

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

### **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UN SISTEMA WEB INTEGRADO QUE PERMITA BRINDAR SOPORTE MEDIANTE VOZ Y CHAT**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y REDES DE LA INFORMACIÓN**

**EDUARDO ANDRÉS GALLEGOS VALDEZ**  
**edugallegos@gmail.com**

**DIRECTOR: ING. XAVIER CALDERÓN HINOJOSA, MSc.**  
**xavier.calderon@epn.edu.ec**

**Quito, Junio 2015**

## DECLARACIÓN

Yo, Eduardo Andrés Gallegos Valdez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

---

**Eduardo Andrés Gallegos Valdez**

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Eduardo Andrés Gallegos Valdez bajo mi supervisión.

---

**Ing. Xavier Calderón Hinojosa, MSc.**  
**DIRECTOR DEL PROYECTO**

## **AGRADECIMIENTOS**

A mi madre por ser el pilar de mi vida, por su apoyo, su ayuda y todo su amor.

A mi padre por todo su esfuerzo en darme cada día lo mejor.

A mi hermana por hacer este camino más fácil.

A mi enamorada por todo su amor, comprensión y apoyo incondicional.

A mi director de proyecto el Ing. Xavier Calderón Msc. por guiarme de una manera tan comprometida en el desarrollo de este proyecto.

Eduardo Andrés Gallegos Valdez



## **DEDICATORIA**

Dedico este trabajo a mi Madre que con todo su amor, paciencia y apoyo ha sido la fuerza que me impulsa a crecer y ser mejor cada día en todos los aspectos de mi vida.

Eduardo Andrés Gallegos Valdez

## CONTENIDO

<b>DECLARACIÓN .....</b>	<b>I</b>
<b>CERTIFICACIÓN .....</b>	<b>II</b>
<b>AGRADECIMIENTOS .....</b>	<b>III</b>
<b>DEDICATORIA .....</b>	<b>IV</b>
<b>CONTENIDO .....</b>	<b>V</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>X</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>XIII</b>
<b>RESUMEN .....</b>	<b>XIV</b>
<b>PRESENTACIÓN .....</b>	<b>XV</b>
<b>CAPÍTULO 1 .....</b>	<b>1</b>
<b>1. MARCO TEÓRICO .....</b>	<b>1</b>
<b>1.1 INTRODUCCIÓN .....</b>	<b>1</b>
<b>1.2 SIP (SESSION INITIAL PROTOCOL)[1] .....</b>	<b>2</b>
1.2.1 INTRODUCCIÓN .....	2
1.2.2 MÉTODOS .....	2
1.2.3 RESPUESTAS .....	3
1.2.4 ELEMENTOS DE UNA RED SIP .....	4
1.2.4.1 AGENTES DE USUARIO .....	4
1.2.4.2 SEVIDORES .....	5
1.2.4.3 SERVICIOS DE LOCALIZACIÓN .....	5
1.2.5 FUNCIONES DE SIP .....	5
1.2.5.1 CONFIGURACIÓN DE SESIÓN .....	6
1.2.5.2 NEGOCIACIÓN DE MEDIOS .....	6
1.2.5.3 TERMINACIÓN Y CANCELACIÓN DE SESIÓN .....	6
<b>1.3 WEBSOCKETS[2][3] .....</b>	<b>7</b>
1.3.1 COMPARATIVA ENTRE WEBSOCKETS Y MÉTODOS ANTERIORES .....	8
<b>1.4 WEBRTC[7] .....</b>	<b>11</b>
1.4.1 MEDIASTREAM API .....	11
1.4.2 PEERCONNECTION API .....	11
1.4.3 DATACHANNEL API .....	11
<b>1.5 SOFTPHONE .....</b>	<b>12</b>
1.5.1 SOFTPHONE WEB O WEBPHONE .....	13

<b>1.6</b>	<b>PUSHER[10]</b> .....	<b>14</b>
<b>1.7</b>	<b>PARSE[11]</b> .....	<b>14</b>
<b>1.8</b>	<b>DESARROLLO WEB</b> .....	<b>15</b>
1.8.1	PHP[12].....	15
1.8.2	FRAMEWORK DE DESARROLLO WEB .....	16
1.8.2.1	Framework MVC[14].....	17
1.8.2.2	Symfony2[15].....	17
1.8.2.3	Doctrine .....	18
<b>1.9</b>	<b>JAVASCRIPT[18]</b> .....	<b>20</b>
1.9.1	JQUERY[20].....	21
<b>CAPÍTULO 2</b> .....		<b>23</b>
<b>2.</b>	<b>DISEÑO DEL SISTEMA INTEGRAL</b> .....	<b>23</b>
<b>2.1</b>	<b>ANÁLISIS DE REQUERIMIENTOS</b> .....	<b>23</b>
2.1.1	CENTRAL TELEFÓNICA.....	24
2.1.1.1	Extensiones.....	25
2.1.1.2	Troncal.....	25
2.1.2	SOFTPHONE WEB .....	26
2.1.2.1	Plivo.....	26
2.1.2.2	Twillio.....	26
2.1.2.3	Flashphoner .....	26
2.1.3	CHAT DE SOPORTE OPEN SOURCE .....	27
2.1.4	LENGUAJES DE PROGRAMACIÓN.....	28
2.1.4.1	SERVIDOR.....	28
2.1.4.1.1	Frameworks .....	29
2.1.4.2	CLIENTE.....	29
<b>2.2</b>	<b>DISEÑO DE LA PLATAFORMA WEB</b> .....	<b>30</b>
2.2.1	MÓDULOS DE LA PLATAFORMA WEB.....	30
2.2.2	ADMINISTRACIÓN DE USUARIOS Y ROLES .....	30
2.2.2.1	Usuarios .....	30
2.2.2.2	Roles .....	31
2.2.2.2.1	Rol Cliente.....	31
2.2.2.2.2	Rol Administración .....	31
2.2.2.2.3	Rol Jefe Departamento .....	31
2.2.2.2.4	Rol Usuario .....	31
2.2.3	ADMINISTRACIÓN DE PROYECTOS .....	32
2.2.3.1	Gestión de Proyectos.....	32
2.2.3.2	Información Detallada de Proyecto .....	32
2.2.3.3	Creación de Etapas .....	32
2.2.3.4	Asignación de Usuarios a Proyectos.....	32
2.2.4	GESTIÓN DE DEPARTAMENTOS .....	33
2.2.5	GESTIÓN DE LLAMADAS.....	33
2.2.5.1	Click-to-call .....	33
2.2.5.2	Marcación a Departamento .....	34
2.2.5.3	Teclado de Marcación .....	34
2.2.5.4	Marcación entre operadores.....	34
2.2.6	CHAT DE SOPORTE.....	34
2.2.7	SERVICIOS WEB .....	35
<b>2.3</b>	<b>DIAGRAMA DE CASOS DE USO DE LA PLATAFORMA WEB</b> .....	<b>35</b>
2.3.1	Diagrama Casos de Uso – Administración de Usuarios y Roles.....	35

2.3.2	Descripción de Casos de Uso – Administración de Usuarios y Roles .....	35
2.3.3	Diagrama Casos de Uso – Administración de Proyectos .....	37
2.3.4	Descripción de Casos de Uso – Administración de Proyectos .....	38
2.3.5	Diagrama Casos de Uso – Gestión de Llamadas .....	40
2.3.6	Descripción de Casos de Uso – Gestión de Llamadas .....	41
2.3.7	Diagrama Casos de Uso – Chat de Soporte .....	42
2.3.8	Descripción de Casos de Uso – Chat de Soporte .....	43
2.3.9	Diagrama Casos de Uso – Servicios Web .....	44
2.3.10	Descripción de Casos de Uso – Servicios Web .....	44
<b>2.4</b>	<b>DIAGRAMA DE CLASES DE LA PLATAFORMA WEB .....</b>	<b>45</b>
<b>2.5</b>	<b>DISEÑO APLICACIÓN MÓVIL .....</b>	<b>47</b>
2.5.1	DIAGRAMA DE CASOS DE USO .....	47
2.5.2	DESCRIPCIÓN CASOS DE USO – APLICACIÓN MÓVIL .....	48
2.5.3	DIAGRAMA DE CLASES .....	49
2.5.4	DISEÑO DE INTERFACES .....	50
2.5.4.1	Interfaz de Acceso .....	50
2.5.4.2	Interfaz de Conversaciones .....	51
2.5.4.3	Interfaz Mensajes .....	51
<b>2.6</b>	<b>DISEÑO PARA LA GESTIÓN DE LLAMADAS Y REDIRECCIONAMIENTO .....</b>	<b>52</b>
2.6.1	PLANES DE MARCADO .....	52
2.6.2	REDIRECCIONAMIENTO .....	53
<b>2.7</b>	<b>INFRAESTRUCTURA PARA EL PROTOTIPO .....</b>	<b>54</b>
<b>CAPÍTULO 3</b>	<b>.....</b>	<b>56</b>
<b>3.</b>	<b>IMPLEMENTACIÓN Y PRUEBAS .....</b>	<b>56</b>
<b>3.1</b>	<b>IMPLEMENTACIÓN .....</b>	<b>56</b>
3.1.1	HARDWARE .....	56
3.1.2	VIRTUALIZACIÓN DE SERVIDORES .....	57
3.1.3	IMPLEMENTACIÓN SERVIDOR WEB .....	58
3.1.3.1	Instalación Sistema Operativo .....	59
3.1.3.2	Implementación Apache .....	60
3.1.3.2.1	Host Virtuales .....	60
3.1.3.3	Implementación PHP .....	61
3.1.3.3.1	Instalación PHP .....	61
3.1.3.3.2	Configuración PHP .....	62
3.1.3.4	Implementación MySQL .....	62
3.1.3.4.1	Instalación MySQL .....	62
3.1.3.4.2	Creación de Bases de Datos .....	63
3.1.4	IMPLEMENTACIÓN SERVIDOR MEDIOS MULTIMEDIA .....	63
3.1.4.1	Instalación Flashphoner Web Call Server .....	64
3.1.4.2	Activación Licencia Flashphoner Server .....	65
3.1.4.3	Ejecución Flashphoner Web Call Server .....	66
3.1.5	IMPLEMENTACIÓN SISTEMA WEB INTEGRAL .....	66
3.1.5.1	Implementación Symfony2 .....	67
3.1.5.1.1	Instalación Symfony2 .....	67
3.1.5.1.2	Instalación Bundles Adicionales .....	69
3.1.5.2	Creación de Entidades .....	70
3.1.5.3	Implementación de Base de Datos .....	72
3.1.5.4	Seguridad .....	76
3.1.5.4.1	Usuarios y Roles .....	76
3.1.5.4.2	Autenticación .....	77

3.1.5.4.3	Autorización .....	79
3.1.5.5	Enrutamiento .....	81
3.1.5.6	Controladores.....	82
3.1.5.7	Vistas .....	83
3.1.5.8	Implementación Chat en Tiempo Real mediante Websockets.....	83
3.1.5.8.1	Integración con Symfony .....	84
3.1.5.8.2	Integración con la Interfaz de Usuario .....	85
3.1.5.9	Implementaciónsoftphone web.....	86
3.1.5.9.1	Marcación por teclado.....	86
3.1.5.9.2	Marcación por usuario para operadores y administradores .....	89
3.1.5.9.3	Marcación por usuario para clientes .....	92
3.1.5.9.4	Marcación por Departamento .....	93
3.1.5.10	Implementación Chat de Soporte.....	94
3.1.5.10.1	Instalación Chat de Soporte Open Source.....	94
3.1.5.10.2	Integración Usuarios.....	95
3.1.5.10.3	Integración de Departamentos .....	97
3.1.5.10.4	Integración de Acceso .....	98
3.1.5.11	Implementación Servicios Web.....	99
3.1.5.11.1	Seguridad .....	100
3.1.6	IMPLEMENTACIÓN APLICACIÓN MÓVIL .....	101
3.1.6.1	Librerías Adicionales .....	102
3.1.6.1.1	AFNetworking.....	102
3.1.6.1.2	JSMessagesViewController .....	103
3.1.6.1.3	Parse.....	103
3.1.6.1.4	CocoaWSSE .....	103
3.1.6.2	Interfaz de Usuario .....	103
3.1.6.2.1	Acceso .....	104
3.1.6.2.2	Listado de Chats .....	105
3.1.6.2.3	Mensajes.....	106
3.1.6.3	Notificaciones Push .....	107
3.1.6.3.1	Envío de notificaciones push .....	108
3.1.6.3.2	Recepción notificaciones push .....	109
3.1.7	CONFIGURACIÓN CENTRAL TELEFÓNICA.....	110
3.1.7.1	Extensiones.....	110
3.1.7.2	Troncales.....	111
3.1.7.3	Planes de Marcado.....	111
3.1.7.3.1	Plan Webphone .....	112
3.1.7.3.2	Plan Nacional .....	112
3.1.7.3.3	Plan Celular.....	113
3.1.7.3.4	Plan Internacional .....	114
<b>3.2</b>	<b>MANUAL DEL USUARIO .....</b>	<b>114</b>
<b>3.3</b>	<b>PRUEBAS .....</b>	<b>115</b>
3.3.1	PRUEBAS FLASHPHONER WEBCALL SERVER.....	115
3.3.2	PRUEBAS SERVIDOR WEB.....	120
3.3.3	ANÁLISIS WEBSOCKETS .....	122
3.3.4	ANÁLISIS WEBRTC .....	124
3.3.5	ANÁLISIS SIP .....	126
<b>3.4</b>	<b>DIMENSIONAMIENTO DE RECURSOS Y SERVIDORES .....</b>	<b>128</b>
3.4.1	DIMENSIONAMIENTO SERVIDOR WEB .....	128
3.4.2	DIMENSIONAMIENTO SERVIDOR MULTIMEDIA .....	129
3.4.3	DIMENSIONAMIENTO CONEXIÓN A INTERNET .....	130
<b>3.5</b>	<b>REQUERIMIENTOS MÍNIMOS .....</b>	<b>132</b>
3.5.1	CENTRAL TELEFÓNICA SIP .....	132
3.5.2	SERVIDOR WEB .....	133
3.5.3	SERVIDOR DE MEDIOS MULTIMEDIA.....	133

