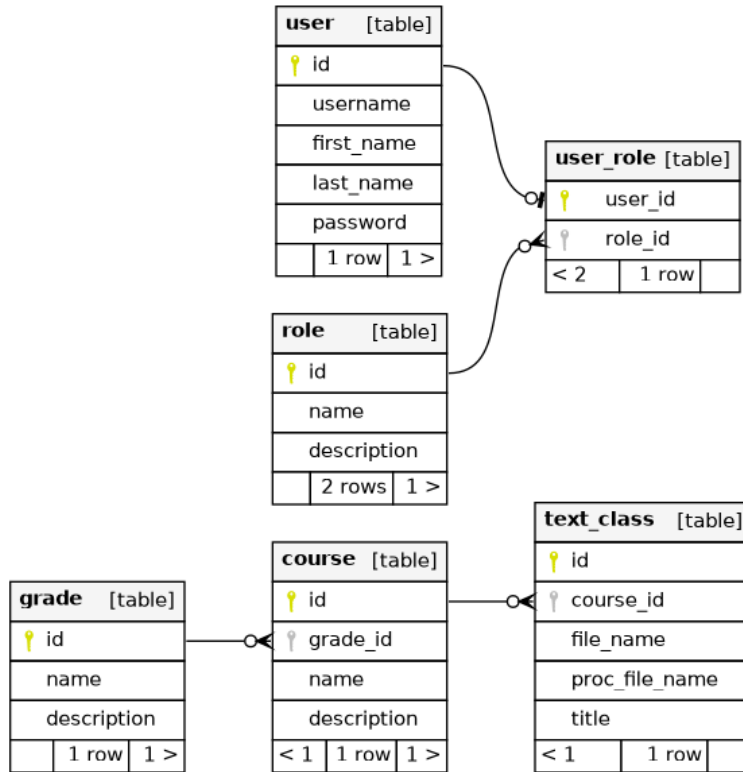


Figura 86: Diagrama entidad-relación utilizada por la interfaz de programación de aplicaciones de la enciclopedia en línea

Nótese que los usuarios no están relacionados a los grados, cursos y clases directamente textuales ya que para el alcance de este trabajo se estableció que cualquier usuario que sea estudiante puede acceder leer a los contenidos de cualquier grado, pero no modificarlo ni borrarlo. El único tipo de usuario que puede modificar y borrar son aquellos que tengan el rol de docente dentro de la plataforma.

7.6.2. Diseño e implementación de las peticiones de la interfaz de programación de aplicaciones

El estilo de arquitectura utilizado para la interfaz de programación de aplicaciones fue REST, por lo que el protocolo de comunicación entre el cliente de la enciclopedia en línea y la interfaz de programación de aplicaciones se encuentra basados en recursos con nombres derivados de las tablas de la base de datos.

Las peticiones relacionadas a la autenticación y registro de usuarios son los siguientes:

- HTTP POST /users Esta petición se encarga de registrar un usuario, recibe un JSON en el cuerpo del mensaje:

```

1  {
2  "username": "jrcustodio",

```

```
3     "firstName": "Rodrigo",
4     "lastName": "Custodio",
5     "password": "123456789"
6 }
```

Si la respuesta es correcta se devuelve un estado 201, indicado que el usuario fue registrado correctamente.

- HTTP POST /auth Esta petición se encarga iniciar sesión a los usuarios, recibe un JSON en el cuerpo con los credenciales del usuario:

```
1 {
2   "username": "jrcustodio",
3   "password": "123456789"
4 }
```

Devuelve un JSON con un JWT que el cliente debe enviar en el encabezado de *Authorization* en las peticiones que requieran de autenticación:

```
1 {
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC..."
3 }
```

- HTTP POST /auth/token Por defecto los tokens que la interfaz de programación de aplicaciones devuelve en la petición anterior duran 1 hora. Al momento en que pase esa hora el token queda invalidado. Debido a esto se realizó una petición que permita refrescar dicho token 30 segundos antes de que expire, y así evitar que el cliente cierre la sesión del cliente. El cliente tiene la obligación de refrescar dicho token antes de que expire.

La petición únicamente necesita de que el encabezado de *Authorization* tenga el token que se encuentra por expirar para devolver uno nuevo. La respuesta del servidor en caso de cumplir con lo necesario es:

```
1 {
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC..."
3 }
```

Las peticiones relacionadas a la tabla de *grade* son las siguientes:

- HTTP POST /grade

Esta petición se encarga de crear un grado, con un nombre y una descripción, recibe un JSON con los parámetros, requiere de autenticación y tener el rol de un maestro:

```
1 {
2   "name": "Primero primaria",
3   "description": "Clase de primero primaria"
4 }
```

Devuelve un JSON parecido al anterior, únicamente con un nuevo identificador indicando que el recurso fue creado.

```
1  {
2    "id": 13,
3    "name": "Primero primaria",
4    "description": "Clase de primero primaria"
5  }
```

- HTTP GET /grade?size=&nextToken=

Esta petición se encarga de devolver el listado de grados creados paginados, recibe por medio de los parámetros URL los atributos de *size* para indicar el tamaño de página y *nextToken* que es un token que sirve para obtener la siguiente página.

La respuesta del servidor es la siguiente:

```
1  {
2    "data": [
3      {
4        "id": 12,
5        "name": "Primero primaria",
6        "description": "Clase de primero primaria"
7      },
8      {
9        "id": 13,
10       "name": "Segundo primaria",
11       "description": "Clase de segundo primaria"
12     }
13   ],
14   "size": 25,
15   "nextToken": 13
16 }
```

- HTTP PUT /grade/:id

Esta petición se encarga de actualizar los datos del recurso de grado, recibe el identificador de un grado en el nombre de la ruta y un JSON con los parámetros a modificar, requiere de autenticación y tener el rol de un maestro:

```
1  {
2    "name": "Sexto primaria",
3    "description": "Grado de sexto primaria"
4  }
```

Devuelve un estado 204 indicando que se modificó el recurso

- HTTP DELETE /grade/:id

Esta petición se encarga de eliminar un grado, recibe el identificador de un grado en el nombre de la ruta, requiere de autenticación y tener el rol de un maestro. Devuelve un 204 indicando que el recurso fue eliminado correctamente.

Las peticiones relacionadas a la tabla de *course* son las siguientes:

- HTTP POST `/grade/:id/course`

Esta petición se encarga de crear un curso, con un nombre y una descripción, recibe el identificador del grado en la dirección URL y un JSON con los parámetros, requiere de autenticación y tener el rol de un maestro:

```
1  {
2    "name": "Ciencias ambientales",
3    "description": "Curso de ciencias ambientales para
4    primer grado"
5  }
```

Devuelve un JSON parecido al anterior, únicamente con dos nuevo identificadores indicando que el recurso fue creado.

```
1  {
2    "id": 10,
3    "gradeId": 14,
4    "name": "Ciencias ambientales",
5    "description": "Curso de ciencias ambientales para
6    sexto primaria"
7  }
```

- HTTP GET `/grade/:id/course?size=&nextToken=`

Esta petición se encarga de devolver el listado de cursos creados paginados, recibe por medio de los parámetros URL los atributos del identificador del grado, *size* para indicar el tamaño de página y *nextToken* que es un token que sirve para obtener la siguiente página.

La respuesta del servidor es la siguiente:

```
1  {
2    "data": [
3      {
4        "id": 10,
5        "gradeId": 14,
6        "name": "Ciencias ambientales",
7        "description": "Curso de ciencias
8        ambientales para sexto primaria"
9      }
10     ],
11     "size": 25,
12     "nextToken": 10
13  }
```

- HTTP PUT `/grade/:id/:courseid`

Esta petición se encarga de actualizar los datos del recurso de curso, recibe el identificador de un grado y un curso en el nombre de la ruta y un JSON con los parámetros a modificar, requiere de autenticación y tener el rol de un maestro:

```
1  {
2    "name": "Matematica",
3    "description": "Curso de matematica"
4  }
```

Devuelve un código 204 indicando que se modificó el recurso

- HTTP DELETE /grade/:id/:courseid

Esta petición se encarga de eliminar un grado, recibe el identificador de un grado y un curso en el nombre de la ruta, requiere de autenticación y tener el rol de un maestro. Devuelve un código 204 indicando que el recurso fue eliminado correctamente.

Las peticiones relacionadas a la tabla de *text_class* son las siguientes:

- HTTP POST /grade/:id/:courseid/textclass

Esta petición se encarga de crear una clase textual, con un nombre y una descripción, recibe el identificador del grado y curso en la dirección URL y un JSON con los parámetros, requiere de autenticación y tener el rol de un maestro:

```
1  {
2    "title": "Que es Guatemala?"
3  }
```

Devuelve un JSON parecido al anterior, únicamente con dos nuevo identificadores (el del curso y de la clase) indicando que el recurso fue creado.

```
1  {
2    "id": 1,
3    "courseId": 10,
4    "title": "Que es Guatemala?"
5  }
```

- HTTP GET /grade/:id/:courseid/textclass?size=&nextToken=

Esta petición se encarga de devolver el listado de clases textuales creadas paginadas, recibe por medio de los parámetros URL los atributos del identificador del grado y curso, *size* para indicar el tamaño de página y *nextToken* que es un token que sirve para obtener la siguiente página.

La respuesta del servidor es la siguiente:

```
1  {
2    "data": [
3      {
4        "id": 1,
5        "courseId": 10,
```

```

6         "title": "Ciencias ambientales 3"
7     }
8 ],
9     "size": 25,
10    "nextToken": 1
11 }

```

- HTTP PUT /grade/:id/:courseid/:classid

Esta petición se encarga de actualizar los datos del recurso de clase textual, recibe el identificador de un grado, un curso y una clase en el nombre de la ruta y un JSON con los parámetros a modificar, requiere de autenticación y tener el rol de un maestro:

```

1 {
2     "title": "Prueba"
3 }

```

Devuelve un código 204 indicando que se modificó el recurso

- HTTP DELETE /grade/:id/:courseid/:classid

Esta petición se encarga de eliminar una clase textual, recibe el identificador de un grado, un curso y una clase textual en el nombre de la ruta, requiere de autenticación y tener el rol de un maestro. Devuelve un código 204 indicando que el recurso fue eliminado correctamente.