3.5.4	CONEXIÓN A INTERNET .....	133
3.5.5	CONEXIÓN TELEFONÍA PÚBLICA .....	134
3.5.6	LICENCIA FLASHPHONER .....	134
<b>3.6</b>	<b>COSTOS DE LA IMPLEMENTACIÓN.....</b>	<b>135</b>
<b>CAPÍTULO 4</b>	<b>.....</b>	<b>136</b>
<b>4.</b>	<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>136</b>
4.1	CONCLUSIONES.....	136
4.2	RECOMENDACIONES .....	138
	<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>140</b>
	<b>ANEXOS .....</b>	<b>142</b>

## ÍNDICE DE FIGURAS

Figura 1.1 Establecimiento de sesión satisfactorio [1, p. 112].....	6
Figura 1.2 Ejemplo de terminación de sesión [1, p. 116].....	7
Figura 1.3 Sondeo vs Websockets [3, p. 7].....	9
Figura 1.4 Sondeo Largo vs Websockets [5].....	10
Figura 1.5 Comparativa entre sondeo y Websockets [6].....	10
Figura 1.6 Funcionamiento Pusher .....	14
Figura 1.7 Esquema básico MVC [14, p. 2].....	17
Figura 1.8 Symfony 2 y Doctrine .....	20
Figura 2.1 Casos de Uso – Administración de Usuarios y Roles .....	35
Figura 2.2 Casos de Uso – Administración de Proyectos .....	38
Figura 2.3 Casos de Uso – Gestión de Llamadas.....	41
Figura 2.4 Casos de Uso – Chat de Soporte.....	43
Figura 2.5 Casos de Uso – Servicios Web .....	44
Figura 2.6 Diagrama de clases - Sistema web.....	47
Figura 2.7 Diagrama Casos de Uso – Aplicación Móvil.....	48
Figura 2.8 Diagrama de Clases – Aplicación móvil .....	50
Figura 2.9 Interfaz de acceso .....	50
Figura 2.10 Interfaz de listado de conversaciones .....	51
Figura 2.11 Interfaz de mensajes .....	51
Figura 2.12 Infraestructura del prototipo .....	55
Figura 3.1 Resumen configuraciones servidor web.....	58
Figura 3.2 Selección ISO para instalación .....	59
Figura 3.3 Configuración host virtual support .lan .....	60
Figura 3.4 Configuración host virtual chat de soporte .....	61
Figura 3.5 Resumen configuraciones servidor medios multimedia .....	64
Figura 3.6 Activación licencia Flashphoner Server.....	66
Figura 3.7 Descarga de componentes Symfony.....	68
Figura 3.8 Archivo parameters.yml.....	68
Figura 3.9 Archivo composer.json.....	69
Figura 3.10 Creación entidad Departamento.....	71
Figura 3.11 Ejemplo entidad creada.....	74
Figura 3.12 Diagrama de tablas creadas por el ORM .....	75
Figura 3.13 Ejemplo repositorio.....	76
Figura 3.14 Archivo security.yml.....	77
Figura 3.15 Formulario acceso.....	78
Figura 3.16 Control de acceso .....	79
Figura 3.17 Archivo routing.yml.....	81
Figura 3.18 Enrutamiento con anotaciones.....	81
Figura 3.19 Código UserController.....	82
Figura 3.20 Configuración parámetros Pusher.....	84
Figura 3.21 Método pusherServerAction.....	85
Figura 3.22 Chat mediante Pusher para usuarios .....	86
Figura 3.23 Encabezado archivo teclado.html.twig .....	87
Figura 3.24 Archivo flashphoner.xml.....	88
Figura 3.25 Sobreescritura método connect .....	88

Figura 3.26 Función para guardar las llamadas desde Javascript .....	89
Figura 3.27 Interfaz marcación por teclado .....	89
Figura 3.28 Etiquetas botones para llamadas .....	90
Figura 3.29 Método llamadaEntrante .....	91
Figura 3.30 Eventos clic para botones llamar, colgar y contestar .....	92
Figura 3.31 Archivo marcacionContactos.html.twig .....	93
Figura 3.32 Interfaz Click-to-Call para clientes .....	93
Figura 3.33 Marcación por departamento .....	94
Figura 3.34 Diagrama de Tablas Chat de Soporte .....	95
Figura 3.35 Creación usuario Webmin desde sistema web .....	96
Figura 3.36 Edición de Usuarios Webmin .....	97
Figura 3.37 Actualización departamentos en Webmin .....	98
Figura 3.38 Menú para Acceso al Chat de Soporte .....	99
Figura 3.39 Rutas para funciones REST .....	100
Figura 3.40 Captura Wireshark solicitud mediante WSSE .....	101
Figura 3.41 Configuración WSSE en security.yml .....	101
Figura 3.42 Diagrama de Interacción Aplicación Móvil .....	104
Figura 3.43 Código para autenticación de usuarios .....	105
Figura 3.44 Código carga de chats en el tableView .....	106
Figura 3.45 Interfaz de Mensajes .....	107
Figura 3.46 Envío Notificación push nuevo chat .....	108
Figura 3.47 Envío notificaciones push nuevo mensaje .....	109
Figura 3.48 Recepción notificación push 1 .....	109
Figura 3.49 Recepción notificación push 2 .....	110
Figura 3.50 Interfaz Creación Extensiones en la Central Telefónica .....	110
Figura 3.51 Configuración Troncal .....	111
Figura 3.52 Plan de Marcado Webphone .....	112
Figura 3.53 Plan de Marcado Nacional .....	112
Figura 3.54 Plan de Marcado Celular .....	113
Figura 3.55 Plan de Marcado Internacional .....	114
Figura 3.56 Captura 10 llamadas central telefónica .....	116
Figura 3.57 Consumo CPU 10 llamadas .....	116
Figura 3.58 Consumo Memoria 10 llamadas .....	117
Figura 3.59 Captura 20 llamadas central telefónica .....	117
Figura 3.60 Consumo CPU 20 llamadas .....	118
Figura 3.61 Consumo memoria 20 llamadas .....	118
Figura 3.62 Captura 30 llamadas central telefónica .....	119
Figura 3.63 Consumo CPU 30 llamadas .....	119
Figura 3.64 Consumo memoria 30 llamadas .....	120
Figura 3.65 Configuración simulación de tráfico .....	121
Figura 3.66 Estadísticas de prueba de carga .....	121
Figura 3.67 Gráfica de resultados prueba de carga .....	122
Figura 3.68 Captura Google Chrome Websockets .....	123
Figura 3.69 Captura Wireshark Websockets .....	123
Figura 3.70 Modelo comunicación WebRTC [22] .....	124
Figura 3.71 Captura Wireshark protocolo ICE/STUN .....	125
Figura 3.72 Captura Wireshark protocolo DTLS .....	126
Figura 3.73 Flujo de tramas RTP .....	126
Figura 3.74 Captura Wireshark flujo SIP .....	128



Figura 3.75 Gráfica tasa de transferencia 10 llamadas .....	131
Figura 3.76 Gráfica tasa de transferencia 20 llamadas .....	131

## ÍNDICE DE TABLAS

Tabla 1.1 Métodos SIP [1, p. 103] .....	2
Tabla 1.2 Tipos de códigos de respuesta SIP [1, p. 103] .....	3
Tabla 1.3 Lenguajes de programación del lado del servidor con mayor popularidad [13] .....	15
Tabla 2.1 Características básicas central telefónica .....	24
Tabla 2.2 Especificaciones principales central telefónica .....	24
Tabla 2.3 Ranking 10 lenguajes de programación [21] .....	28
Tabla 2.4 Descripción Caso de Uso – Crear Usuarios .....	36
Tabla 2.5 Descripción Caso de Uso – Listar Usuarios .....	36
Tabla 2.6 Descripción Caso de Uso – Modificar Usuarios .....	36
Tabla 2.7 Descripción Caso de Uso – Eliminar Usuarios .....	37
Tabla 2.8 Descripción Caso de Uso – Crear un proyecto .....	38
Tabla 2.9 Descripción Caso de Uso – Listar Proyectos .....	39
Tabla 2.10 Descripción Caso de Uso – Modificar proyecto .....	39
Tabla 2.11 Descripción Caso de Uso – Eliminar proyecto .....	40
Tabla 2.12 Descripción Caso de Uso – Definir Prioridad de Empleados en Etapas .....	40
Tabla 2.13 Descripción Caso de Uso – Llamar a Empleados .....	41
Tabla 2.14 Descripción Caso de Uso – Llamar a Departamento .....	42
Tabla 2.15 Descripción Caso de Uso – Marcación mediante teclado .....	42
Tabla 2.16 Descripción Caso de Uso – Enviar Mensaje .....	43
Tabla 2.17 Descripción Caso de Uso – Abrir consola de Agente .....	44
Tabla 2.18 Descripción Caso de Uso – Autenticación .....	44
Tabla 2.19 Descripción Caso de Uso – Solicitar Listado de Conversaciones .....	45
Tabla 2.20 Descripción Caso de Uso – Enviar Mensajes .....	45
Tabla 2.21 Descripción Caso de Uso – Autenticación .....	48
Tabla 2.22 Descripción Caso de Uso – Ver Mensajes .....	49
Tabla 2.23 Descripción Caso de Uso – Enviar Mensaje .....	49
Tabla 2.24 Permisos según planes de marcado .....	53
Tabla 2.25 Planes de Marcado según roles .....	53
Tabla 3.1 Especificaciones servidor x3100 M4 .....	56
Tabla 3.2 Características servidor web .....	58
Tabla 3.3 Configuraciones php.ini .....	62
Tabla 3.4 Características servidor medios multimedia .....	63
Tabla 3.5 Componentes principales Symfony2 .....	67
Tabla 3.6 Anotaciones ORM para entidades .....	72
Tabla 3.7 Descripción archivos para implementación softphone web .....	87
Tabla 3.8 Funciones REST implementadas .....	99
Tabla 3.9 Recursos para desarrollo aplicación móvil .....	102
Tabla 3.10 Resumen pruebas Web Call Server .....	120
Tabla 3.11 Campos paquete WebSocket .....	124
Tabla 3.12 Llamadas concurrentes empresa tipo .....	129
Tabla 3.13 Licencias básicas Flashphoner .....	134
Tabla 3.14 Resumen de costos .....	135

## RESUMEN

El presente proyecto consiste en la implementación de un software basado en la web que ayude a una empresa tipo integradora de servicios a brindar soporte mediante voz y chat a sus clientes. Para el desarrollo del sistema integral se tomó en cuenta tecnologías usadas en el desarrollo web como Websockets y WebRTC de HTML5, la interoperabilidad con la infraestructura de telefonía IP de una empresa tipo y la integración con aplicaciones de terceros para mejorar las funcionalidades del sistema.

En el primer capítulo se realiza una descripción de los conceptos de telefonía IP como SIP y softphones, características de HTML5 como Websockets y WebRTC además de conceptos relacionados con desarrollo web, las metodologías aplicadas y los servicios web.

En el segundo capítulo se hace un análisis de los requerimientos necesarios para la implementación del sistema integral, se realiza una selección de las herramientas utilizadas en el desarrollo de la aplicación y se estructura el diseño básico tanto de la aplicación web como de la aplicación móvil.

En el tercer capítulo se realiza la implementación de toda la aplicación integral empezando por la instalación del ambiente de desarrollo mediante virtualización de servidores, la instalación del framework de desarrollo, el desarrollo de los diferentes módulos, la integración con las aplicaciones de terceros, el desarrollo de la aplicación para móviles y la implementación de los servicios web.

En el cuarto capítulo se incluyen todas las conclusiones obtenidas como las recomendaciones que se realizan según la experiencia adquirida en la implementación del presente proyecto.

## PRESENTACIÓN

El continuo avance en las tecnologías web ha permitido que cada día aumenten las funcionalidades ofrecidas por las aplicaciones basadas en ambientes web y sean comparables a las aplicaciones de escritorio, facilitando el acceso a los usuarios sin necesidad de instalar un programa adicional al navegador.

De esta manera se ha implementado una aplicación web que permita comunicar a los usuarios y clientes de una empresa tipo mediante la telefonía IP, de forma que los clientes puedan realizar llamadas a los empleados de la empresa utilizando simplemente la interfaz web de la aplicación, reduciendo las comunicaciones mediante la telefonía pública de tal forma que se ahorren costos al cliente y a la empresa en la adquisición de nuevas líneas analógicas o troncales SIP. Además mejora la eficiencia en la comunicación, puesto que en caso de que el usuario no se encuentre conectado en su extensión las llamadas son automáticamente redireccionadas a sus respectivos teléfonos móviles.

El sistema también cuenta con la integración de un chat de soporte que ayuda a los clientes a comunicarse de manera rápida con los empleados de la empresa mediante mensajes, para mejorar la respuesta de estos mensajes se ha creado una aplicación para dispositivos móviles que facilite la interacción del empleado con los clientes en cualquier momento.

La administración de proyectos incluida en la aplicación permite que los empleados de la empresa sean asignados a determinado proyecto con una prioridad independiente en cada etapa del mismo, de forma que el soporte hacia los clientes sea canalizado directamente por el personal adecuado.

# CAPÍTULO 1

## 1. MARCO TEÓRICO

### 1.1 INTRODUCCIÓN

Todos los avances en el desarrollo de aplicaciones se han orientado a facilitar y aumentar el acceso a las mismas, por esta razón se percibe un crecimiento en el desarrollo de aplicaciones orientadas a la web y la migración de aplicaciones utilizadas en ambientes de escritorio, que en la actualidad tienen su versión web o migraron toda su infraestructura. En las aplicaciones web se han incorporado funcionalidades que se tienen en entornos de escritorio tales como: reproducción de audio y video, creación y modificación de archivos, comunicación entre usuarios, entre otras, cada día aumentan los requerimientos por parte de los usuarios en busca de nuevas alternativas para facilitar el uso cotidiano de este tipo de herramientas.

En el presente capítulo se detallarán los principales conceptos asociados con el desarrollo del proyecto, entre los cuales se destacan temas como: centrales telefónicas, desarrollo de software, arquitectura de desarrollo, servicios web y aplicaciones móviles.

El sistema web desarrollado en el presente proyecto tiene por objetivo facilitar la comunicación entre el cliente y el personal de soporte de la empresa, a través de la integración de telefonía IP, teléfonos inteligentes, mensajería instantánea y servicios web, que permitan realizar una comunicación eficiente y de fácil acceso.

Con la incorporación de un softphone web o webphone en el sistema que establezca una fácil comunicación desde el browser del cliente y la telefonía IP de la empresa o la telefonía celular por medio de redirecciones previamente configuradas en la central telefónica, en caso de que el técnico o agente de ventas

no se encuentre disponible en su extensión.

Mediante la inclusión del módulo de proyectos en el sistema se tendrá un registro de los proyectos que pertenecen o fueron asignados en las distintas etapas a cada usuario, logrando un enlace directo entre los clientes, sus proyectos y los vendedores o técnicos asignados a cada uno de ellos.

Se acoplará con una aplicación de mensajería instantánea para soporte de código abierto que permita interactuar de una manera alternativa a los clientes con el personal asignado al proyecto en caso de que el técnico o agente de ventas no esté disponible o su extensión se encuentre ocupada. Se realizará una integración con móviles mediante la creación de una aplicación para *smartphones* orientada hacia el usuario de soporte que cree un canal de comunicación entre el usuario y el cliente por medio de mensajes y dar soluciones de una manera rápida, en cualquier momento y desde cualquier lugar.

## **1.2 SIP (SESSION INITIAL PROTOCOL)[1]**

### **1.2.1 INTRODUCCIÓN**

Es un protocolo desarrollado por la IETF<sup>1</sup> MMUSIC<sup>2</sup> basado en elementos de HTTP, como su nombre lo indica su principal función es la del inicio de sesión pero también tiene otras funciones como la notificación de presencia y los mensajes cortos.

### **1.2.2 MÉTODOS**

Las peticiones realizadas por SIP conocidos como métodos se muestran en la Tabla 1.1 que se muestra a continuación.

**Tabla 1.1** Métodos SIP [1, p. 103]

---

<sup>1</sup>**IETF** Internet Engineering Task Force

<sup>2</sup>**MMUSIC** Multiparty Multimedia Session Control

MÉTODO	DESCRIPCIÓN (INGLÉS)	DESCRIPCIÓN (ESPAÑOL)
INVITE	Session setup	Configuración de Sesión
ACK	Acknowledgment of final response to INVITE	Acuse de recibo para la respuesta final a INVITE
BYE	Session termination	Terminación de Sesión
CANCEL	Pending session cancellation	Cancelación de sesión pendiente
REGISTER	Registration of a user's URI	Registro de un URI de usuario
OPTIONS	Query of options and capabilities	Consulta de opciones y capacidades
INFO	Mid-call signaling transport	Transporte de señalización de intermedio de llamada
PRACK	Provisional response acknowledgment	Acuse de recibo de respuesta provisional
UPDATE	Update session information	Actualización de información de sesión
REFER	Transfer user to a URI	Transferencia de usuario a un URI
SUSCRIBE	Request notification of an event	Notificación de solicitud de un evento
NOTIFY	Transport of subscribed event notification	Transporte de notificación de evento suscrito
MESSAGE	Transport of an instant message body	Transporte del cuerpo de un mensaje instantáneo
PUBLISH	Upload presence state to a server	Carga de estado de presencia a un servidor

### 1.2.3 RESPUESTAS

Las respuestas que se obtienen en SIP son numéricas, muchas de las cuales fueron tomadas de HTTP, como otras fueron creadas, las cuales son divididas en 6 clases identificadas por el primer dígito del código. En la Tabla 1.2 se muestran las diferentes clases de respuestas.

**Tabla 1.2** Tipos de códigos de respuesta SIP [1, p. 103]

CLASE	DESCRIPCIÓN (INGLES)	DESCRIPCIÓN (ESPAÑOL)
1xx	Provisional or Informational - Request is progressing but not yet complete	Provisional o informativo - La solicitud está progresando pero aún no está completa
2xx	Success - Request has been completed successfully	Éxito - La solicitud ha sido completada satisfactoriamente
3xx	Redirection - Request should be tried at another location	Redirección - La solicitud debe ser probada en otro lugar
4xx	Client Error - Request was not completed because of an error in the request, can be retried when corrected	Error del Cliente - La solicitud no fue completada por causa de un error en la solicitud, puede ser reintentado cuando se corrija
5xx	Server Error - Request was not completed because of an error in the recipient, can be retried at another location	Error del Servidor - La solicitud no fue completa a causa de un error en el recipiente, puede ser reintentado a otro lugar
6xx	Global Failure - Request has failed and should not be retried again	Falla Global - La solicitud ha fallado y no se debería reintentar otra vez

## 1.2.4 ELEMENTOS DE UNA RED SIP

### 1.2.4.1 AGENTES DE USUARIO

Son los dispositivos finales en una red SIP, estos pueden ser un teléfono SIP o un softphone SIP, como de igual forma puede ser un Gateway hacia otra red.

Un agente de usuario está compuesto por un Agente de Usuario Cliente (UAC) que inicia las peticiones y un Agente de Usuario Servidor (UAS) que se encarga de recibir las peticiones y emitir las respuestas.



### 1.2.4.2 SEVIDORES

Son intermediarios que ayudan a los agentes de usuario a establecer sesiones. Existen 3 tipos de servidores:

- **Proxy SIP:** Recibe peticiones SIP de un agente de usuario y envía las peticiones a otra ubicación.
- **Servidor de Redirección:** Recibe las peticiones de un agente de usuario y retorna una respuesta de clase redirección (3xx).
- **Servidor de Registro:** Recibe peticiones de registro y actualiza la información del usuario en el servicio de localización u otra base de datos.

Los servidores SIP son opcionales en una comunicación puesto que se podría realizar una comunicación punto a punto entre agentes de usuario.

### 1.2.4.3 SERVICIOS DE LOCALIZACIÓN

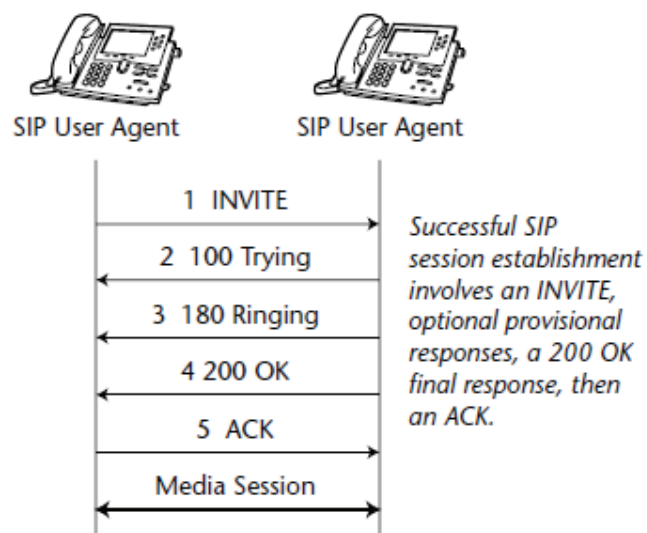
Hace referencia a la base de datos que contiene información de los usuarios, como también puede contener información de enrutamiento de la red SIP. Los agentes de usuario no se comunican directamente con esta base de datos, lo hacen mediante un servidor de los antes mencionados.

### 1.2.5 FUNCIONES DE SIP

Entre las principales funciones de SIP tenemos: resolución de direcciones, relacionadas con la sesión, señalización intermedia de llamada, control de llamada, configuración de calidad de servicio de llamadas y no relacionadas a la sesión. Al ser tan extensas cada una de estas funciones nos enfocaremos en las principales relacionadas a la sesión concerniente al presente proyecto.

### 1.2.5.1 CONFIGURACIÓN DE SESIÓN

SIP utiliza una petición INVITE para configurar la sesión, la cual contiene la descripción del tipo de sesión que el agente de usuario quiere establecer. La configuración de sesión de SIP se la realiza mediante el intercambio de 3 mensajes que pueden ser INVITE/200/ACK como satisfactorio o INVITE/4xx-5xx-6xx/ACK como fallida. En la Figura 1.1 se muestra la interacción de mensajes en un establecimiento de sesión satisfactorio.



**Figura 1.1** Establecimiento de sesión satisfactorio [1, p. 112]

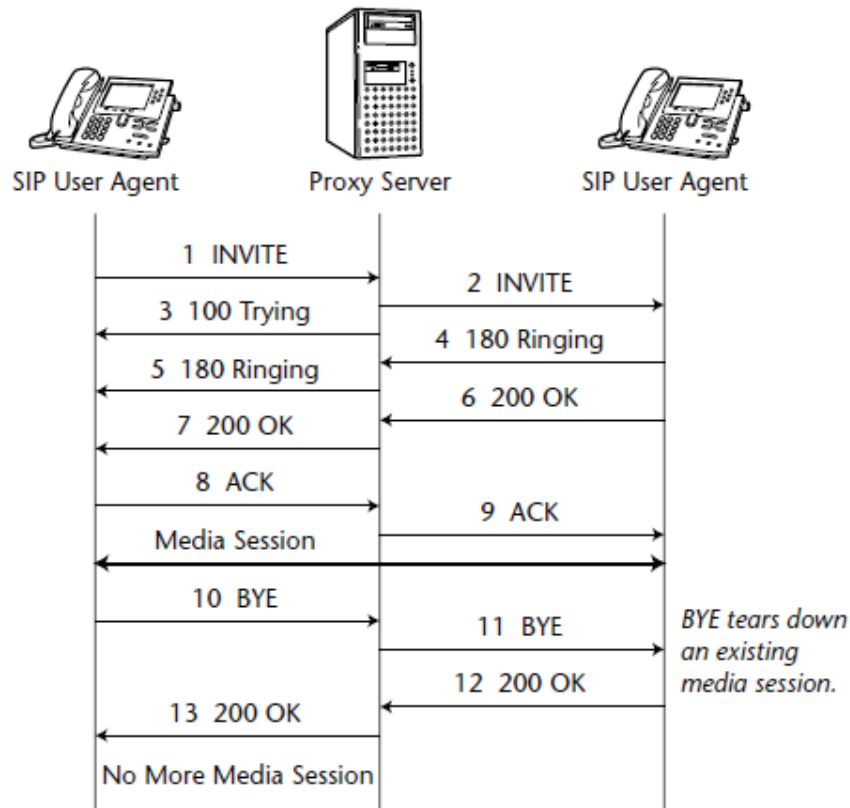
### 1.2.5.2 NEGOCIACIÓN DE MEDIOS

SIP por sí solo no provee negociación de medios, pero esto se lo realiza con la ayuda de Session Description Protocol (SDP), en el cual un agente de usuario propone uno o más tipos de medios y el otro acepta o niega en respuesta. En el cuerpo de SDP el agente de usuario indica el tipo de medios, codec, dirección IP y número de puerto para la transmisión de medios.

### 1.2.5.3 TERMINACIÓN Y CANCELACIÓN DE SESIÓN

Terminación de sesión es cuando un agente de usuario envía una petición BYE referenciada a una llamada existente, en la Figura 1.2 se muestra un ejemplo de

este proceso. La cancelación de sesión se presenta cuando una gente de usuario ha enviado una petición INVITE pero no ha recibido respuesta, entonces envía una petición CANCEL.



**Figura 1.2** Ejemplo de terminación de sesión [1, p. 116]

### 1.3 WEBSOCKETS[2][3]

“The WebSocket specification—developed as part of the HTML5 initiative—introduced the WebSocket JavaScript interface, which defines a full-duplex single socket connection over which messages can be sent between client and server.”[4]

Con la incorporación del soporte HTML5 en la mayoría de navegadores, los desarrolladores adoptan sus nuevas API<sup>3</sup> para crear aplicaciones que aprovechen los nuevos beneficios, como es el caso de Websockets una de las más importantes

<sup>3</sup>API Application Programming Interface

novedades de HTML5.

La especificación Websocket define una comunicación *full-duplex* bidireccional entre el cliente y el servidor, es decir permite que el intercambio de mensajes sea iniciado por el cliente o el servidor. Esta nueva especificación resuelve los inconvenientes mediante una gran reducción en tráfico innecesario y latencia en comparación con métodos anteriores como el sondeo y el sondeo largo.