- HTTP POST /grade/:id/:courseid/:classid/file

Esta petición se encarga de crear un archivo de Markdown de la clase textual en el servidor y marcarlo para su procesamiento que será discutido en la siguiente sección.

Recibe un cuerpo de tipo *form-data* con un parámetro file, donde va colocada el archivo de texto plano que contiene el Markdown. Devuelve un valor 204 en caso de que el archivo Markdown fue aceptado.

- HTTP GET /grade/:id/course/:courseid/textclass/:classid/file

Esta petición devuelve el archivo de texto que representa la clase textual, en caso de que el archivo ya fue procesado, devuelve el archivo procesado. En todos los otros casos devuelve el archivo que se subió sin modificaciones.

7.6.3. Procesador de archivos Markdown

La idea del proyecto es que una vez la red en malla deje de tener Internet la experiencia en la enciclopedia en línea sea lo más pegada posible a cuando la red tenía Internet. Para cumplir con este objetivo se implementó un procesador de archivos Markdown que detecta direcciones *URL* de imágenes y páginas web dentro del texto que el maestro ingresó, las descarga y las reemplaza por su versión local. Para que así se muestren incluso cuando ya no haya una conexión a Internet.

Debido a que el procesador de archivos se ideó con flexibilidad en mente, se puede colocar en el nodo maestro para ahorrar ancho de banda en la red en malla. O si bien

esta última no posee una conexión tan mala se puede colocar en los nodos réplica para descargar directamente las imágenes y tener los archivos procesados localmente más rápido. La interfaz de programación de aplicaciones se encarga de exponer las imágenes de los archivos procesados automáticamente sin necesidad de algún tipo de configuración

```
#Ejemplo
```

```
Esto es un markdown de prueba
```

```
![test](https://google.com/common/images/icon48.png "Logo Title Text 1")
```

Figura 87: Ejemplo de archivo de Markdown sin procesar

```
#Ejemplo
```

```
Esto es un markdown de prueba
```

```
![test](https://localhost:3000/3/3/4/icon48.png "Logo Title Text 1")
```

Figura 88: Ejemplo de archivo de Markdown procesado

Nótese que dirección URL de la imagen es reemplazada por la dirección estática que la interfaz de programación de aplicaciones expone.

7.7. Integración del sistema de replicación de archivos con la enciclopedia en línea y la plataforma de aprendizaje mediante foros activos

7.7.1. Integración con la enciclopedia en línea

Para lograr la integración del sistema de replicación de archivos con la enciclopedia en línea y la plataforma de aprendizaje, basta con instalar la herramienta *rsync* en el nodo maestro, crear el archivo de configuración de `rsyncd.conf` en la ubicación `/etc/` del sistema operativo Linux donde se estará ejecutando el servicio de la enciclopedia en línea.

Dentro del archivo se debe colocar lo siguiente `rsyncd.conf`:

```

lock file = /var/run/rsync.lock
log file = /var/log/rsyncd.log
pid file = /var/run/rsync.pid

[default]
    path = /home/default
    read only = yes
    list = yes
    auth users = def
    secrets file /etc/rsyncd.secrets

[wiki]
    path = /home/wiki
    comment = Wiki directory
    read only = no
    host allow = 0.0.0.0/0

```

Figura 89: Archivo de configuración de rsyncd

Luego se debe crear un archivo TOML del replicador, con las siguientes configuraciones:

```

[options]
directory = "/home/wiki/sync"
module = "wiki"
polling-rate = 10000

[server]
address = "192.168.1.1"

```

Figura 90: Archivo de configuración `config.toml` de replicador para sincronizar la carpeta de archivos Markdown de la interfaz de programación de aplicaciones de la enciclopedia en línea

Aquí se le especifica al replicador que sincronice el directorio `/home/wiki/sync` con el módulo de *wiki* del servidor con `192.168.1.1`.

Para ejecutar el replicador simplemente se inicia con el parámetro “-c” con el nombre del archivo:

```
$ ./rclient -c config.toml
```

El replicador mostrará una salida como la siguiente:

```
INFO[2019-11-03T21:36:53-06:00] Initializing replicator client...
INFO[2019-11-03T21:36:53-06:00] Config file loaded
INFO[2019-11-03T21:36:53-06:00] Initializing rclient with directory "/home/wiki/sync/"
INFO[2019-11-03T21:36:58-06:00] Pulling changes
INFO[2019-11-03T21:36:58-06:00] receiving incremental file list

sent 20 bytes  received 228 bytes  496.00 bytes/sec
total size is 3.15M  speedup is 12,703.40
INFO[2019-11-03T21:37:03-06:00] Pulling changes
INFO[2019-11-03T21:37:03-06:00] receiving incremental file list

sent 20 bytes  received 228 bytes  496.00 bytes/sec
total size is 3.15M  speedup is 12,703.40
INFO[2019-11-03T21:37:08-06:00] Pulling changes
```

Figura 91: Salida del replicador indicando que se ha logrado sincronizar

Luego solo basta con inicializar la interfaz de programación de aplicaciones en el mismo directorio (con la opción “-dir”)

```
→ wiki-backend git:(master) X ./wiki -dir /home/wiki/sync/
{"level":"info","ts":1572838735.1186152,"caller":"log/log.go:52","msg":"initialazing resource",
 "resource":"backend","info":"port:3000"}
```

Figura 92: Salida de la interfaz de programación de aplicaciones de que ha iniciado correctamente

Por último solo se inicializa el procesador de imágenes en el mismo directorio (con la opción de “-dir”) y ya se habrá integrado el replicador con la enciclopedia en línea por completo:

```
→ wiki-backend git:(master) X ./imgproc -dir /home/wiki/sync/
{"level":"info","ts":1572839067.863515,"caller":"log/log.go:52","msg":"initialazing resource",
 "resource":"imageproc","info":"with directory /home/wiki/sync/"}
```

Figura 93: Salida del replicador indicando que se ha logrado sincronizar

7.7.2. Integración con la plataforma de aprendizaje mediante foros activos

Muy parecido a la sección anterior, únicamente se debe agregar un módulo nuevo al archivo de la Figura 89:

```

lock file = /var/run/rsync.lock
log file = /var/log/rsyncd.log
pid file = /var/run/rsync.pid

[default]
    path = /home/default
    read only = yes
    list = yes
    auth users = def
    secrets file /etc/rsyncd.secrets

[wiki]
    path = /home/wiki
    comment = Wiki directory
    read only = no
    host allow = 0.0.0.0/0

[comchat]
    path = /home/comchat
    comment = Active learning forum directory
    read only = no
    host allow = 0.0.0.0/0

```

Figura 94: Archivo de configuración de rsyncd

Luego solo se procede ejecutar el replicador con el siguiente archivo de configuración:

```

[options]
directory = "/home/comchat/chat-server/data"
module = "comchat"
polling-rate = 10000

[server]
address = "192.168.1.1"

```

Figura 95: Archivo de configuración `config.toml` de replicador para sincronizar la carpeta de archivos estáticos de la plataforma de foros activos

Se obtendrá una salida similar a la de la Figura 91, para luego únicamente ejecutar la interfaz de programación de aplicaciones de la plataforma de aprendizaje mediante foros activos.

7.8. Diseño de la red

- El diseño de la red fue realizado tomando en cuenta las limitantes y restricciones que podrían existir en una implementación en el área rural. La mejor alternativa que se

encontró para manejar el acceso de los usuarios a la red es mediante la implementación de un portal captivo. OpenWrt, que es un firmware que le fue instalado a las antenas, posee una implementación de un portal captivo que puede ser utilizada en dado caso no se desea implementar el portal captivo desde cero.

El siguiente es un diagrama de la red mesh haciendo uso de un servidor web y un portal captivo. En este diagrama se puede observar que los usuarios, mediante distintos dispositivos, logran conectarse con los puntos de acceso (antenas de la red) para acceder a los sitios deseados tras pasar por el portal captivo.

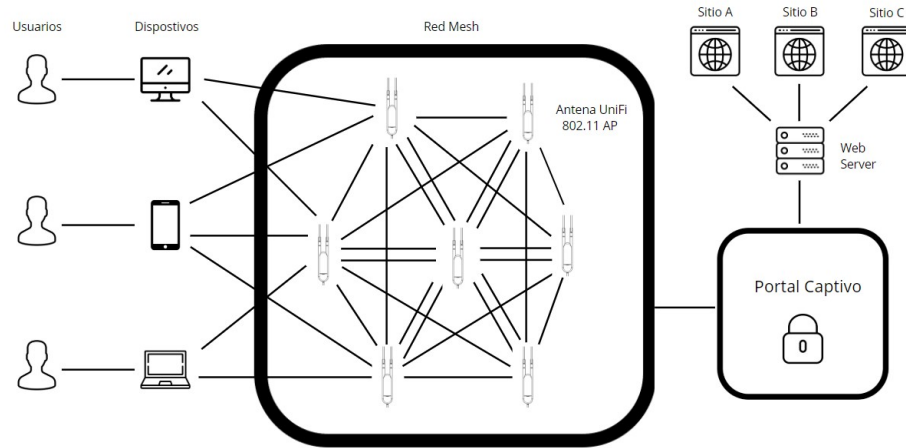


Figura 96: Diagrama de red mesh.

- Según un análisis entre los distintos proveedores se obtuvo el siguiente cuadro:

Proveedores de paneles solares en Guatemala									
Proveedor	Soporte de instalación	Tiempo de vida		Retención del equipo	Cantidad de antenas soportadas	Ventajas	Desventajas	Frecuencia de pago	Costo
		Paneles solares	Baterías						
Solar Guatemala	Si	20 - 25 años	2 - 3 años	Si	3	Equipo tipo "plug and play"	1. Costo inicial alto 2. Poca diversidad en tamaño de paneles	Único	Q5,500.00
SolarGuat	No	20 - 25 años	2 - 3 años	Si	3	Equipo variado en tamaño	Falta de soporte técnico en la instalación de las antenas	Único	Q2,825.00
Kingo Energy	Si	10 - 15 años	2 - 3 años	No	1	Experiencia previa en comunidades rurales	1. Costo mensual 2. No se adquieren los equipos	Mensual	Q 110.00

Decisión Kingo Energy

Cuadro 17: Cuadro comparativo de proveedores de paneles solares en Guatemala.

Basándonos en términos de costos y calidad del servicio que ofrece cada una de estas alternativas, se decidió que la alternativa de Kingo Energy es la mejor para este proyecto. Se escogió esta alternativa porque es la que requiere de una menor cantidad de capital inicial a invertir y porque facilita el proceso de instalación del equipo. Además,

se consideró que la experiencia previa por parte de su equipo trabajando con comunidades rurales sería una gran ayuda para integrarse con la comunidad. Otra ventaja de Kingo es que, dependiendo de las necesidades de la implementación de la red, permite que el servicio pueda ser interrumpido sin ningún problema o repercusión. Aún así, se tiene presente que esta alternativa requiere de un administrador que se encargue de cumplir con los pagos del servicio. Se consideró que decisión permite que el proyecto sea más viable en términos económicos y logísticos.

- La guía de acercamiento a las comunidades resultó en un enfoque que tiene como objetivo tomar en cuenta la participación de la mayor cantidad de personas de parte de la comunidad, en la ejecución y seguimiento de un proyecto, para mejores resultados.

Las municipalidades y alianzas con otras organizaciones son el mejor punto de entrada para cualquier proyecto que está en sus inicios. La integración de distintos proyectos, como los proyectos de salud, educación y tecnología hace que las personas de las comunidades estén más interesadas en formar parte de estos proyectos. Una vez se haya realizado el proyecto en una comunidad replicar el proceso se vuelve más fácil e incluso en algunas ocasiones son las comunidades quienes buscan que en su localidad se implementen los proyectos.

Es importante recordar que adaptarse a las barreras que se puedan presentar al momento de necesitar la participación del usuario. Esto se debe hacer con el fin eliminar la mayor cantidad de barreras posibles y obtener la mejor participación por parte de un usuario.

- Se logró implementar un servicio de comunicación de tres funcionalidades: publicación de anuncios, servicio de chat instantáneo y videollamadas uno a uno.
- Se diseñó e implementó una interfaz gráfica usable, con tiempos promedio aceptables para la realización de cada tarea importante y tasas de éxito del 100 %.
- Mediante la implementación del chat y las videollamadas, se habilitan nuevos canales de comunicación entre maestro y alumnos, lo cual sienta las bases para un foro de educación activa.
- Las funcionalidades del foro de aprendizaje interactivo se lograron implementar a través de una API, que brinda los servicios necesarios para la completa funcionalidad del cliente.
- Para la construcción de un foro de aprendizaje interactivo que permita la transmisión de texto, audio y video en una red con un acceso a internet limitado o nulo, fue necesario la implementación de la comunicación mediante WebRTC y el uso de *sockets*. Ya que WebRTC ayuda a eliminar la dependencia de una aplicación externa al sistema, mientras que los *sockets* permiten la comunicación entre el servidor y el cliente sin la dependencia de una conexión a internet.
- Es importante definir una topología de red al momento de implementar la comunicación WebRTC, en este caso se eligió utilizar una topología de anillo por las limitaciones de ancho de banda de la red. Cada topología ofrece una serie de ventajas y desventajas que impactarán el funcionamiento de la aplicación y el servidor, hablando de capacidad computacional; y también dependerá del ancho de banda de la red.
- Se cumplió con el objetivo de realizar una interfaz de usuario para una aplicación educativa, puesto que se desarrolló esta interfaz previo al diseño y ejecución de las pruebas de usabilidad.
- En cuanto al objetivo de diseñar una interfaz de usuario que se acople a los recursos tecnológicos disponibles en la comunidad, se cumplió de manera que se escogió un *stack*

de tecnologías que apoyaron en esto: el desarrollo de la aplicación con el *framework* VueJS y el uso de documentos con lenguaje *markdown* para el material educativo.

- En vista que se obtuvo un porcentaje de completación arriba del 75% para todas las tareas evaluadas en las pruebas de usabilidad, y que además se obtuvo un número de pasos efectuados por los usuarios cercano al número de pasos efectuados por el desarrollador, se concluye que el prototipo elaborado cumple con los requisitos mínimos de usabilidad. Esta afirmación también se ve apoyada en la aprobación verbal que se obtuvo de los usuarios que realizaron las pruebas, puesto que indicaron tener una satisfacción general al usar el prototipo.
- Para la implementación el replicador de archivos y la interfaz de programación de aplicaciones, fue necesario establecer primero que arquitectura iba a tener el sistema objetivo, para que basado en las características de ésta se pudiera diseñar ambos sistemas y acoplarlos sin mayor dificultad.
- Para construir un sistema distribuido con resiliencia y que funcione sobre una red de bajos recursos es de suma importancia subdividirlo en sistemas independientes que hagan una sola cosa y que la hagan bien.
- El replicador diseñado e implementado en este trabajo no es solamente útil en situaciones en donde los recursos de la red son limitados con conexiones a Internet intermitentes, si no que también puede ser utilizado en cualquier escenario en donde se necesite transmitir archivos de manera bidireccional en intervalos de tiempo.
- En situaciones en donde existen muchos nodos replica, es recomendable colocar el procesador de archivos Markdown en el nodo maestro, ya que esto asegura que las imágenes sean distribuidas a los otros nodos replica más rápido.
- Utilizar la herramienta *rsync* para sincronizar una base de datos como SQLite en media transacción puede ocasionar que esta se corrompa y se propague a todos los nodos replica. Por eso para esta situación siempre es recomendable utilizar software como SymmetricDS que consideran estos casos extremos.
- Según los requerimientos de seguridad, identificados en un análisis de riesgos, se diseñó un modelo de Red *Mesh* que cumple los requerimiento de seguridad de una red comunitaria.
- Las municipalidades y alianzas con otras organizaciones son el mejor punto de entrada con las comunidades cuando se quiere iniciar un nuevo proyecto.
- El uso de paneles solares para el suministro de energía eléctrica de los nodos de la red garantiza el suministro continuo de los nodos sin importar la fluctuación de energía eléctrica en la comunidad.
- Kingo Energy es el proveedor que mejores características presentó acorde a las necesidades planteadas para este proyecto.
- El acercamiento hacia las comunidades rurales debe empezar por el contacto e identificación de los miembros o representantes de la comunidad.

Recomendaciones

1. Se recomienda usar *web sockets* y especialmente la librería de *socket.io* para la implementación del servicio de chat, por clara sintáxis y rápido funcionamiento.
2. Se recomienda el uso de *Vue.js* para la realización de la interfaz gráfica, aunque elegir una herramienta como *React* puede ser igual de funcional.
3. La utilización de la librería *Bulma.io* simplifica el diseño de la interfaz, pero existen otras opciones que se pueden utilizar como *Buefy* que tienen más componentes.
4. Se recomienda el uso de *WebRTC* para la realización de videollamadas, pero para el mejor funcionamiento de estas y lograr hacer llamadas de uno a muchos, se debe implementar o usar una implementación existente de un *Media Server* como Kurento.
5. En el diseño de la aplicación se decidió hacer una conexión directa entre navegadores para la comunicación de audio y video. Sin embargo, mientras la cantidad de nodos va aumentando, la transmisión de esta información es cada vez más pesada, tanto en poder computacional como en ancho de banda. Por lo que recomienda usar un Servidor para *Media*, el cual sirve para centralizar esta información, llevando un mejor control y eliminando la carga computacional localmente en los usuarios. Existen varias implementaciones de un servidor de este tipo, entre ellos, el más conocido, el MCU (por sus siglas en inglés, *Multipoint Control Unit*).

El MCU trabaja bajo una topología estrella, en donde todos los nodos se conectan a este de forma bidireccional, de esta forma, el MCU es capaz de capturar todas las transmisiones que se realicen y retransmitirlas hacia todos los nodos en una única conexión. Esto es especialmente útil si la transmisión de audio y video se quiere hacer desde varios nodos, es decir, no existe solo un transmisor, y entonces, la carga computacional necesitaría ser aún más pesada localmente en los usuarios.