Para entregar una mayor seguridad en las comunicaciones mediante Websockets se amplió el soporte para que también se permita la implementación de Websockets sobre SSL. El protocolo de Websocket fue diseñado para trabajar en total compatibilidad con la infraestructura Web existente, por este motivo para empezar una comunicación mediante Websockets se realiza mediante una conexión HTTP.

Los navegadores que tienen incorporado Websockets son los siguientes: Chrome, Firefox, Safari, IE, etc.

### **1.3.1 COMPARATIVA ENTRE WEBSOCKETS Y MÉTODOS ANTERIORES**

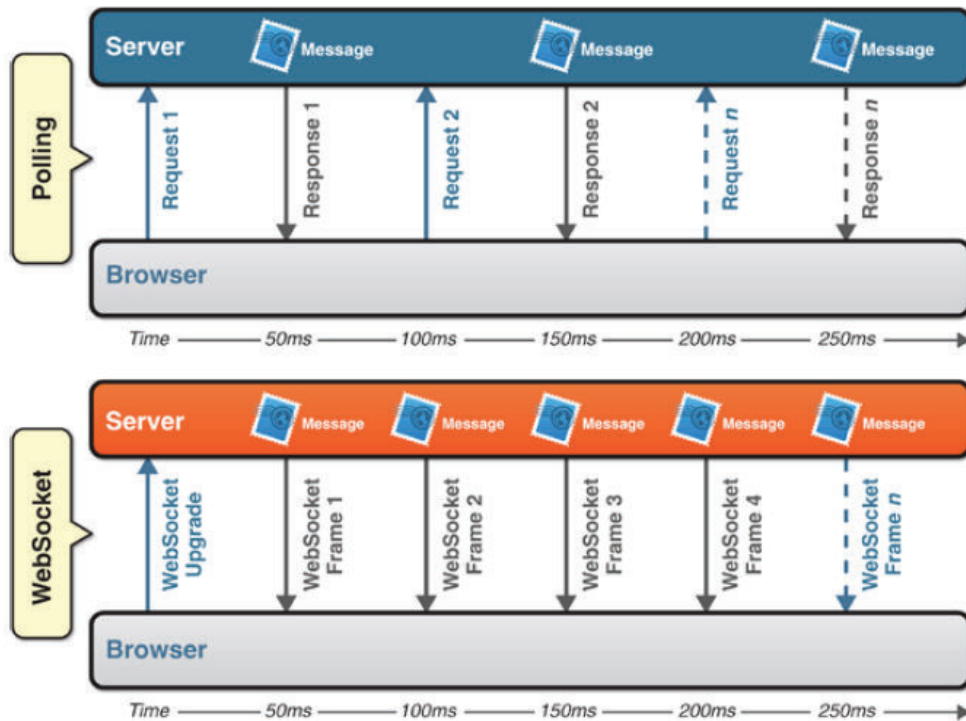
Entre las aplicaciones más frecuentes que utilizan Websockets, son aquellas en las que el servidor necesita notificar al cliente en tiempo real la nueva información o el cambio de estado de algunas variables. Antes de la llegada de HTML5 se crearon diferentes métodos que permitieron realizar este tipo de comunicaciones entre el servidor y el cliente, las principales son el sondeo y el sondeo largo.

El sondeo es realizado mediante una técnica de comunicación asincrónica en entornos web denominada AJAX <sup>4</sup>, mediante la cual se puede realizar comunicaciones con el servidor en segundo plano. Como se visualiza en la Figura 1.3 en la parte superior indicada como *polling*, para la implementación del método de sondeo se generan peticiones recurrentes al servidor en búsqueda de nueva información, que al compararlo con Websockets en la parte inferior de la Figura 1.3,

---

<sup>4</sup>**AJAX** Asynchronous JavaScript And XML (JavaScript asíncrono y XML)

se puede evidenciar el desperdicio de recursos que involucra el realizar peticiones recurrentes al servidor sin saber si el mismo tiene o no nueva información.



**Figura 1.3** Sondeo vs Websockets [3, p. 7]

El sondeo largo es un método que fue denominado Comet, en la Figura 1.4 se muestra la comparativa entre sondeo largo en inglés *long polling* y Websockets, el primero consiste en mantener la petición hacia el servidor abierta hasta que el servidor tenga información nueva que enviar. Aunque este método solucionó el problema de las conexiones recurrentes persiste un problema de eficiencia puesto que como se indica en la Figura 1.4 tiene que realizar una petición y esperar una respuesta HTTP, lo cual genera una sobrecarga de 871 bytes comparado con Websockets que apenas añade 2 bytes de sobrecarga.

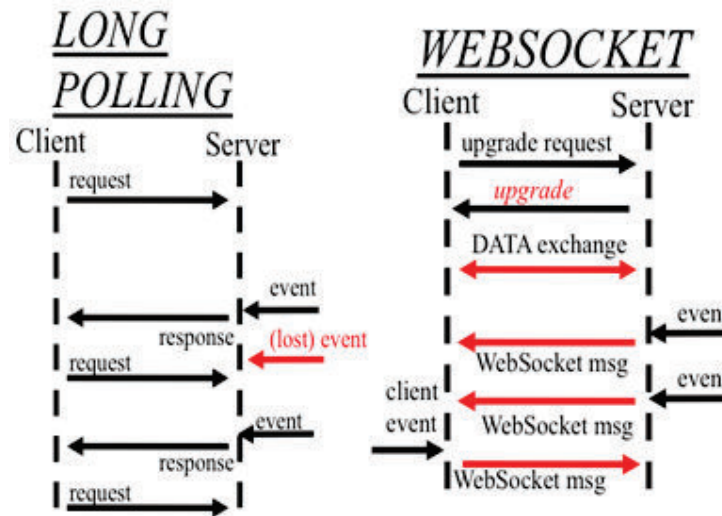


Figura 1.4 Sondeo Largo vs Websockets [5]

En la Figura 1.5 se puede apreciar una comparativa evaluada en 3 casos, con 1000, 10000 y 100000 clientes respectivamente, todos ellos tomando en consideración que el sondeo se lo hace cada segundo y que de igual forma se recibe un mensaje por segundo en Websockets.

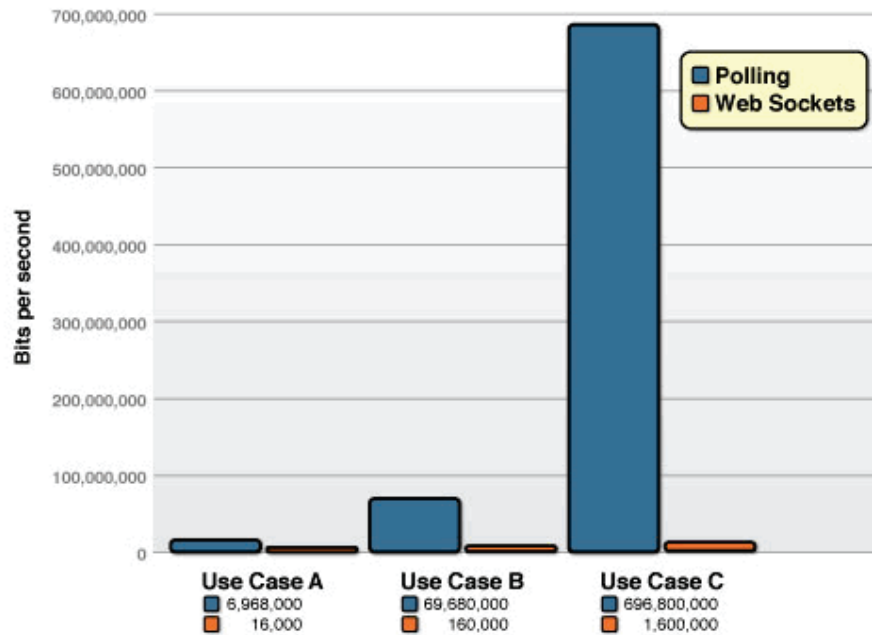


Figura 1.5 Comparativa entre sondeo y Websockets [6]

## **1.4 WEBRTC[7]**

“WebRTC is a free, open project that enables web browsers with Real-Time Communications (RTC) capabilities via simple JavaScript APIs. The WebRTC components have been optimized to best serve this purpose.”[8]

WebRTC (Web Real-Time Communicatios) cuya traducción de sus siglas en inglés pertenece a Comunicación en Tiempo Real Web y es utilizado para la transmisión de audio, video u otros datos entre navegadores Web sin instalar complementos o programas de terceros. Actualmente se encuentran en desarrollo MediaStream API, PeerConnection API y DataChannel API.

### **1.4.1 MEDIASTREAM API**

Esta API se basa en el manejo de un objeto MediaStream que representa un flujo de datos de audio y video. Un objeto MediaStream local es creado con los datos de audio y video del micrófono y la cámara respectivamente. Cuando se trata de un MediaStream no local, se hace referencia a un flujo de datos que fue originado en la red y fue obtenido mediante el PeerConnection API.

### **1.4.2 PEERCONNECTION API**

Representada principalmente por el objeto RTCPeerConnection que ayuda al intercambio de objetos MediaStream entre los navegadores.

### **1.4.3 DATACHANNEL API**

Permite el soporte del uso de WebRTC para otro tipo de datos diferentes a audio y video como son Juegos, escritorio remoto, transferencia de archivos entre otros.

## 1.5 SOFTPHONE

“A softphone (software telephone) is an application program that enables voice over Internet Protocol (VoIP) telephone calls from computing devices. In the enterprise, softphones are sometimes referred to as soft clients.”[9]

Permite utilizar las mismas funcionalidades de un teléfono SIP sin necesidad de tener un hardware adicional. Los softphones en la actualidad han sido implementados para una gran cantidad de dispositivos como son: computadoras, teléfonos inteligentes, tabletas entre otros dispositivos, lo que ha permitido que la integración de la telefonía IP implementada en una empresa aumente continuamente.

En la actualidad existen softphones que implementan los estándares SIP y/o H.323<sup>5</sup> y los que manejan estándares propios como son el caso de Skype, Hangouts, Line entre otros.

Las funcionalidades y características que se pueden obtener de un softphone son variados puesto que existen empresas que han desarrollado softphones que incluso permiten la inclusión de video en las llamadas.

Los códecs de audio que generalmente se utilizan en la mayoría de softphones son G.711<sup>6</sup> y G.729<sup>7</sup>.

Los softphones convencionalmente son desarrollados para ambientes de escritorio, puesto que la carga de procesamiento y la complejidad de la integración con la telefonía SIP, ha hecho que los softphones orientados a entornos web sean muy pocos y de costos muy elevados.

---

<sup>5</sup>H.323 creador por la ITU-T (International Telecommunication Union), define los protocolos para proveer sesiones de comunicación audiovisual sobre paquetes de red.

<sup>6</sup>G.711 es un estándar de la ITU-T para la codificación de audio.

<sup>7</sup>G.729 es un algoritmo de compresión de datos de audio para voz



### 1.5.1 SOFTPHONE WEB O WEBPHONE

Es un softphone basado en web que tiene similares características a un softphone convencional pero presta funcionalidad exclusiva en la web logrando que los usuarios en la mayoría de casos no necesiten instalar un software adicional a su navegador para realizar o recibir una llamada e integrarse con la telefonía IP de la empresa o con algún proveedor de servicios de telefonía IP mediante SIP.

Una de las principales funcionalidades que se ha adoptado actualmente en la web es añadir botones de *click-to-call* que permiten a los usuarios mediante la utilización de un webphone realizar una llamada instantánea desde el navegador hacia una extensión de la empresa directamente, entregando al usuario la mayor facilidad para comunicarse con cualquier representante de la empresa.

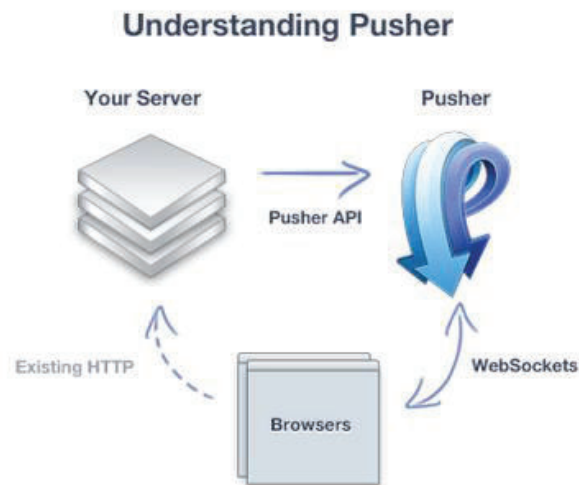
Entre las implementaciones más comunes que existen son las basadas en Flash, Java Applets, Websockets e Híbridas. Siendo de estas las más estables las versiones basadas en Flash por motivo de su soporte para medios multimedia pero la necesidad de complementos y la aparición de HTML5 ha causado que vayan desapareciendo de los entornos Web.

Las versiones basadas en Java Applets han quedado obsoletas por su lentitud y carga. Las versiones basadas completamente en Websockets y WebRTC son las implementaciones más actuales, pero hasta el momento no existe una versión de producción, todas se encuentran en desarrollo por motivo de que HTML5 que incluyó el soporte de Websockets y WebRTC tiene cambios continuos y no está completamente implementado en todos los navegadores.

Las versiones híbridas que integran Flash, Websockets y WebRTC utilizan las nuevas características de HTML5, pero ayudan a mantener una mayor compatibilidad con aquellos navegadores que no soportan Websockets, por medio de la utilización de las funcionalidades de Flash.

## 1.6 PUSHER[10]

Pusher es un servicio que facilita y agiliza la implementación de funcionalidades bidireccionales en tiempo real usando Websockets. En caso de navegadores que no soportan Websockets utiliza mecanismos de respaldo mediante Flash o HTTP.



**Figura 1.6** Funcionamiento Pusher

Utiliza un modelo publicador/suscriptor basado en canales que permite controlar que cliente recibe los mensajes. Provee también mecanismos de autenticación para canales privados y funcionalidades de presencia para conocer que clientes se encuentran en línea. En la Figura 1.6 se muestra el funcionamiento básico de Pusher.

## 1.7 PARSE[11]

Parse es un servicio que provee almacenamiento en la nube para las aplicaciones mediante una base de datos que se puede acceder desde las APIs creadas para diferentes lenguajes y plataformas, entre estas tenemos: PHP, Javascript, Android, iOS y Windows Phone, también permite el acceso mediante una API REST.

La gestión de notificaciones *push* es otra funcionalidad que también incorpora este servicio, mediante esta funcionalidad permite crear, programar y segmentar el envío de las notificaciones asociadas a las aplicaciones creadas para las plataformas iOS, Android y Windows Phone.

## 1.8 DESARROLLO WEB

### 1.8.1 PHP[12]

“PHP is a popular general-purpose scripting language that is especially suited to web development. Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.”[12]

PHP acrónimo recursivo de PHP: Hypertext Preprocessor, es el lenguaje más utilizado en el lado del servidor para desarrollo Web, alcanzando el 82% en popularidad con respecto a otros lenguajes como se muestra en la Tabla 1.3. Su gran popularidad fue ganada por su facilidad de uso, haciendo de este un lenguaje de programación muy comprensible aplicado por programadores y no programadores para sus páginas web.

**Tabla 1.3** Lenguajes de programación del lado del servidor con mayor popularidad [13]

POSICIÓN	LENGUAJE	USO
1	PHP	82,0%
2	ASP.NET	17,0%
3	JAVA	2,8%
4	COLDFUSION	0,7%
5	RUBY	0,6%

Es un lenguaje interpretable puesto que lo que necesita para su ejecución es un intérprete PHP soportado en el servidor web, el cual se encarga de procesar el script PHP y devolver información hacia el cliente. PHP también permite la ejecución de órdenes o scripts en líneas de comandos como Perl o Python y aunque

su orientación original es hacia la web se han creado implementaciones que permiten ejecutarlo en entornos de escritorio, aunque aún no son muy populares.

Su sintaxis se asemeja mucho a lenguajes como C y Perl esto facilita aún más a los programadores a reducir el tiempo y costo en aprender un nuevo lenguaje de programación. Tiene soporte para una gran cantidad de bases de datos tanto relacionales como no relacionales entre las que podemos mencionar algunas como: MySQL, Microsoft SQL Server, Oracle, MongoDB, PostgreSQL, entre otras.

En sus inicios siendo un lenguaje muy abierto y fácil de usar provocó una gran cantidad de malas prácticas y generación de contenido web que era muy difícil de mantener. En consecuencia se crearon frameworks de desarrollo que permitieron a los programadores evitar en gran medida las malas prácticas, crear código con un mantenimiento más eficiente y con menos complicaciones.

En la actualidad con la incorporación de nuevas versiones y frameworks, PHP ha logrado superar sus inconvenientes y ser el lenguaje de programación con más adeptos en el desarrollo de aplicaciones web.

### **1.8.2 FRAMEWORK DE DESARROLLO WEB**

Es la integración de un conjunto de prácticas, patrones y módulos que proveen la base y la metodología de trabajo para el desarrollo de una aplicación web, con el objetivo de organizar, estandarizar y reutilizar el código.

Los frameworks de desarrollo generan una estructura base para la implementación de una aplicación web. De esta manera el programador utiliza la mayor cantidad de tiempo en las partes más relevantes de la aplicación y utiliza menos tiempo en actividades rutinarias.

Un framework puede ser evaluado en la agilidad que represente al programador para efectuar la mayor cantidad de actividades en el menor tiempo posible.

### 1.8.2.1 Framework MVC[14]

Es un patrón de diseño de software creado en base a la interconexión de 3 componentes Modelo, Vista y Controlador con el fin de optimizar el desarrollo de aplicaciones web y alcanzar una escalabilidad y mantenimiento eficiente. En la Figura 1.7 se puede observar la interacción entre los componentes del patrón de diseño.

Las funciones correspondientes a los 3 componentes son las siguientes:

**Modelo:** Representa toda la lógica del negocio mediante la implementación de la persistencia y abstracción de la información.

**Vista:** Encargada de mostrar la información e interactuar con el usuario.

**Controlador:** Conecta todos los eventos y requerimientos entre la vista y el modelo. Separa la lógica del negocio de un modelo de la interfaz de la vista.

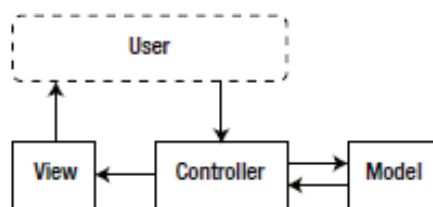


Figura 1.7 Esquema básico MVC [14, p. 2]

### 1.8.2.2 Symfony2[15]

Es considerado por parte de la comunidad un Framework MVC para PHP, pero como su creador en su página oficial lo afirma, es un Framework HTTP[16], es decir, orientado explícitamente a la Web y diseñado para este tipo de comunicación. Es importante considerar que el modelo de desarrollo MVC es utilizado tanto en programación de aplicaciones para teléfonos inteligentes y para aplicaciones web,

por este motivo Symfony al ofrecer una solución completamente enfocada en la Web permite optimizar el código y reducir los conocidos inconvenientes que conlleva el desarrollo de aplicaciones web.

Integra las mejores características probadas con otros lenguajes o frameworks, entre estas podemos hacer referencia a las principales:

**Twig:** Es un motor de plantillas creado para PHP, pero basa su sintaxis en Jinga y Django Templates que fueron creados para facilitar la sintaxis en las vistas.

**Yaml:** Formato para serialización de información que es utilizado por Symfony 2 para facilitar la modificación de configuraciones.

**Console:** Librería integrada en Symfony 2 para interactuar con la línea de comandos y poder realizar creaciones de entidades, base de datos, tareas programadas entre otras.

Symfony tiene una gran flexibilidad en el código puesto que para sus archivos de configuración permite usar Yaml, XML o PHP y para sus archivos de plantillas igual deja abierta la elección según las experticias del usuario en utilizar PHP o Twig.

Para el almacenamiento de información tiene la capacidad para utilizar tanto bases de datos relacionales como no relacionales.

Su comunidad tiene más de 200 usuarios que realizan colaboraciones con el proyecto, una cifra importante puesto que en comparación con proyectos de similar importancia como Zend Framework que posee solo 43, representa un valor alto y permite evidenciar la cantidad de trabajo realizado en el mismo.

### 1.8.2.3 Doctrine

“The Doctrine Project is the home of a selected set of PHP libraries primarily focused

on providing persistence services and related functionality. Its prize projects are a Object Relational Mapper and the Database Abstraction Layer it is built on top of.”[17]

Es una de las herramientas esenciales integradas en de Symfony 2, proporciona toda la capa de persistencia y abstracción de base de datos, impulsando todas las tareas asignadas al modelo desde el controlador.

La capa de abstracción y acceso a base de datos DBAL<sup>8</sup> es una API que se implementa sobre PDO<sup>9</sup> para unificar la comunicación entre las aplicaciones y las bases de datos, centralizando el acceso a los *drivers* de cada una de las plataformas, esquemas y los tipos de variables. Las bases de datos soportadas son las siguientes:

- MySQL
- Oracle
- Microsoft SQL Server
- PostgreSQL
- SAP Sybase SQL Anywhere
- SQLite
- Drizzle

Sobre la capa de abstracción a base de datos se encuentra el mapeo objeto-relacional ORM<sup>10</sup>, que entre una de las principales funciones es la opción de escribir sentencias para la base de datos utilizando Doctrine Query Language (DQL), el cual es un dialecto SQL orientado a objetos, lo que facilita en gran medida el acceso a los datos al traerlos a PHP directamente como objetos.

En la Figura 1.8 se muestra gráficamente como interactúa Symfony 2 con Doctrine.

---

<sup>8</sup>**DBAL** Database Abstraction Layer – Capa de abstracción de la base de datos.

<sup>9</sup>**PDO** PHP Data Objects – Extensión que provee una capa de abstracción de acceso a datos para PHP 5.

<sup>10</sup>**ORM** Object-Relational Mapping- Mapeo objeto-relacional

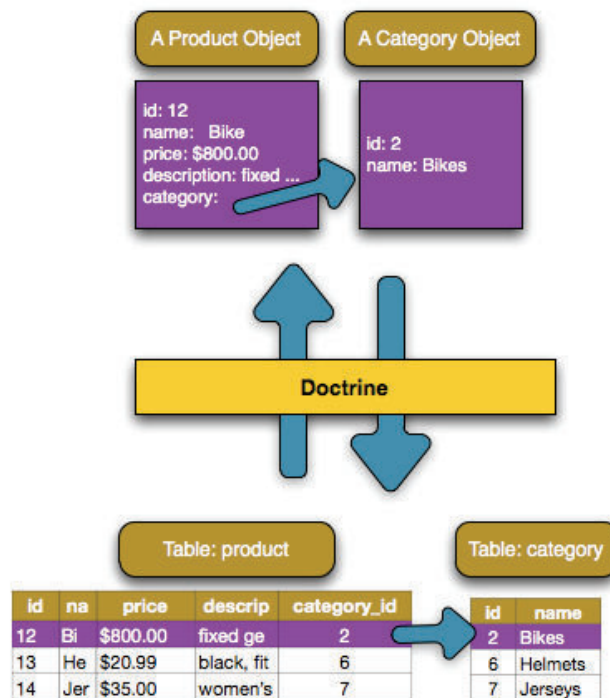


Figura 1.8 Symfony 2 y Doctrine

## 1.9 JAVASCRIPT[18]

“JavaScript is the scripting language of the Web. All modern HTML pages are using JavaScript”[19]

“JavaScript® es el lenguaje interpretado orientado a objetos desarrollado por Netscape que se utiliza en millones de páginas web y aplicaciones de servidor en todo el mundo.”[18]

Javascript es el lenguaje que se enfoca principalmente en presentar la mayor interoperabilidad entre un sistema web y el usuario, es decir permite la ejecución de procedimientos simples o complejos directamente en el browser del usuario ayudando a que la experiencia en la manipulación del sistema sea mucho más amigable e intuitiva en los entornos web.

Es un lenguaje ligero y orientado a objetos, por lo cuál ha sido uno de los lenguajes



con mayor aceptación actualmente, logrando incluso que comunidades de software libre se interesen y desarrollen frameworks que faciliten a los desarrolladores la creación de código más eficiente, ordenado y limpio.

Entre los frameworks más utilizados y con mayores comunidades se mencionan los siguientes: JQuery, Mootools, Prototype, Backbone.js, entre otros. Siendo de estos el más usado JQuery.

Aunque la utilización principal de Javascript ha sido por muchos años en el lado del cliente, en los últimos años se ha expandido para permitir la utilización del mismo en el lado del servidor, plataformas como Node.js han hecho que la popularidad en la utilización de Javascript en el servidor aumente. Incluso se ha expandido tanto su uso en los diferentes campos de la tecnología que existen implementaciones que permiten la utilización de lenguajes como Javascript para la creación de aplicaciones móviles.

### 1.9.1 JQUERY[20]

“jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.”[20]

Es el *framework* de Javascript más importante, simplifica en gran medida la programación mediante el uso de nuevas funcionalidades e implementaciones, cambia por completo la forma de escribir código Javascript, disminuyendo considerablemente el tiempo de implementación y generando un código mucho más fácil de leer e interpretar. Su gran aceptación la logró gracias a que facilita a los desarrolladores la creación de tareas rutinarias en menos tiempo y sin ninguna carga significativa en la velocidad de ejecución.

Es compatible con la mayoría de navegadores actuales y las comunidades adeptas a la creación de complementos basadas en jQuery crecen continuamente, dándole una gran cantidad de opciones, mejoras y funcionalidades como: calendarios, tablas con paginación, eventos drag and drop, animaciones, entre muchas otras.

## CAPÍTULO 2

### 2. DISEÑO DEL SISTEMA INTEGRAL

#### 2.1 ANÁLISIS DE REQUERIMIENTOS

Para el análisis de requerimientos se tomará en cuenta las características del sistema integral que se indican a continuación:

- Implementación basada en Web.
- Administración y gestión de roles y usuarios.
- Administración y gestión de proyectos y etapas.
- Administración y gestión de departamentos.
- Soporte de voz al cliente.
- Integración con chat de soporte.
- Chat interno para usuarios.
- Integración con la telefonía IP mediante softphone web.
- Integración del chat de soporte con aplicación móvil para iPhone.
- Notificaciones en la aplicación móvil de nuevos mensajes y conversaciones.

También se debe considerar las especificaciones particulares de la implementación del prototipo que se muestran a continuación:

- Orientado para empresas tipo integradoras de servicios.
- 10 usuarios mínimo para soporte técnico.
- 5 usuarios mínimo para soporte de ventas.
- Pruebas en ambiente de laboratorio.

Para garantizar el correcto funcionamiento del sistema prototipo, el dimensionamiento de los recursos como el *hardware* en el cuál operará el sistema y la conexión a internet que depende del tráfico generado, se lo realizará luego de la implementación del mismo basado en los resultados de las pruebas de carga.