6. En este proyecto se utilizó una topología estrella al momento de conectar los usuarios por medio de WebRTC, donde cada cliente nuevo se conectaba de manera automática al usuario que estaba realizando la transmisión de audio y video. Existen diferentes formas de cómo se pueden conectar los usuarios para reducir la carga computacional

sobre el transmisor. Entre ellas, se puede utilizar la idea de una topología en red parcial en malla. Esto significa que para pertenecer a la red debe estar conectado al menos a un nodo que ya pertenece a la red. Basados en esta topología, se pueden realizar dos implementaciones:

- a) Conforme se van agregando nodos, elegir nodos que pueden soportar la carga computacional para redirigir la transmisión que reciben hacia otros nodos. De esta forma, se crean pequeños grupos de nodos que forman una pequeña red centralizada con forma de estrella, pero que a grandes escalas, sigue funcionando como una topología de red en malla.
 - b) Cada nodo únicamente se conecta a dos nodos, uno que le transmite la información, y otro al que le comunica dicha información. De esta manera, se crea una gran fila de nodos interconectados sin saturar ninguno de estos nodos con tanta carga computacional. El problema de esta implementación radica en el algoritmo para enrutar los nodos cuando uno de ellos sale de la red.
7. Para las siguientes interacciones, se recomiendan realizar pruebas guiadas de salto múltiple por cada protocolo de enrutamiento, así podemos tener una mejor comparación con respecto a los saltos que debe realizar un paquete de información dentro de la red tipo malla, para lo cual se requiere una mayor cantidad de nodos, como se muestra en las pruebas del siguiente artículo [60].
 8. Se recomienda, realizar pruebas de acoplamiento en caso que un nodo se desconecte de la red, para comparar el tiempo que tarda la red en proporcionar el mantenimiento adecuado, y si es posible efectuarlo. Para lo cual es necesario un *script* como se muestra en las pruebas de los siguientes artículos [63], [60]
 9. Si se desea implementar acceso a internet dentro de este prototipo 107, se presentan diferentes opciones las cuales pueden implementarse:
 - a) Uso de una red existente (Municipal o Privada).
 - Conexión física, Figura 108.
 - Conexión inalámbrica, Figura 109.
 - b) Uso de una conexión satélite
 - Conexión física, Figura 110.
 - Conexión inalámbrica, Figura 111.
 - c) Uso de conexión por medio de una red LTE
 - Conexión física, Figura 112.
 - Conexión inalámbrica, Figura 113.
 10. Como se menciona en la parte de resultados, para obtener una mayor aceptación ante la comunidad, es necesario tener prestigio y confianza, por lo cual para organizaciones nuevas es muy complicado, dicho esto una de las soluciones pertinentes, es realizar alianzas con otras organizaciones que manejen proyectos a los alrededores e incluir el proyecto dentro de las funcionalidades de la alianza.
 11. En cuanto a la usabilidad de la aplicación educativa, se recomienda realizar un acercamiento más profundo hacia los usuarios finales del producto previo al desarrollo del

prototipo, para conocer mejor cada uno de los aspectos relacionados a ellos que se definen en el *framework* ABCS: el aspecto antropométrico, de comportamiento, cognitivo y social. Esto ayudará a reducir no solo a mejorar la usabilidad del producto, sino que también ayudará a acercar la tecnología a comunidades en las que el contacto con esta misma ha sido mínimo.

12. Relacionado a esto, también se recomienda que las pruebas finales se realicen en un ambiente lo más cercano y similar posible al ambiente en que se instalará la red *mesh*, para confirmar siempre que el acercamiento tomado durante el desarrollo del proyecto será óptimo ante las circunstancias y resiliente.
13. Se recomienda hacer énfasis en el desarrollo de un sistema de navegación que sea sencillo para el usuario final, pero que al mismo tiempo pueda notificarle en todo momento en qué estado se encuentra y cómo puede regresar a un estado anterior. En vista de los resultados obtenidos en las pruebas, se recomienda utilizar colores y figuras llamativas para estos comandos de navegación, puesto que el problema principal que se tuvo en esta parte fue que no se encontraron las opciones de navegación en las vistas.
14. Se recomienda habilitarle a la herramienta *rsync* la opción para utilizar *ssh* ya que permite sincronizar los archivos de una manera mucho más segura, debido al cifrado de la información.
15. Si se desea continuar con el desarrollo del procesador de archivos Markdown de la enciclopedia en línea desarrollada en este trabajo, la siguiente funcionalidad podría implementar la descarga de vídeos mediante los enlaces en el texto.
16. Para mejorar la escalabilidad del sistema distribuido presentado en este trabajo de graduación, se podría idear una manera en que sea posible escalar el nodo maestro en una manera horizontal.
17. En la implementación de la interfaz de programación de aplicaciones de este trabajo en la petición para crear archivos Markdown, se podría mejorar considerablemente la seguridad comprobando que el archivo que el cliente ha proveído es en verdad un archivo Markdown y no un archivo malicioso.
18. Si se desea mejorar considerablemente la calidad de los sistemas implementados en este trabajo, se tiene que trabajar más en el área de seguridad. Debido a todo el tráfico de datos que se da en los sistemas distribuidos.
19. Para mejor rendimiento en el servidor central de la red se recomienda utilizar como mínimo una Raspberry Pi 4.
20. Para obtener un alto nivel de disponibilidad a nivel de servicio se recomienda utilizar más de una Raspberry para el manejo de instancias del servidor web en caso de fallos.
21. Es recomendable que los paneles solares sean instalados y mantenidos por un profesional en el área.
22. Se recomienda que el acercamiento hacia una comunidad rural se haga mediante una alianza con una organización que previamente haya implementado otros proyectos en esa comunidad.

23. En casos de fallo del equipo de la red lo más recomendable es cambiar el equipo en lugar de tratar de repararlo. Puede resultar más costoso reparar que reemplazar.

-
- [1] L. Álvarez, *Acceso a tecnología 4G avanza lento en el país*. dirección: <https://elperiodico.com.gt/inversion/2018/12/01/acceso-a-tecnologia-4g-avanza-lento-en-el-pais/>.
 - [2] MINEDUC. (2016). Anuario Estadístico de la Educación en Guatemala, dirección: <http://estadistica.mineduc.gob.gt/anuario/data/anuario/2016/data/Resultado.htm?Nivel=45&Depto=00&Pob=3&Inf=2&Sector=0&Desgloce=3&Submit=Mostrar+cuadro>.
 - [3] P. Duarte, *Infraestructura escolar en Guatemala*. dirección: https://www.deguate.com/artman/publish/educa_articulos/relacion-de-la-infraestructura-escolar-con-el-rendimiento-academico.shtml.
 - [4] PNUD. (s.f). Human Development Index and its components, dirección: <http://hdr.undp.org/en/composite/HDI>.
 - [5] Instituto Nacional de Estadística. (2014). Caracterización, dirección: <https://www.ine.gob.gt/sistema/uploads/2014/02/26/L5pNHMXzxy5FFWmk9NHCrK9x7E5Qqvvy.pdf>.
 - [6] "Think with Google". (2018). Guatemala: las claves de la digitalización, dirección: <https://www.thinkwithgoogle.com/intl/es-419/insights/guatemala-las-claves-de-la-digitalizacion/>.
 - [7] RACHEL. (2019). Our Story, dirección: <https://worldpossible.org/story>.
 - [8] C. y. J. G. Gall, *U.S. Promotes Network to Foil Digital Spying*. New York, USA: The New York Times, 2014.
 - [9] M. G. y. v. S. G. Backens J., «A Rural Implementation of a 52 Node Mixed Wireless Mesh Network in Macha, Zambia», *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 38, 2010.
 - [10] P. D. Ishmael J. Bury S. y R. N.J.P., «Deploying Rural Community Wireless Mesh Networks», *IEEE Internet Computing*, vol. 12, 2008.

- [11] UNESCO. (). Uso de TIC's en la Educación, dirección: <http://unesco Guatemala.org/comunicacion-e-informacion/acceso-al-conocimiento/uso-de-tics-en-la-educacion/>.
- [12] A. Stix, *Social Studies Strategies for Active Learning*. California, USA: Shell Education, 2004.
- [13] M. Bernárdez, *Diseño Producción Implementación de e-Learning*. Indiana, USA: AuthorHouse, 2007.
- [14] J. S. et al, *Guide to Web Applications and Platform Architectures*. New York, USA: Springer Science y Business Media, 2013.
- [15] S. D. y M. Pinto, *The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws*. Indiana, USA: Wiley Publishing Inc., 2011.
- [16] A. Lastovetska, *Native App Development vs. Hybrid and Web App Building*, 2018. dirección: <https://mlsdev.com/blog/native-app-development-vs-web-and-hybrid-app-development>.
- [17] B. Baxley, *Making the Web Work*. New York, USA: New Riders, 2002.
- [18] A. Goel, *Why should you learn Angular in 2019*, 2018. dirección: <https://hackr.io/blog/why-should-you-learn-angular-in-2019>.
- [19] —, *Top 10 Web Development Frameworks in 2019*, 2019. dirección: <https://hackr.io/blog/top-10-web-development-frameworks-in-2019>.
- [20] E. Wohlgetan, «Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js», Hamburg University of Applied Sciences, 2018.
- [21] A. Fedosejev, *React.js Essentials*. Birmingham, Reino Unido: Packt Publishing Ltd, 2015.
- [22] Vuejs, *What is Vue.js*. dirección: <https://vuejs.org/v2/guide/>.
- [23] A. y. K. M. Kyriakidis, *The Majesty of Vue.js*. Birmingham, UK: Packt Publishing, 2016.
- [24] P. Beijer y T. de Klerk, *IT Architecture: Essential Practice for IT Business Solutions*. España, Madrid: RA-MA, 2018.
- [25] D. Roldán, P. Valderas y V. Torres, *Microservicios: Un enfoque integrado*. España, Madrid: RA-MA, 2018.
- [26] R. Somasundaram, *Git: version control for everyone*. Birmingham, UK: Packt Publishing, 2013.
- [27] J. Wachsman, *Git Flow A Successful Git Branching Model*. Leanpub, 2015.
- [28] P. Hammant. (2019). Trunk Based Development, dirección: <https://trunkbaseddevelopment.com/>.
- [29] R. P. y J. Vainikainen, *ABCs of z/OS System Programming*. IBM Redbooks, 2012.
- [30] W. V. et al, *The Definitive Guide to HTML5 WebSocket*. New York, USA: apress, 2013.
- [31] T. Cadenhead, *Socket.IO Cookbook*. Birmingham, UK: Packt Publishing, 2015.