Basado en las características y especificaciones mencionadas es necesario realizar un análisis específico de los siguientes requerimientos:

- Central telefónica
- Softphone web
- Chat de soporte open source
- Lenguajes de programación

### 2.1.1 CENTRAL TELEFÓNICA

Para la implementación del prototipo se utilizará una central telefónica IP con soporte SIP, la cual permita la integración con el softphone web y la interacción de la telefonía de la empresa tipo. Para este prototipo se implementará mediante una central telefónica marca Masterbox IP, pero su diseño será orientado a cualquier central telefónica IP con similares características.

En la Tabla 2.1 se indican las características básicas de la central telefónica a considerar para cumplir las especificaciones del prototipo.

**Tabla 2.1** Características básicas central telefónica

<b>REQUERIMIENTOS BÁSICOS</b>	
Protocolos	SIP, RTP
Códecs de Audio	G711 o G729
Extensiones	15 o superior
Troncales	Mínimo 1
Interfaces de Red	Mínimo 1

En la Tabla 2.2 se presenta las especificaciones técnicas de la central telefónica que será utilizada en la implementación del prototipo.

**Tabla 2.2** Especificaciones principales central telefónica

<b>INFORMACIÓN GENERAL</b>	
Marca	MasterBox IP

Sistema Operativo	Centos 5.8 64bits
Procesador	Intel® Pentium® Processor G620 Dual Core
Memoria	2 GB
Disco Duro	500 GB
Interfaces de Red	1
IP Local	192.168.1.16
Extensiones	50
Troncales	Ilimitadas

#### **2.1.1.1 Extensiones**

La cantidad de extensiones se definen según la cantidad de usuarios que se requieran en la plataforma web para el uso del softphone web y los empleados de la empresa que tienen asignada una extensión en su teléfono SIP o softphone.

Para el prototipo la central telefónica utilizada soporta una cantidad de 50 extensiones según las especificaciones técnicas de la misma. Esto cubre el requerimiento de 15 usuarios de soporte técnico y ventas, además de dar una escalabilidad de 35 extensiones para usuarios clientes del sistema, pues a cada cliente se le asignará una extensión para que pueda realizar llamadas a los operadores de soporte.

#### **2.1.1.2 Troncal**

Para la implementación del prototipo se configurará en la central telefónica, una troncal que con la ayuda de un Gateway FXO comunique mediante una base celular a la telefonía pública.

En el caso de una empresa tipo se podría usar el Gateway FXO, para conectar líneas analógicas o utilizar directamente una troncal SIP y configurar en la central telefónica los datos de registro que entregue el proveedor.

## 2.1.2 SOFTPHONE WEB

Para la implementación del prototipo se debe contar con un softphone web que permita manejar un ambiente híbrido, en donde si el navegador soporta se pueda utilizar Websockets y WebRTC, caso contrario utilice Flash. Esto proporciona la escalabilidad necesaria para este tipo de aplicaciones, puesto que las tendencias actuales indican que las aplicaciones embebidas basadas en Flash cada día son menores por los tiempos de carga y el consumo de recursos. A continuación se indicarán algunas soluciones que cumplen con las características mencionadas.

### 2.1.2.1 Plivo

Esta solución implementa una librería de Javascript que permite incorporar en un sitio web las funcionalidades para realizar llamadas mediante el protocolo SIP. Esta solución es exclusiva para el lado del cliente y no requiere ningún tipo de configuración en el lado del servidor, ya que lo ofrecen como servicio pagado por el tiempo de cada llamada generada.

### 2.1.2.2 Twilio

Al igual que la solución anterior es una solución basada en una librería de Javascript que de idéntica forma requiere de la adquisición de un servicio para el lado del servidor, pagado en función del tiempo involucrado en las llamadas realizadas.

### 2.1.2.3 Flashphoner

De igual forma que las soluciones anteriores cuenta con una librería de Javascript que permite implementarlo en cualquier aplicación web. Es la única solución de las mencionadas que incluye el *software* para el servidor, de forma que el mismo sea instalado en los servidores del acreedor del software.

Para la integración en el prototipo se ha escogido Flashphoner puesto que ofrece una solución completa sin depender de contratar un servicio adicional y permite el uso del softphone web en la red local sin depender estrictamente de una conexión a internet.

### **2.1.3 CHAT DE SOPORTE OPEN SOURCE**

El software open source que se utilizará para este propósito tiene el nombre Mibew Messenger, que se encuentra actualmente en su versión 1.6, este software cuenta con las características necesarias que se aplicarán en el sistema web. El software esta realizado mediante PHP y base de datos MySQL. Se distribuye bajo licencia Apache 2.0.

A continuación se mostrarán algunas de las características principales del chat de soporte:

- Listado de visitantes en espera
- Historial de mensajes
- Estadísticas
- Bloqueo de usuarios
- Administración de operadores
- Mensajes frecuentes
- Grupos o departamentos

Para la instalación del software son necesarios los siguientes requerimientos:

- Servidor web
- PHP 5.3.3 o superior
- Extensiones de PHP como PDO, *pdo\_mysql* y *gd*
- MySQL 5.0 o superior

## 2.1.4 LENGUAJES DE PROGRAMACIÓN

En esta sección se definirán los diferentes lenguajes de programación que se utilizarán para el desarrollo de la plataforma web, las interfaces e integraciones requeridas, para lo cual se dividirá entre los lenguajes utilizados en el servidor y los que son utilizados para el cliente.

### 2.1.4.1 SERVIDOR

Para el servidor se necesita un lenguaje de programación que sea de fácil implementación para que el mantenimiento del código no sea complejo, que tenga una gran cantidad de aplicaciones open source para facilitar su integración con otros sistemas, debe permitir ejecutarse sobre Linux para no subir los costos de la implementación en la compra de licencias y debe estar orientado para la Web.

Siendo estos los requerimientos necesarios se tienen opciones como Java, PHP, Python, Ruby. A continuación en la Tabla 2.3 se mostrará el listado de lenguajes más utilizados según la página *redmonk.com* la que está completamente enfocada en el análisis de la industria del desarrollo.

**Tabla 2.3** Ranking 10 lenguajes de programación[21]

RANKING	LENGUAJE DE PROGRAMACIÓN
1	Javascript
2	Java
3	PHP
4	Python
5	C#
6	C++
7	Ruby
8	CSS
9	C
10	Objective-C

Al observar la información de la tabla se evidencia que los lenguajes más utilizados son JavaScript, Java y PHP. Entre estos Javascript no es una opción puesto que



su mayor uso está orientado al lado del cliente. Java y PHP son las mejores opciones por la gran cantidad de soporte que se encuentra.

Considerando que el sistema web será basado completamente para la web, se escogerá a PHP como lenguaje de programación base de la plataforma puesto que su principal enfoque va orientado al desarrollo web, por consecuencia el soporte en este tipo de aplicaciones es mayor que cualquier otro lenguaje y facilitará en gran medida la integración con otras aplicaciones.

#### *2.1.4.1.1 Frameworks*

Los *frameworks* permiten que las aplicaciones se desarrollen fácilmente y sistemáticamente, puesto que las tareas más comunes se encuentran implementadas y se las tiene que adaptar según las necesidades a la aplicación que se desea desarrollar.

En el caso de PHP se utilizará un *framework* denominado Symfony2 que es uno de los *frameworks* más comunes, con mayor comunidad y documentación.

#### **2.1.4.2 CLIENTE**

En el lado del cliente para la implementación de la interfaz y las diferentes funcionalidades que se mostrarán en los navegadores se utilizará como lenguaje de programación Javascript y para la interfaz visual se utilizará el lenguaje de etiquetas HTML junto con CSS(Cascading Style Sheets) para mejorar la presentación de las interfaces. Para facilitar la implementación de Javascript se utiliza el *framework* JQuery que ayuda en tareas como validaciones, selectores de etiquetas, peticiones mediante AJAX entre otras actividades.

También se utilizará como *framework* base Twitter Bootstrap que provee archivos CSS y Javascript, para facilitar la implementación de las vistas teniendo estilos predefinidos como botones, listas, menús, entre otros.

Para la aplicación móvil se utilizará Objective-C, puesto que es el lenguaje más utilizado para el desarrollo de aplicaciones nativas para iPhone.

## **2.2 DISEÑO DE LA PLATAFORMA WEB**

### **2.2.1 MÓDULOS DE LA PLATAFORMA WEB**

Los módulos que se implementarán en la plataforma web para cumplir con todas las funcionalidades son los siguientes:

- Administración de usuarios y roles
- Administración de proyectos
- Gestión de departamentos
- Gestión de llamadas
- Chat de soporte
- Servicios web

### **2.2.2 ADMINISTRACIÓN DE USUARIOS Y ROLES**

La administración de usuarios y roles permiten al sistema integrar funcionalidades de acceso según permisos a los diferentes módulos del sistema.

#### **2.2.2.1 Usuarios**

Los usuarios deben tener información general como nombre, apellido, contraseña, dirección y email. También es importante que el usuario pueda asociarse con una extensión para que pueda generar y recibir llamadas. La asignación de cada usuario de la empresa a un departamento permite la agrupación según las funciones que desempeña. También se debe tener una relación con los proyectos asignados y la etapa en la cual se lo asignó.

### **2.2.2.2 Roles**

Los usuarios serán designados a un rol en específico según el cual se definirá los permisos y la interfaz a la que accederá. Los roles también se diferenciarán según el cargo que ocupa en la empresa y sus funciones.

#### *2.2.2.2.1 Rol Cliente*

Permite tener acceso a la información de los proyectos que pertenecen al cliente, ver los técnicos asociados a sus proyectos y poder comunicarse con ellos.

#### *2.2.2.2.2 Rol Administración*

Es el rol con permisos globales en el sistema, permite la gestión de usuarios y asignación de roles, gestión de proyectos, etapas, gestión de empresas, generación de llamadas, gestión de mensajes.

#### *2.2.2.2.3 Rol Jefe Departamento*

Tiene permisos para crear asignación de proyectos, gestión de proyectos, gestión de empresas, generación y recepción de llamadas, gestión chat de soporte.

El usuario con este rol puede realizar asignaciones de proyectos a los usuarios que pertenezcan a su departamento.

#### *2.2.2.2.4 Rol Usuario*

Permite mostrar la información sobre los proyectos asignados y los clientes, iniciar la consola del chat de soporte, además de generar y recibir llamadas.

### **2.2.3 ADMINISTRACIÓN DE PROYECTOS**

Este módulo debe permitir gestión de proyectos, mostrar información detallada de cada proyecto, gestionar etapas y los usuarios asociados a cada una de ellas.

#### **2.2.3.1 Gestión de Proyectos**

Permite crear, editar y eliminar proyectos, la información básica asociada a cada proyecto son: el nombre, las fechas de ejecución del proyecto, la empresa, un detalle básico del proyecto y las etapas en que se divide.

#### **2.2.3.2 Información Detallada de Proyecto**

Esta vista mostrará la información básica de cada proyecto, las fechas de ejecución del mismo. Además mostrará todas las etapas en que se divide el proyecto delimitadas entre fechas.

#### **2.2.3.3 Creación de Etapas**

Esta interfaz permitirá dividir los proyectos en etapas, asignar un nombre único a cada una de ellas, además muestra los rangos de fechas disponibles para la creación de la misma facilitando la selección entre todo el periodo de tiempo disponible o entre fechas parciales.

#### **2.2.3.4 Asignación de Usuarios a Proyectos**

Cada etapa permite designar a los usuarios con una prioridad específica acorde al departamento al que pertenecen y esta actividad será realizada por los jefes de departamento.

#### **2.2.4 GESTIÓN DE DEPARTAMENTOS**

Este módulo permite organizar a los usuarios según el departamento al que pertenecen, también incluye funcionalidades para crear, editar y eliminar los mismos. La vista detallada muestra la información de los usuarios asociados a este departamento.

#### **2.2.5 GESTIÓN DE LLAMADAS**

Este módulo se encarga de integrar al sistema un softphone web desarrollado por terceros, el cual se registrará con el protocolo SIP a una central telefónica para generar llamadas a otras extensiones o a la telefonía pública. Los registros de llamadas entrantes y salientes quedarán almacenados para mantener un control o seguimiento de las mismas. Todas las llamadas son asociadas a los usuarios involucrados y a un proyecto si esta fue generada por un usuario cliente.

En el sistema se implementará diferentes interfaces para la generación de llamadas que serán mostradas según el rol del usuario autenticado en la plataforma, las cuales pueden ser *click-to-call*, marcación a departamento, teclado de marcación y marcación entre operadores.

##### **2.2.5.1 Click-to-call**

Esta es la manera para establecer una comunicación entre el usuario cliente y la empresa, se realiza mediante la incorporación de un botón o un enlace que al presionarlo genera una llamada desde el navegador del cliente directamente a una extensión de un usuario de la empresa que se puede encontrar registrada mediante el sistema, un softphone o un teléfono IP físico.

### **2.2.5.2 Marcación a Departamento**

Muestra una lista de los departamentos involucrados en el proyecto escogido previamente y permite realizar una llamada a cualquiera de ellos, redirigiendo la llamada a la extensión con mayor prioridad. De esta manera el cliente se comunicará directamente con el usuario que tiene más información sobre el estado de su proyecto para no perder tiempo comunicándose con personal del mismo departamento que no mantiene un seguimiento constante en el proyecto.

### **2.2.5.3 Teclado de Marcación**

Permite realizar una llamada a cualquier extensión o número de telefonía pública empleando la infraestructura telefónica de la empresa, está orientado exclusivamente para personal de soporte para comunicarse sin necesidad de un softphone instalado en el computador o un teléfono IP físico. La interfaz mostrará un teclado similar a un softphone tradicional instalado en el computador.

### **2.2.5.4 Marcación entre operadores**

Esta interfaz permitirá realizar llamadas entre usuarios operadores mediante un listado de contactos, que muestra el estado de los usuarios conectados y desconectados. Este listado de contactos se mostrará en todas las instancias de la interfaz de operadores permitiendo realizar y recibir llamadas.

## **2.2.6 CHAT DE SOPORTE**

Para el chat de soporte el sistema se integra a una aplicación de código abierto, en la cual los usuarios de la empresa podrán brindar soporte mediante mensajes a los clientes, la singularidad de un chat de soporte radica en que cada usuario del sistema que pertenece a la empresa pasa a ser agente y pueden atender uno a varios clientes que se comunican en busca de soporte.

## 2.2.7 SERVICIOS WEB

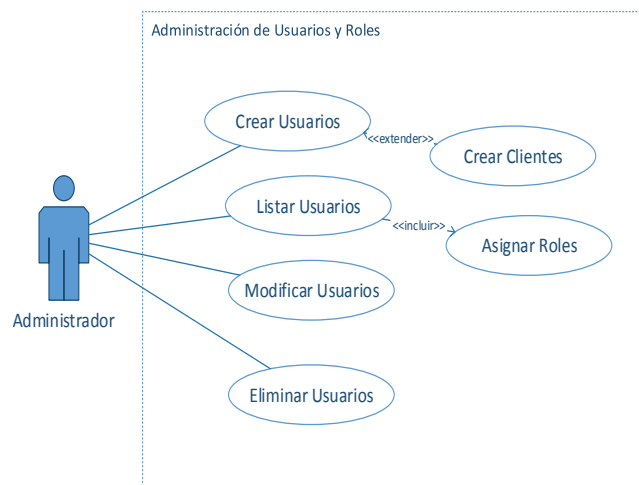
La implementación de servicios web en la plataforma se debe realizar para crear una integración entre la plataforma web y los dispositivos móviles. Las funciones implementadas en este servicio web deben proveer información sobre las conversaciones, los mensajes y los usuarios asociados.

## 2.3 DIAGRAMA DE CASOS DE USO DE LA PLATAFORMA WEB

En esta sección se muestra los diagramas de casos de uso que se crearon basados en los módulos de la plataforma web.

### 2.3.1 Diagrama Casos de Uso – Administración de Usuarios y Roles

En la Figura 2.1 se muestra los casos de uso involucrados en la Administración de Usuarios y Roles.



**Figura 2.1** Casos de Uso – Administración de Usuarios y Roles

### 2.3.2 Descripción de Casos de Uso – Administración de Usuarios y Roles

En la Tabla 2.4 se observa la descripción del caso de uso Crear Usuario que se encuentra incluido en la Figura 2.1.

**Tabla 2.4** Descripción Caso de Uso – Crear Usuarios

<b>Caso de Uso</b>	<b>Crear Usuarios</b>
<b>Descripción</b>	Permite al administrador crear un usuario nuevo
<b>Actores</b>	Administrador
<b>Precondiciones</b>	Autenticarse como Administrador
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El Administrador accede a la interfaz Administración de Usuarios desde el menú.</li> <li>2) Accede a Crear Nuevo</li> <li>3) Ingresa la información del Usuario</li> <li>4) Asigna uno o varios roles (Cliente, Usuario, Jefe, Administrador)</li> <li>5) Asigna la información de la extensión asociada</li> <li>6) Presiona el botón Crear</li> <li>7) El sistema comprueba la validez de la información y la almacena en la base de datos del sistema y del chat de soporte.</li> </ol>
<b>Flujo Alternativo</b>	7A) El sistema comprueba la información ingresada, si los datos no son correctos informa al actor de los errores para que los corrija.
<b>Postcondiciones</b>	El usuario se almacena en la base de datos y queda activo para el acceso a la plataforma.

En la Tabla 2.5 se observa la descripción del caso de uso Listar Usuarios que se encuentra incluido en la Figura 2.1.

**Tabla 2.5** Descripción Caso de Uso – Listar Usuarios

<b>Caso de Uso</b>	<b>Listar Usuarios</b>
<b>Descripción</b>	Permite al administrador listar todos los usuarios del sistema
<b>Actores</b>	Administrador
<b>Precondiciones</b>	Autenticarse como Administrador
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El Administrador accede a la interfaz Administración de Usuarios desde el menú.</li> <li>2) El sistema lista en un tabla todos los usuarios que se encuentran en la plataforma</li> </ol>
<b>Flujo Alternativo</b>	No aplica
<b>Postcondiciones</b>	No aplica

En la Tabla 2.6 se observa la descripción del caso de uso Modificar Usuario que se encuentra incluido en la Figura 2.1.

**Tabla 2.6** Descripción Caso de Uso – Modificar Usuarios

<b>Caso de Uso</b>	<b>Modificar Usuario</b>
<b>Descripción</b>	Permite modificar la información de cada usuarios del sistema
<b>Actores</b>	Administrador
<b>Precondiciones</b>	Autenticarse como Administrador



<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El Administrador accede a la interfaz Administración de Usuarios desde el menú.</li> <li>2) Escoge un usuario y accede a la opción Editar</li> <li>3) Realiza los cambios en la información</li> <li>4) Presiona el botón Editar</li> <li>5) El sistema comprueba la validez de la información y la almacena en la base de datos del sistema y del chat de soporte.</li> </ol>
<b>Flujo Alternativo</b>	5A) El sistema comprueba la información ingresada, si los datos no son correctos informa al actor de los errores para que los corrija.
<b>Postcondiciones</b>	La información es modifica en las bases de datos del sistema y del chat de soporte.

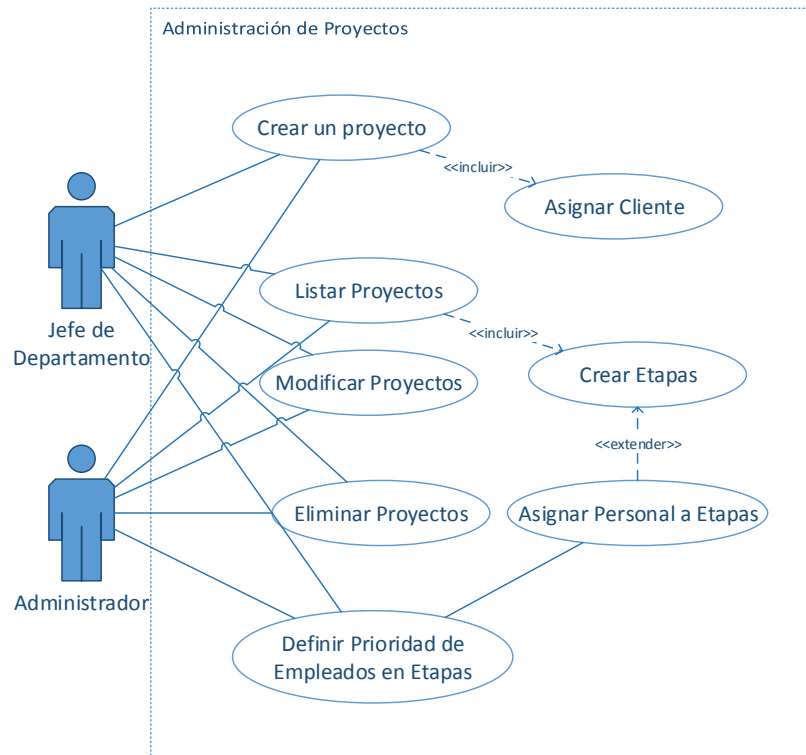
En la Tabla 2.7 se observa la descripción del caso de uso Eliminar Usuarios que se encuentra incluido en la Figura 2.1.

**Tabla 2.7** Descripción Caso de Uso – Eliminar Usuarios

<b>Caso de Uso</b>	<b>Eliminar Usuario</b>
<b>Descripción</b>	Permite eliminar los usuarios del sistema
<b>Actores</b>	Administrador
<b>Precondiciones</b>	Autenticarse como Administrador
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El Administrador accede a la interfaz Administración de Usuarios desde el menú.</li> <li>2) Escoge un usuario y accede a la opción Ver o Editar</li> <li>3) Presiona el botón Eliminar</li> <li>4) El sistema elimina el usuario del sistema y del chat de soporte</li> </ol>
<b>Flujo Alternativo</b>	No aplica
<b>Postcondiciones</b>	El usuario es eliminado de las bases de datos del sistema y del chat de soporte

### 2.3.3 Diagrama Casos de Uso – Administración de Proyectos

En la Figura 2.2 se observa los casos de uso involucrados en la Administración de Proyectos.



**Figura 2.2** Casos de Uso – Administración de Proyectos

### 2.3.4 Descripción de Casos de Uso – Administración de Proyectos

En la Tabla 2.8 se observa la descripción del caso de uso Crear un Proyecto que se encuentra incluido en la Figura 2.2.

**Tabla 2.8** Descripción Caso de Uso – Crear un proyecto

Caso de Uso	Crear un proyecto
Descripción	Permite la creación de nuevos proyectos
Actores	Administrador, Jefe de Departamento
Precondiciones	Autenticarse como Administrador o Jefe de Departamento
Flujo Normal	<ol style="list-style-type: none"> <li>1) El actor accede a la interfaz Listado de Proyectos desde el menú.</li> <li>2) Presiona el botón Crear Nuevo Proyecto</li> <li>3) Ingresa la información del proyecto</li> <li>4) Define las fechas en que se va a ejecutar el proyecto</li> <li>5) Escoge el Cliente – Empresa al cual pertenece el proyecto</li> <li>6) Presiona el botón Crear</li> <li>7) El sistema verifica que la información sea correcta y procede a almacenarlo en la base de datos del sistema</li> </ol>
Flujo Alternativo	7A) El sistema verifica la consistencia de la información ingresada, si los datos no son correctos informa al actor de los errores para que los corrija
Postcondiciones	El proyecto se almacena en la base de datos

En la Tabla 2.9 se observa la descripción del caso de uso Listar Proyectos que incluye el caso de uso Crear Etapas los cuales son parte del diagrama en la Figura 2.2.

**Tabla 2.9** Descripción Caso de Uso – Listar Proyectos

<b>Caso de Uso</b>	<b>Listar Proyectos -&gt; Crear Etapas</b>
<b>Descripción</b>	Permite al actor listar todos los proyectos
<b>Actores</b>	Administrador, Jefe de Departamento
<b>Precondiciones</b>	Autenticarse como Administrador o Jefe de Departamento
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El actor accede a la interfaz Listado de Proyectos desde el menú.</li> <li>2) El sistema lista todos los proyectos que se encuentran en el sistema</li> <li>3) El actor selecciona un proyecto y presiona el botón Etapas</li> <li>4) El actor presiona el botón nueva etapa</li> <li>5) Ingresa la información de la etapa</li> <li>6) Asigna usuarios al proyecto ordenados según la prioridad</li> <li>7) Presiona el botón crear</li> <li>8) El sistema verifica la información y la almacena en la base de datos</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>4A) Si el proyecto no tiene tiempo para más etapas informa al actor</li> <li>8A) El sistema verifica la consistencia de la información ingresada, si los datos no son correctos informa al actor de los errores para que los corrija</li> </ol>
<b>Postcondiciones</b>	La etapa es creada en el proyecto que se seleccionó

En la Tabla 2.10 se observa la descripción del caso de uso Modificar Proyecto que se encuentra incluido en la Figura 2.2.

**Tabla 2.10** Descripción Caso de Uso – Modificar proyecto

<b>Caso de Uso</b>	<b>Modificar proyecto</b>
<b>Descripción</b>	Permite la modificación de un proyecto
<b>Actores</b>	Administrador, Jefe de Departamento
<b>Precondiciones</b>	Autenticarse como Administrador o Jefe de Departamento
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El actor accede a la interfaz Listado de Proyectos desde el menú.</li> <li>2) Presiona el botón Editar</li> <li>3) Edita la información del proyecto</li> <li>4) Edita las fechas en que se va a ejecutar el proyecto si lo requiere</li> <li>5) Edita el Cliente – Empresa al cual pertenece el proyecto</li> <li>6) Presiona el botón Actualizar</li> <li>7) El sistema verifica que la información sea correcta y procede a almacenarlo en la base de datos del sistema</li> </ol>
<b>Flujo Alternativo</b>	7A) El sistema verifica la consistencia de la información ingresada y que las fechas no afecten a ninguna etapa, si los datos no son correctos informa al actor de los errores para que los corrija
<b>Postcondiciones</b>	El proyecto es actualizado en la base de datos

En la Tabla 2.11 se observa la descripción del caso de uso Eliminar Proyecto que se encuentra incluido en la Figura 2.2.