- [32] L. S. y S. Romano, *Real-Time Communication with WebRTC: Peer-to-Peer in the Browser*. California, USA: O'Reilly Media, Inc, 2014.
- [33] MDN. (2019). Signaling and Video Calling, dirección: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Signaling_and_video_calling.
- [34] A. S. Tanenbaum y M. V. Steen, *Distributed systems Principles and paradigms*. Prentice-hall, 2017.
- [35] A. Tridgell, *Efficient Algorithms for Sorting and Synchronization*. The Australian National University: The Australian National University, 1999.
- [36] T. G. Project. (2019). Documentation, dirección: <https://golang.org/doc/>.
- [37] Uber inc. (2019). Uber golang, dirección: <https://github.com/uber-go>.
- [38] Netflix inc. (2019). Netflix Go, dirección: <https://github.com/Netflix?language=go>.
- [39] Go Team. (2019). Effective Go, dirección: https://golang.org/doc/effective_go.html#goroutines.
- [40] SQLite. (2019). About SQLite, dirección: <https://www.sqlite.org/about.html>.
- [41] —, (2019). 35 % Faster Than The Filesystem, dirección: <https://www.sqlite.org/fasterthanfs.html>.
- [42] —, (2019). Size Of The SQLite Library, dirección: <https://www.sqlite.org/footprint.html>.
- [43] F. e. a. Ritter, *User-Centered Systems Design*. Springer, 2014, pág. 460.
- [44] D. Normal, *The design of everyday things*. Basic Books, 2013, pág. 368.
- [45] HHS. (2017), dirección: <https://dca.gob.gt/noticias-guatemala-diario-centro-america/guatemala-con-53-millones-de-usuarios-en-facebook/> (visitado 13-05-2019).
- [46] S. Brooke. (2018), dirección: <https://uxplanet.org/how-to-conduct-a-usability-test-in-six-steps-from-start-to-finish-4082e8d57858> (visitado 15-09-2019).
- [47] D. Rubin J. y Chisnell, *The Process of Conducting a test*. Wiley Publishing, 2008, págs. 65-288.
- [48] J. Nielsen, *Usability Testing*. Morgan Kaufman, 1993, págs. 165-192.
- [49] A. Dix, *Heuristics*. Person Education, 2004, pág. 861.
- [50] M. Cone. (2019). Getting started with Markdown Language, dirección: <https://www.markdownguide.org/getting-started/> (visitado 05-11-2019).
- [51] A. D. et al, *Medidas numéricas de estadística descriptiva*. Cengage Learning, 2012, pág. 2.
- [52] A. S. Tanenbaum, *Computer Networks*. USA: Pearson, 1981.
- [53] H. E. y K. Leung, *Wireless Mesh Networks*. Winnipeg, Canadá: Springer, 2017.
- [54] IGF, *Declaration on Community Connectivity*. dirección: https://www.intgovforum.org/multilingual/index.php?q=filedepot_download/4391/1316.
- [55] R. R. Roy, *Mobile Ad Hoc Networks*. Boston, MA: Springer, 2010.

- [56] M. J. Vincent Zimmer Jiming Su y S. Reinauer, *Embedded Firmware Solutions: Development Best Practices for the Internet of Things*. USA: Apress, 2015.
- [57] S. software decision, *What is DD-WRT?* Dirección: https://forum.dd-wrt.com/wiki/index.php/What_is_DD-WRT%3F.
- [58] AdvancedTomato, *AdvancedTomato*. dirección: <https://advancedtomato.com/>.
- [59] hauke, *About*. dirección: <https://openwrt.org/about>.
- [60] F. J. C. Berga, «Anonimato en redes ad-hoc mediante integración de los protocolos HIP y OLSR», *Murdoch University*, vol. 1, 2009.
- [61] *Open-Mesh*. dirección: <https://www.open-mesh.org/projects/open-mesh/wiki/BranchesExplained>.
- [62] S. F. Benjamin Sliwa y C. Wietfeld, «Performance Evaluation and Optimization of B.A.T.M.A.N. V Routing for Aerial and Ground-based Mobile Ad-hoc Networks», *Dortmund University*, vol. 1, 2019.
- [63] J. C.-P. W. M. Abolhasan Brett Hagelstein, «Real-world performance of current proactive multihop mesh protocols», *University of Wollongong*, vol. 1, 2009.
- [64] K. D. M, «Performance Evaluation of Mesh Protocols in Real Time Mesh Test Bed», *MS Ramaiah Institute of Technology*, vol. 2, 2015.
- [65] L. N. Axel Neumann Ester López, «Evaluation of mesh routing protocols for wireless community networks», *elsevier*, vol. 1, 2015.
- [66] T. K. David Murray Michael Dixon, «An Experimental Comparison of Routing Protocols in Multi Hop Ad Hoc Networks», *Murdoch University*, vol. 1, 2013.
- [67] CISCO. (2019). *Wireless, LAN (WLAN)*, dirección: https://www.cisco.com/c/es_mx/tech/wireless-2f-mobility/wireless-lan-wlan/index.html.
- [68] F. Andreu, I. Pellejero y A. Lesta, *Fundamentos y Aplicaciones De Seguridad En Redes WLAN: De La teoría a La práctica*. Barcelona, España: Marcombo, 2006.
- [69] Linksys. (2019). *What is an Access Point and How is it Different from a Range Extender?*, dirección: <https://www.linksys.com/us/r/resource-center/what-is-a-wifi-access-point/>.
- [70] Cisco. (2018). *Wireless Security Threats in an Enterprise Environment*, dirección: https://documentation.meraki.com/MR/Monitoring_and_Reporting/Air_Marshal.
- [71] Rootsh3ll. (2019). *CAPTIVE PORTAL: The Definitive Guide*, dirección: <https://rootsh3ll.com/captive-portal-guide/>.
- [72] Sombreronegro.cl. (2019). *Alta disponibilidad*, dirección: <https://www.netlinux.cl/soluciones-it/alta-disponibilidad>.
- [73] Mozilla.org. (2019). *What is a web server?*, dirección: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server.
- [74] Nginx. (2019). *What is NGINX?*, dirección: <https://www.nginx.com/resources/glossary/nginx/>.
- [75] —, (2019). *What Is NGINX Web Application Firewall?*, dirección: <https://www.nginx.com/products/nginx-waf/>.

- [76] Techopedia. (2019). *What are Network Security Protocols?*, dirección: <https://www.techopedia.com/definition/29036/network-security-protocols>.
- [77] A. Gómez, *Enciclopedia de la Seguridad Informática. 2ª edición*. Madrid, España: RAMA, S.A., 2014.
- [78] A. González, D. Beltrán y E. Fuentes. (2016). *PROPUESTA DE PROTOCOLOS DE SEGURIDAD PARA LA RED INALÁMBRICA LOCAL DE LA UNIVERSIDAD DE CIENFUEGOS*, dirección: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202016000400017.
- [79] J. Luaces. (2019). *Seguridad en redes inalámbricas de área local (WLAN)*, dirección: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/18804/6/jluacesTFC0113memoria.pdf>.
- [80] W. Méndez, D. Mosquera y E. Rivas. (2015). *Vulnerabilidad de protocolos de encriptación WEP, WPA y WPA2 en redes inalámbricas con plataforma Linux*, dirección: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0123-921X2015000500007.
- [81] C. Bollmann. (2013). *Guatemala: presentan estudio sobre niveles socioeconómicos*, dirección: <https://www.estrategiaynegocios.net/lasclavesdeldia/562566-330/guatemala-presentan-estudio-sobre-niveles-socioeconomicos>.
- [82] O. R. R. MIRANDA, «Guía Para La Gestión De Proyectos Priorizados De Infraestructura, Ante Los Consejos De Desarrollo Urbano Y Rural, Y Cooperantes Internacionales.», *Universidad de San Carlos de Guatemala*, vol. 17, 2013.
- [83] J. Sauro. (2011), dirección: <https://measuringu.com/task-completion/> (visitado 16-09-2019).
- [84] M. Meyer, «Microservices Are a Keystone of Continuous Delivery: Continuous Delivery & Automation», *The DZone Guide to DevOps*, vol. 4, 2017.
- [85] Internet Engineering Task Force (IETF), *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*, 2014. dirección: <https://tools.ietf.org/html/rfc7231>.
- [86] C. Hahmann. (2013). *La sociedad guatemalteca estaría dividida en siete grupos.*, dirección: <https://www.estrategiaynegocios.net/lasclavesdeldia/562566-330/guatemala-presentan-estudio-sobre-niveles-socioeconomicos>.
- [87] W. Spark. (2019). *El clima promedio en Ciudad de Guatemala*, dirección: <https://es.weatherspark.com/y/11693/Clima-promedio-en-Ciudad-de-Guatemala-Guatemala-durante-todo-el-ano>.
- [88] K. Energy. (2019). *Kingo Energy.*, dirección: <http://www.kingoenergy.com/>.
- [89] R. Cohen. (2018). Why you can have millions of Goroutines but only thousands of Java Threads, dirección: <https://rcoh.me/posts/why-you-can-have-a-million-go-routines-but-only-1000-java-threads/>.
- [90] Uber inc. (2019). Uber Zap, dirección: <https://github.com/uber-go/zap>.

11.1. Diseño de una *red mesh*

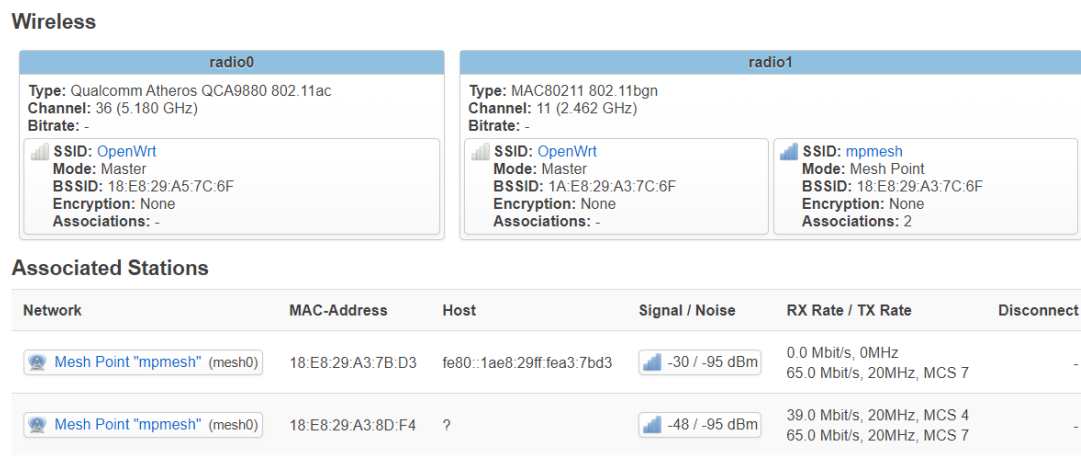


Figura 97: Demostración gráfica de la conectividad de los nodos a la red malla "mpmesh"

```
root@OpenWrt:~# vi /etc/config/network
config interface 'loopback'
    option ifname 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fdbb:7cd1:a592::/48'

config interface 'lan'
    option type 'bridge'
    option ifname 'eth0'
    option proto 'static'
    option ipaddr '192.168.0.15'
    option netmask '255.255.255.0'
    option ip6assign '60'
```

Figura 98: Configuración de *Network*, donde se establece una dirección estática para la interfaz de lan.

```

root@OpenWrt:~# vi /etc/config/wireless
config wifi-device 'radio0'
    option type 'mac80211'
    option channel '36'
    option hwmode '11a'
    option path 'pci0000:00/0000:00:00.0'
    option htmode 'VHT80'

config wifi-iface 'default_radio0'
    option device 'radio0'
    option network 'lan'
    option mode 'ap'
    option ssid 'OpenWrt1'
    option encryption 'none'

config wifi-device 'radio1'
    option type 'mac80211'
    option channel '11'
    option hwmode '11g'
    option path 'platform/ahb/18100000.wmac'
    option htmode 'HT20'

config wifi-iface 'default_radio1'
    option device 'radio1'
    option network 'lan'
    option mode 'ap'
    option ssid 'OpenWrt1'
    option encryption 'none'

```

Figura 99: Configuración de las antenas emisoras de wifi, del equipo, cada antena puede emitir una red diferente.

```

config wifi-iface 'mesh0'
    option device 'radiol1'
    option ifname 'mesh0'
    option network 'nwi_mesh0'
    option mode 'mesh'
    option mesh_fwding '0'
    option mesh_id 'mpmesh'
~

```

Figura 100: Configuración para crear una red que soporte el tipo malla.

```

config interface 'bat0'
    option proto 'batadv'
    option routing_algo 'BATMAN_IV'
    option aggregated_ogms 1
    option ap_isolation 0
    option bonding 0
    option fragmentation 1
    #option gw_bandwidth '10000/2000'
    option gw_mode 'off'
    #option gw_sel_class 20
    option log_level 0
    option orig_interval 1000
    option bridge_loop_avoidance 1
    option distributed_arp_table 1
    option multicast_mode 1
    option network_coding 0
    option hop_penalty 30
    option isolation_mark '0x00000000/0x00000000'

```

Figura 101: Configuración de la interfaz de control para B.A.T.M.A.N. dentro de la configuración *network*.

```

config interface 'nwi_mesh0'
    option mtu '2304'
    option proto 'batadv_hardif'
    option master 'bat0'

```

Figura 102: Configuración de la interfaz encargada de establecer conexión de forma "física" bajo el control bat0 dentro de la configuración. *network*.

Megaproyecto red en malla
[Octubre/2019]
Consentimiento informado para participación en el proyecto

Usted/su comunidad/aldea/etc. ha sido invitada a participar en el proyecto megaproyecto red en malla. Le pedimos que tome el tiempo para leer el siguiente consentimiento informado para tener mayor información. Si tiene dudas al terminar de leer, pregunte a un miembro del equipo.

1. Propuesta de participación conjunta

Como parte de trabajo de graduación, estudiantes de 5to año carrera Licenciatura en Ingeniería en ciencia de la computación y tecnologías de la información de la Universidad del Valle de Guatemala estamos realizando el proyecto red en malla que consiste en el diseño e implementación de una red tipo malla. Su municipio/aldea/comunidad fue seleccionado para formar parte del proyecto por lo que le invitamos a conocer más acerca del mismo.

2. Propósitos

El propósito principal de este proyecto es crear una red tipo malla alrededor de varias localidades, que de acceso a diferentes servicios de comunicación sin los servicios de cableado y redes físicas. Como objetivo inmediato deseamos mejorar las condiciones de la comunicación dentro de la comunidad.

3. Criterios de selección

Este municipio/aldea/comunidad ha sido seleccionado debido a que es una comunidad sin medios de comunicación tecnológicos.

4. Duración

Este proyecto tendrá una duración de 4 meses. La fase inicial consistirá en la construcción de una red maya con un servicio de comunicación y se llevará a cabo de 7 de octubre 2019 a 03 de noviembre 2019.

5. Técnicas y procedimientos

Se utilizará la metodología ágil, ya que se realizará por medio de hitos.

6. Posibles riesgos e inconvenientes

Se puede especificar que no existe un riesgo físico pero que por ejemplo el uso de la red tendrá ciertos límites.

El equipo solo podrá utilizarse para comunicación dentro de las comunidades afectadas. No podrá usarse para establecer una conexión a internet de forma permanente por lo cual no podrán hacer uso de redes sociales.

Identificamos como otro posible riesgo que, si forma parte del proyecto y dado que la divulgación de resultados se realizará en la Universidad del Valle, los datos de la comunidad como del uso que hagan del equipo podrá conocerse por otras personas que no necesariamente formen parte del proyecto.

Sin embargo, tomaremos todas las medidas necesarias para asegurar el bienestar de las personas participantes. En ningún momento este proyecto representa riesgos que dañen la salud o integridad de las personas.

Figura 103: Consentimiento informado

La participación en este proyecto es voluntaria, no existen consecuencias de ningún tipo si no desea participar.

7. Beneficios

- Acceso a red tipo malla – Esta establece conexión entre nodos, a los cuales a su vez se establece conexión a los diferentes usuarios. Al estar todos los nodos conectados entre sí, esta red puede manejar situaciones donde un nodo deja de funcionar, los otros proceden a dar soporte hasta que se le dé el mantenimiento adecuado, dando a lugar que el área afectada no quede sin el servicio de la red.
- Técnicación de personas de la comunidad – Se procederá a involucrar a la comunidad por medio de talleres informativos, con el fin que las personas de la comunidad puedan dar mantenimiento a las redes.
- Posibles usos para acceder a información o posibles – La red permite montar varios servicios en ella, tales como servicios educativos, culturales, sociales, económicos, salud y tecnológicos.
- Además, el beneficio también es la contribución a generar conocimiento porque no es un tema muy conocido en Guatemala y por ende, la ejecución del proyecto podrá contribuir para que, en el futuro, otras personas/comunidades que no tienen acceso a internet o a fuentes de información puedan utilizar esta red.

8. Compensación

Este proyecto no otorgará ningún tipo de compensación económica. El servicio tendrá un costo mensual mínimo por un tiempo específico para recuperar el costo inicial, y así proceder a equipar otra comunidad vecina, una vez el tiempo termina se hará entrega del equipo y que el mismo pasa a ser parte de la propiedad comunitaria. En caso la comunidad deba pagar explicar que el pago será proporcional. (Agregar el costo mensual del servicio y la cantidad de meses)

9. Términos de confidencialidad y alcances de la divulgación de la información

Toda la información que nos sea compartida será utilizada con fines estrictamente académicos y en función del proyecto red en malla. Es importante especificar que la investigación con sus resultados podrá ser expuesta en el Campus Central de la UVG o en otros espacios académicos. Sin embargo, se garantiza que los datos personales de las personas involucradas en el proyecto o que hagan uso de la red no serán expuestos.

10. Contactos esenciales



Nombre	Relevancia	Teléfono	Correo electrónico
Andres Oliva	Estudiante de Ingeniería en las ciencias de computación y tecnologías de la información/investigador, soporte, desarrollador	42158802	oli12149@uvg.edu.gt
Sebastián Galindo	Estudiante de Ingeniería en las ciencias de computación y	59301990	gal15452@uvg.edu.gt

Figura 104: Consentimiento informado

	tecnologías de la información/investigador, soporte		
Rodrigo Arriaza	Estudiante de Ingeniería en las ciencias de computación y tecnologías de la información/ soporte, desarrollador	30144106	arr15334@uvg.edu.gt
Rodrigo Barrios	Estudiante de Ingeniería en las ciencias de computación y tecnologías de la información/ soporte, desarrollador	42167490	bar15009@uvg.edu.gt
José Custodio	Estudiante de Ingeniería en las ciencias de computación y tecnologías de la información/soporte, desarrollador	57466715	cus15220@uvg.edu.gt
Alejandro Chaclán	Estudiante de Ingeniería en las ciencias de computación y tecnologías de la información/ soporte, desarrollador	30020538	c ha15018@uvg.edu.gt
Douglas Barrios	Catedrático/ Respaldo institucional de UVG		dlbarrios@uvg.edu.gt



Figura 105: Consentimiento informado

Certificación de la persona que obtiene el consentimiento

1. Se le ha informado y entregado un documento a la persona acerca de:
 - Los objetivos del proyecto y el procedimiento para alcanzarlos en determinado tiempo de duración
 - Posibles inconvenientes y riesgos, así como los beneficios y la compensación derivados de su participación en el proyecto
 - Los contactos del proyecto a los que podrá abocarse para más información
2. Cualquier duda respecto al proyecto fue resuelta.

Lugar: _____

Fecha: / /

Nombre y firma de la persona que obtiene el consentimiento

Nombre y firma de encargado del proyecto que extiende el consentimiento

Figura 106: Consentimiento informado

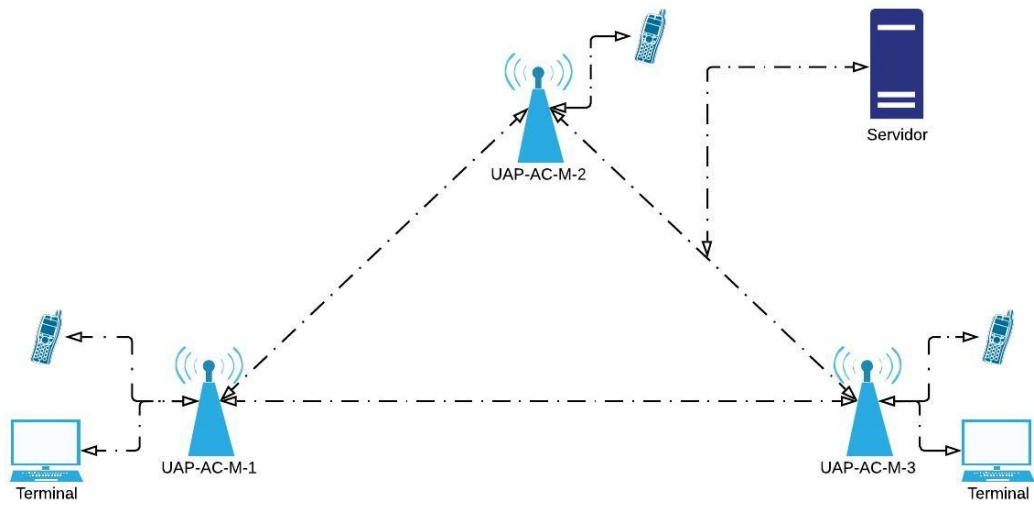


Figura 107: Diagrama red tipo malla

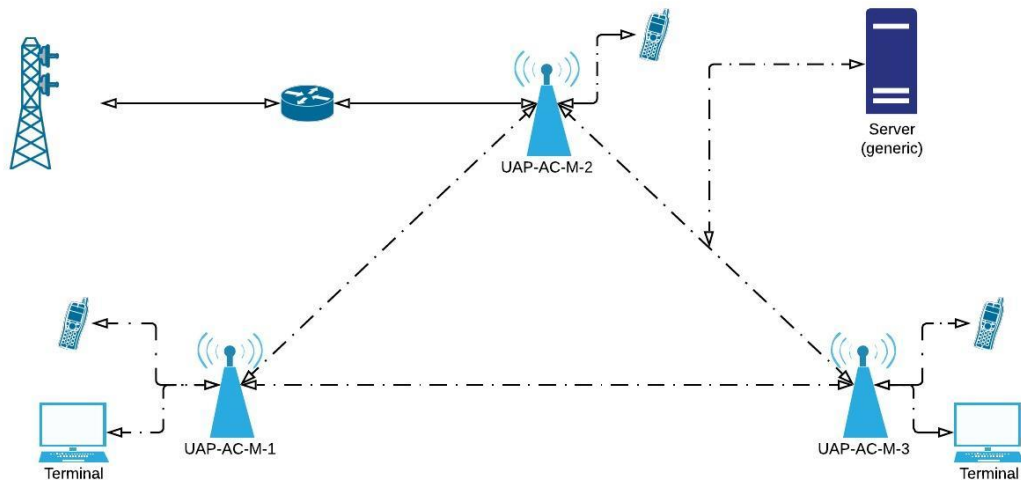


Figura 108: Diagrama red tipo malla con conexión física a internet, por medio de una red existente

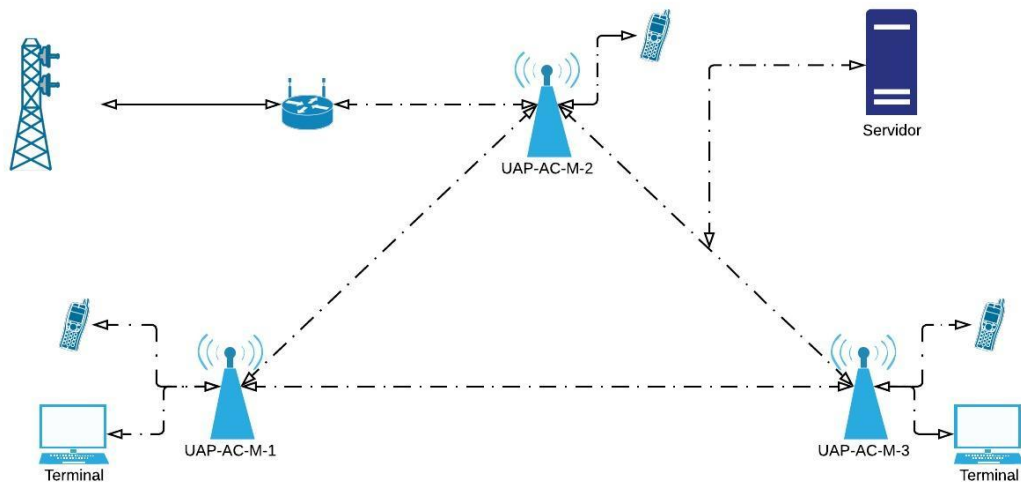


Figura 109: Diagrama red tipo malla con conexión inalámbrica a internet, por medio de una red existente

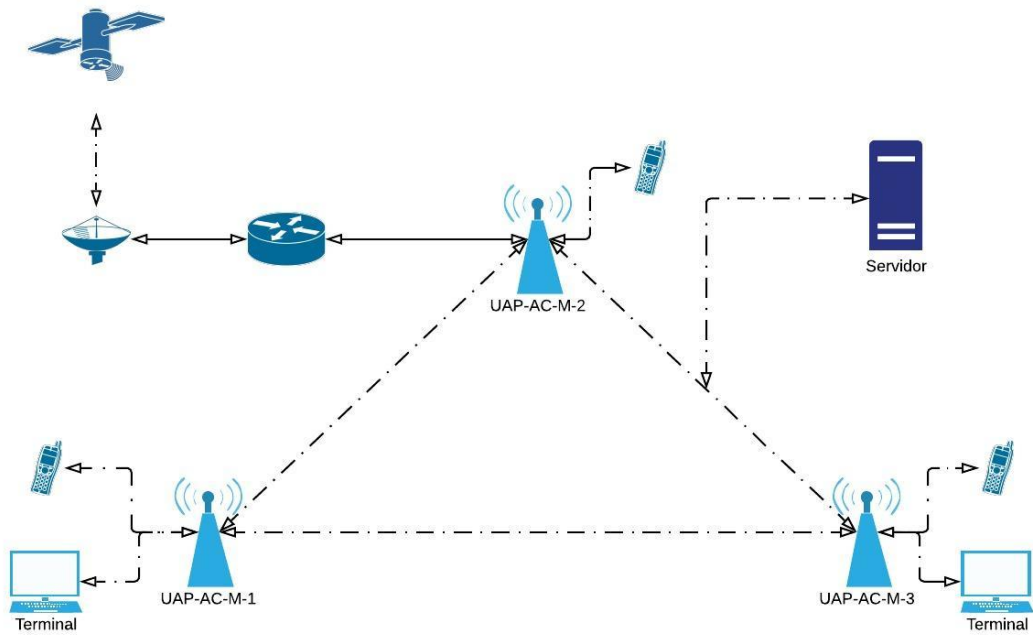


Figura 110: Diagrama red tipo malla con conexión física a internet, por medio de una red satelital

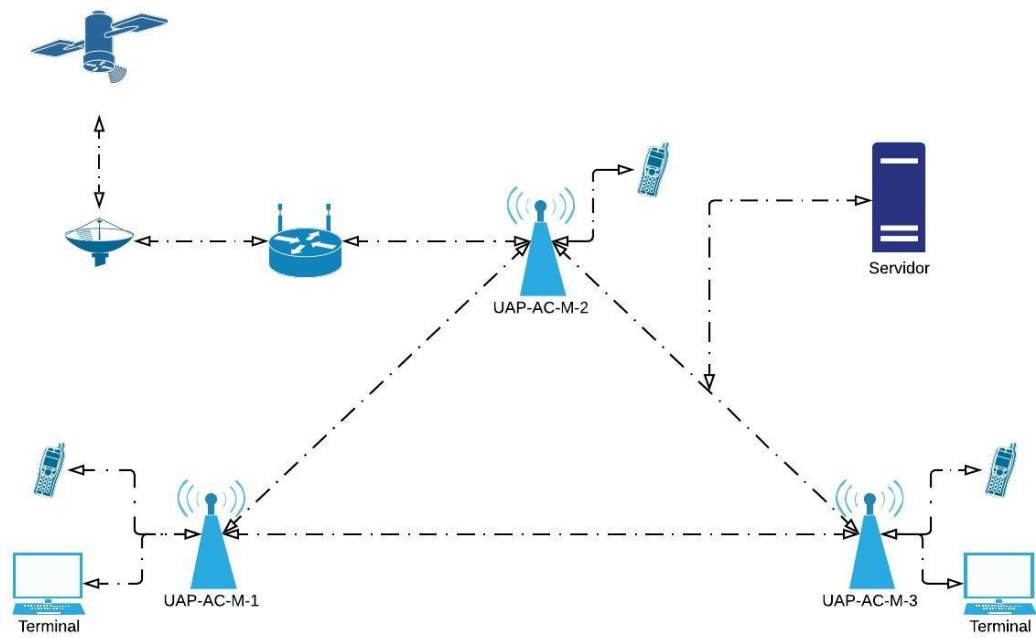


Figura 111: Diagrama red tipo malla con conexión inalámbrica a internet, por medio de una red satelital

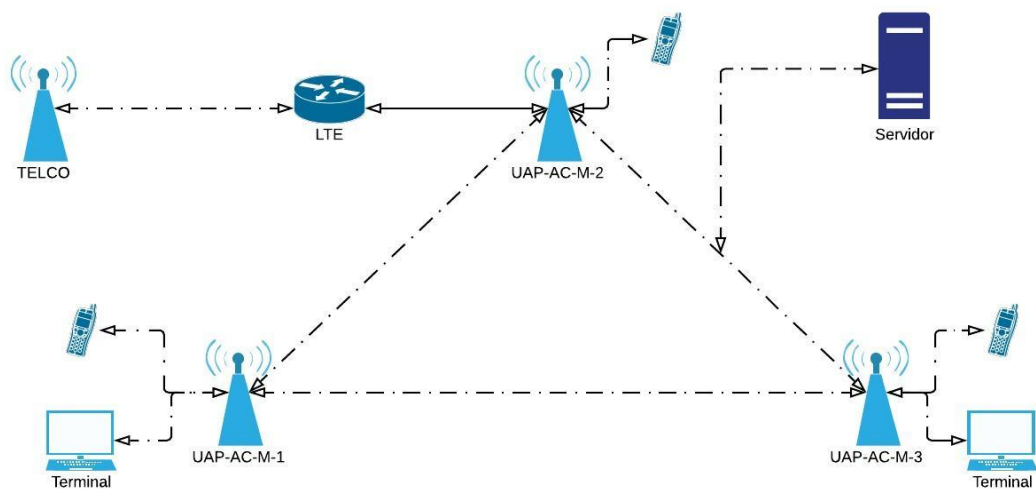


Figura 112: Diagrama red tipo malla con conexión física a internet, por medio de una red LTE

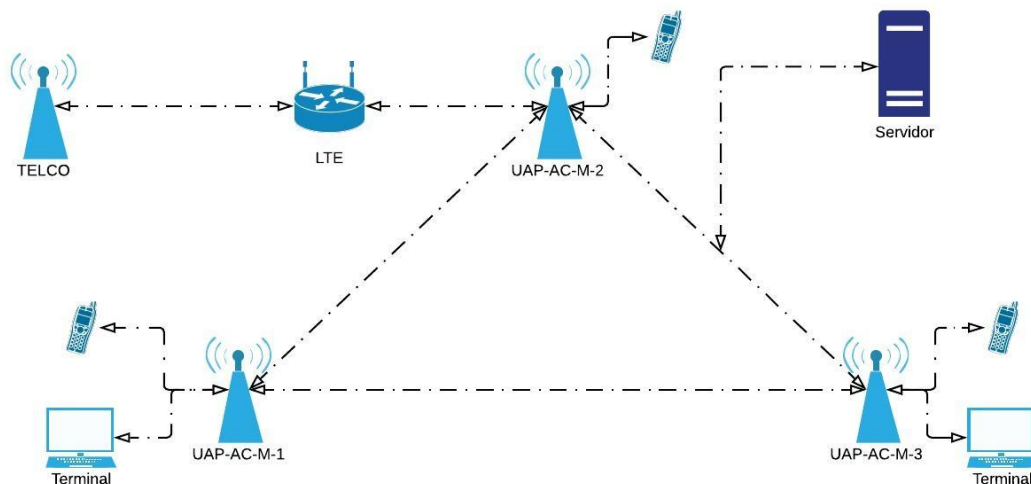


Figura 113: Diagrama red tipo malla con conexión inalámbrica a internet, por medio de una red LTE

11.2. Realización de la interfaz gráfica para la aplicación educativa

11.2.1. Guion para introducción a las pruebas de usabilidad

Se utilizó el siguiente guion para informar a los usuarios acerca de la prueba que se llevaría a cabo con ellos:

"Buen día, nosotros somos estudiantes de la Universidad del Valle de Guatemala y actualmente cursamos el último año de la carrera de Ingeniería en Ciencia de la Computación y Tecnologías de la Información. El motivo de esta reunión es realizar una prueba de usabilidad con usted para evaluar nuestro proyecto de graduación, el cual consiste en el prototipo de una aplicación educativa para comunidades rurales de Guatemala. En esta aplicación, maestros de las escuelas locales de dichas comunidades podrán publicar material educativo que ayudará a reforzar los temas vistos en clase, el cual estará disponible para que pueda ser consultado por sus alumnos.

A continuación, se le mostrará un prototipo inicial de la aplicación y se le pedirá que realice una serie de tareas dentro del prototipo, únicamente tomando como guía la descripción que se le dio de la aplicación, la descripción de la tarea y la información disponible dentro del mismo prototipo. Siéntase libre de comunicarnos cualquier comentario o sugerencia acerca del prototipo durante la prueba. El principal objetivo de esta prueba no es evaluarlo a usted como usuario, sino que es evaluar qué tan usable es el prototipo, para determinar posibles mejoras en un futuro. No se divulgará ningún tipo de información personal acerca de usted. Muchas gracias por su cooperación."

11.2.2. Resultados individuales de las pruebas de usabilidad

A continuación, se presentan tablas con los resultados de las pruebas de usabilidad realizadas con el prototipo de la aplicación educativa. Los números de las tareas son los mismos que los que se encuentran enumerados en la sección 5.3. Los datos de los usuarios evaluados son anónimos, por lo que se les identifica con un número (usuario 1, usuario 2, etc).

Cuadro 18: Resultados de la prueba realizada al usuario 1

Tarea	Estado de la tarea	Tiempo (s)	Cantidad de pasos
1	Completa	9	4
2	Completa	18	4
3	Completa	33	6
4	Completa	33	5
5	Completa	6	2
6	Completa	10	1
7	Completa	3	1
8	Completa	4	1
9	Completa	5	1

Cuadro 19: Resultados de la prueba realizada al usuario 2

Tarea	Estado de la tarea	Tiempo (s)	Cantidad de pasos
1	Completa	11	4
2	Completa	8	4
3	Completa	47	6
4	Completa	25	5
5	Completa	4	2
6	Completa	10	1
7	Incompleta	33	4
8	Completa	2	1
9	Completa	3	1

Cuadro 20: Resultados de la prueba realizada al usuario 3

Tarea	Estado de la tarea	Tiempo (s)	Cantidad de pasos
1	Completa	11	4
2	Completa	11	4
3	Completa	56	6
4	Completa	31	5
5	Completa	7	2
6	Completa	6	1
7	Completa	4	1
8	Completa	4	1
9	Completa	3	1

Cuadro 21: Resultados de la prueba realizada al usuario 4

Tarea	Estado de la tarea	Tiempo (s)	Cantidad de pasos
1	Completa	11	4
2	Completa	8	4
3	Completa	25	6
4	Completa	22	5
5	Completa	8	2
6	Completa	5	1
7	Completa	2	1
8	Completa	2	1
9	Completa	2	1

ACL Es una lista que maneja los permisos adjuntos a un objeto que especifican qué usuarios tienen acceso a ese objeto y las operaciones que puede realizar. 46

DevOps Es una práctica de ingeniería de software que tiene como objetivo unificar el desarrollo de software y la operación del software. 50

DNS *Domain Name System*, es un sistema que utiliza el internet para realizar la traducción entre nombres alfabéticos y direcciones IP numéricas. 47, 48

DoS *Denial of Service*, es un tipo de ataque en el cual los atacantes intentan prevenir el legítimo acceso de los usuarios de un servicio mediante la saturación del mismo. 84

ethernet Es un sistema para conectar varios sistemas informáticos para formar una red de área local, con protocolos para controlar el paso de información y evitar la transmisión simultánea para dos o más sistemas. 46

HTTP *Hypertext Transfer Protocol*, es un protocolo de transferencia de archivos, tales como texto, imágenes, video y otro tipo de contenido multimedia a lo largo de web. 49

hub Es un punto de conexión común para dispositivos en una red. Son comúnmente utilizados para conectar segmentos de una red LAN. 46

IRR Un IRR o *Information Risk Register* es una manera estructurada de dejar constancia de registrar los riesgos de seguridad de la información. Este documento consiste en el registro de los potenciales riesgos, probabilidad y su potencial impacto. Se presenta en forma de una tabla en el cual se van registrando los datos mencionados anteriormente. 82

JavaScript Es un lenguaje de programación comúnmente utilizado en el desarrollo web. 49

JSON *JavaScript Object Notation* es un formato de texto sencillo para el intercambio de datos. 81

- ONG** Siglas para organización no gubernamental, es una institución sin ánimo de lucro que no depende del gobierno y realiza actividades de interés social. 77, 78
- Reverse Proxy** Es un tipo de servidor proxy que recolecta los recursos que un cliente solicita desde uno o más servidores. 50
- router** Es un dispositivo que se encarga del direccionamiento de paquetes a lo largo de una red. Es utilizado tanto en redes LAN como WLAN. 46
- Sistema de alimentación ininterrumpida** Es un sistema que mediante la utilización de baterías, durante un apagón eléctrico, se encarga de la distribución de energía eléctrica durante un tiempo limitado el cuál está definido por la capacidad de las baterías y la cantidad de dispositivos que tenga conectados. 87
- Streaming** Es un método de transmisión o recepción de datos (especialmente material de audio y video) sobre una red de computadoras de una manera estable y con un flujo continuo, permitiendo que el material se empiece a reproducir mientras el resto de los datos son recibidos. 78
- switch** Es un hardware de red que conecta dispositivos en una red informático mediante la conmutación de paquetes para recibir y reenviar datos al dispositivo destino. 46
- TCP** *Transmission Control Protocol*, es un protocolo que define el estándar sobre cómo establecer y mantener una conversación entre distintos dispositivos. 49
- UDP** *User Datagram Protocol*, es una alternativa para el protocolo TCP para la transmisión de paquetes entre dispositivos. 49
- UniFi** Modelo de antenas de la marca Ubiquiti para el manejo e implementación de sistemas Wi-Fi. 80
- Wi-Fi** Es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos. Es una marca registrada de la Alianza Wi-Fi que cumple con los estándares 802.11. 46, 48, 50–52
- Wiki** Sistema de trabajo informático utilizado en los sitios web que permite a los usuarios modificar o crear su contenido de forma rápida y sencilla. 77, 81