**Tabla 2.11** Descripción Caso de Uso – Eliminar proyecto

<b>Caso de Uso</b>	<b>Eliminar proyecto</b>
<b>Descripción</b>	Permite la modificación de un proyecto
<b>Actores</b>	Administrador, Jefe de Departamento
<b>Precondiciones</b>	Autenticarse como Administrador o Jefe de Departamento
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El actor accede a la interfaz Listado de Proyectos desde el menú.</li> <li>2) Presiona el botón Editar o Ver</li> <li>3) Presiona el botón Eliminar</li> <li>4) El sistema elimina el proyecto de la base de datos</li> </ol>
<b>Flujo Alternativo</b>	No aplica
<b>Postcondiciones</b>	El proyecto las etapas y las asignaciones de usuarios son eliminados de la base de datos

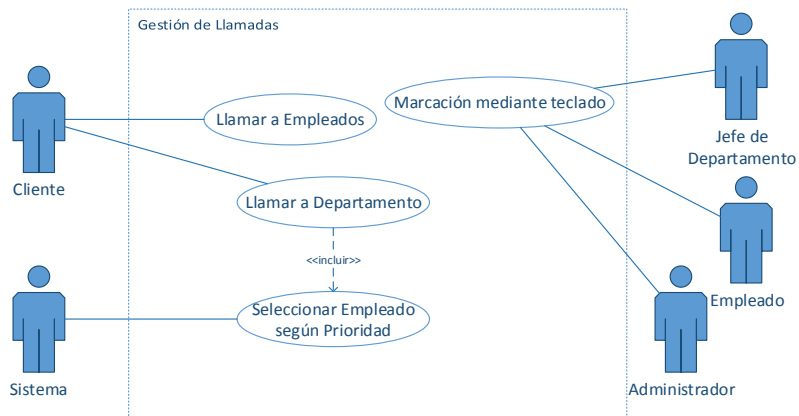
En la Tabla 2.12 se observa la descripción del caso de uso Definir Prioridad de Empleados en Etapas que incluye Asignar Personal a Etapa los cuales forman parte de la Figura 2.2.

**Tabla 2.12** Descripción Caso de Uso – Definir Prioridad de Empleados en Etapas

<b>Caso de Uso</b>	<b>Definir Prioridad de Empleados en Etapas -&gt; Asignar Personal a Etapas</b>
<b>Descripción</b>	Permite asignar y definir las etapas
<b>Actores</b>	Administrador, Jefe de Departamento
<b>Precondiciones</b>	Autenticarse como Administrador o Jefe de Departamento
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El actor accede a la interfaz Listado de Proyectos desde el menú.</li> <li>2) El sistema lista todos los proyectos que se encuentran en el sistema</li> <li>3) El actor selecciona un proyecto y presiona el botón Etapas</li> <li>4) El actor selecciona una etapa y presiona el botón Editar</li> <li>5) Asigna usuarios al proyecto ordenados según la prioridad</li> <li>6) Presiona el botón Actualizar</li> <li>7) El sistema verifica la información y la almacena en la base de datos</li> </ol>
<b>Flujo Alternativo</b>	7A) El sistema verifica la consistencia de la información ingresada, si los datos no son correctos informa al actor de los errores para que los corrija
<b>Postcondiciones</b>	La etapa es actualizada y las relaciones con los usuarios creadas en la base de datos.

### 2.3.5 Diagrama Casos de Uso – Gestión de Llamadas

En la Figura 2.3 se observa los casos de uso involucrados en la Gestión de Llamadas.



**Figura 2.3** Casos de Uso – Gestión de Llamadas

### 2.3.6 Descripción de Casos de Uso – Gestión de Llamadas

En la Tabla 2.13 se observa la descripción del caso de uso Llamar a Empleados que se encuentra incluido en la Figura 2.3.

**Tabla 2.13** Descripción Caso de Uso – Llamar a Empleados

Caso de Uso	Llamar a Empleados
Descripción	Generación de llamadas a los empleados
Actores	Cliente
Precondiciones	Autenticarse como Cliente
Flujo Normal	<ol style="list-style-type: none"> <li>1. El actor desde su perfil escoge un proyecto</li> <li>2. Presiona el botón Soporte del proyecto seleccionado</li> <li>3. El sistema muestra los empleados/usuarios asignados a la etapa actual.</li> <li>4. El actor escoge el empleado al que quiere llamar</li> <li>5. Presiona el botón llamar del empleado seleccionado</li> <li>6. El sistema realiza una llamada al usuario destino</li> </ol>
Flujo Alternativo	No aplica
Postcondiciones	El registro es almacenado en la base de datos al finalizar la llamada

En la Tabla 2.14 se observa la descripción del caso de uso Llamar a Departamento que incluye Seleccionar Empleado según Prioridad los cuales forman parte de la Figura 2.3.

**Tabla 2.14** Descripción Caso de Uso – Llamar a Departamento

<b>Caso de Uso</b>	<b>Llamar a Departamento -&gt; Seleccionar Empleado según Prioridad</b>
<b>Descripción</b>	Generación de llamadas a departamentos
<b>Actores</b>	Cliente
<b>Precondiciones</b>	Autenticarse como Cliente
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El actor desde su perfil escoge un proyecto</li> <li>2) Presiona el botón Departamentos del proyecto seleccionado</li> <li>3) El sistema muestra los departamentos asignados a la etapa actual.</li> <li>4) El actor escoge el departamento al que quiere llamar</li> <li>5) Presiona el botón llamar del departamento seleccionado</li> <li>6) El sistema realiza una llamada al usuario con mayor prioridad</li> </ol>
<b>Flujo Alternativo</b>	No aplica
<b>Postcondiciones</b>	El registro es almacenado en la base de datos al finalizar la llamada

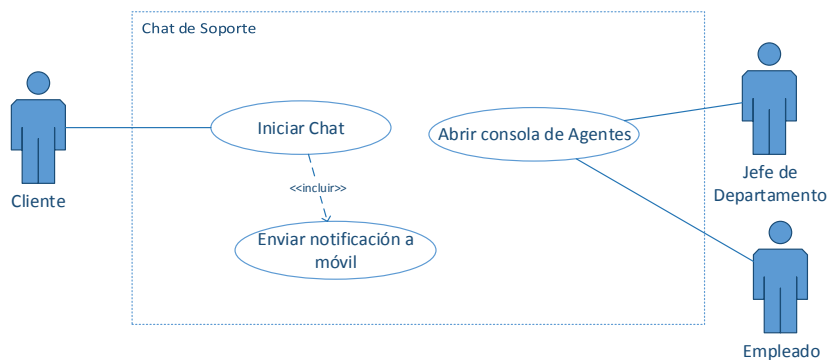
En la Tabla 2.15 se observa la descripción del caso de uso Marcación mediante teclado que se encuentra incluido en la Figura 2.3.

**Tabla 2.15** Descripción Caso de Uso – Marcación mediante teclado

<b>Caso de Uso</b>	<b>Marcación mediante teclado</b>
<b>Descripción</b>	Generación de llamadas a extensión o números de la telefonía pública
<b>Actores</b>	Jefe de Departamento, Empleado, Administrador
<b>Precondiciones</b>	Autenticarse como Empleado, Administrador o Jefe de Departamento
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El actor desde accede a Llamadas por Teclado mediante el menú</li> <li>2) El sistema muestra un teclado de marcación</li> <li>3) El usuario marca el número o extensión</li> <li>4) El sistema genera la llamada hacia el número destino</li> </ol>
<b>Flujo Alternativo</b>	No aplica
<b>Postcondiciones</b>	No aplica

### 2.3.7 Diagrama Casos de Uso – Chat de Soporte

En la Figura 2.4 se observa los casos de uso involucrados en el Chat de Soporte.



**Figura 2.4** Casos de Uso – Chat de Soporte

### 2.3.8 Descripción de Casos de Uso – Chat de Soporte

En la Tabla 2.16 se observa la descripción del caso de uso Enviar Mensaje que incluye Enviar Notificación a móvil los cuales forman parte de la Figura 2.4.

**Tabla 2.16** Descripción Caso de Uso – Enviar Mensaje

Caso de Uso	Enviar Mensaje -> Enviar Notificación a móvil
Descripción	Envío de mensajes entre clientes y operadores del sistema
Actores	Cliente
Precondiciones	Autenticarse como Cliente
Flujo Normal	<ol style="list-style-type: none"> <li>1) El actor empieza una nueva conversación presionando el botón nuevo chat que se encuentra en la parte inferior derecha de todas las vistas.</li> <li>2) El sistema abre una nueva ventana de chat y notifica a todos los agentes conectados al sistema.</li> <li>3) El sistema envía una notificación a todos los usuarios conectados en la aplicación móvil.</li> </ol>
Flujo Alternativo	No aplica
Postcondiciones	No aplica

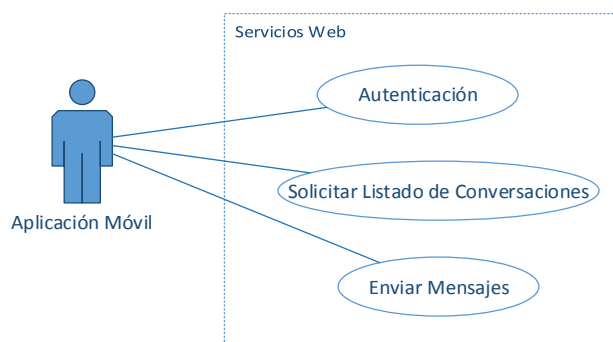
En la Tabla 2.17 se observa la descripción del caso de uso Abrir consola de Agente que se encuentra incluido en la Figura 2.4.

**Tabla 2.17** Descripción Caso de Uso – Abrir consola de Agente

<b>Caso de Uso</b>	<b>Abrir consola de Agente</b>
<b>Descripción</b>	Abrir consola para el Chat de Soporte
<b>Actores</b>	Jefe de Departamento, Empleado
<b>Precondiciones</b>	Autenticarse como Jefe de Departamento o Empleado
<b>Flujo Normal</b>	1) El actor ingresa a Activar Chat desde el menú 2) El sistema autentica en el chat de Soporte 3) El sistema muestra la interfaz principal del chat de soporte
<b>Flujo Alternativo</b>	No aplica
<b>Postcondiciones</b>	No aplica

### 2.3.9 Diagrama Casos de Uso – Servicios Web

En la Figura 2.5 se observa los casos de uso involucrados en los Servicios Web.

**Figura 2.5** Casos de Uso – Servicios Web

### 2.3.10 Descripción de Casos de Uso – Servicios Web

En la Tabla 2.18 se observa la descripción del caso de uso Autenticación que se encuentra incluido en la Figura 2.5.

**Tabla 2.18** Descripción Caso de Uso – Autenticación

<b>Caso de Uso</b>	<b>Autenticación</b>
<b>Descripción</b>	Autenticación aplicación móvil al sistema
<b>Actores</b>	Aplicación Móvil
<b>Precondiciones</b>	No aplica
<b>Flujo Normal</b>	1) El actor envía una petición al sistema para solicitar la autenticación 2) El sistema verifica que la información sea correcta
<b>Flujo Alternativo</b>	2A) Si la información es incorrecta el sistema informa
<b>Postcondiciones</b>	No aplica



En la Tabla 2.19 se observa la descripción del caso de uso Solicitar Listado de Conversaciones que se encuentra incluido en la Figura 2.5.

**Tabla 2.19** Descripción Caso de Uso – Solicitar Listado de Conversaciones

Caso de Uso	Solicitar Listado de Conversaciones
<b>Descripción</b>	Solicitar el listado de conversaciones del usuario autenticado en la aplicación móvil
<b>Actores</b>	Aplicación Móvil
<b>Precondiciones</b>	Estar autenticado en la aplicación móvil
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El actor envía una petición al sistema para solicitar las conversaciones nuevas y las del usuario autenticado</li> <li>2) El sistema responde la solicitud con la información de las conversaciones</li> </ol>
<b>Flujo Alternativo</b>	No aplica
<b>Postcondiciones</b>	No aplica

En la Tabla 2.20 se observa la descripción del caso de uso Enviar Mensajes que se encuentra incluido en la Figura 2.5.

**Tabla 2.20** Descripción Caso de Uso – Enviar Mensajes

Caso de Uso	Enviar Mensajes
<b>Descripción</b>	Enviar mensajes a una determinada conversación
<b>Actores</b>	Aplicación Móvil
<b>Precondiciones</b>	Estar autenticado en la aplicación móvil
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El actor envía una solicitud con la información del mensaje</li> <li>2) El sistema almacena el mensaje en la base de datos del chat de soporte</li> </ol>
<b>Flujo Alternativo</b>	No aplica
<b>Postcondiciones</b>	No aplica

## 2.4 DIAGRAMA DE CLASES DE LA PLATAFORMA WEB

El diagrama muestra las clases involucradas en el sistema incluyendo los atributos de cada una de las clases y las relaciones que las asocian.

En la Figura 2.6 se puede observar la clase `Usuario` que está relacionada directa o indirectamente con todas las clases del sistema, pues esta clase representa a todos los usuarios y clientes que utilizan el sistema, identificándose mediante la asociación con la clase `Ro1`. La clase `Usuario` se relaciona con la clase `Extension` para vincular a cada usuario una extensión y su respectiva clave de registro para la central telefónica. La relación de usuarios con la clase `Llamada` es doble con el fin de identificar a los usuarios destino y origen que generan y reciben cada una de las llamadas respectivamente, de igual forma la clase `Llamada` se relaciona con la clase `Proyecto` de manera que cuando un usuario cliente genere una llamada a un usuario esta se encuentre asociada a un proyecto.

La clase `Empresa` mantiene una relación uno a muchos con la clase `Proyecto` y esta a su vez una relación similar con la clase `Etapas`, estas relaciones posibilitan asociar a cada empresa los proyectos que esta tenga y cada proyecto segmentarlo en intervalos de tiempo denominados etapas.

La clase `Etapas` se relaciona indirectamente con la clase `Usuario` mediante la clase `Asignación`, esto facilita definir los usuarios que están asignados en las etapas de un proyecto según la prioridad que se le ha definido en cada asignación.

La clase `Departamento` mantiene una relación con la clase `Usuario`, la cual permite agrupar los usuarios empleados en departamentos.

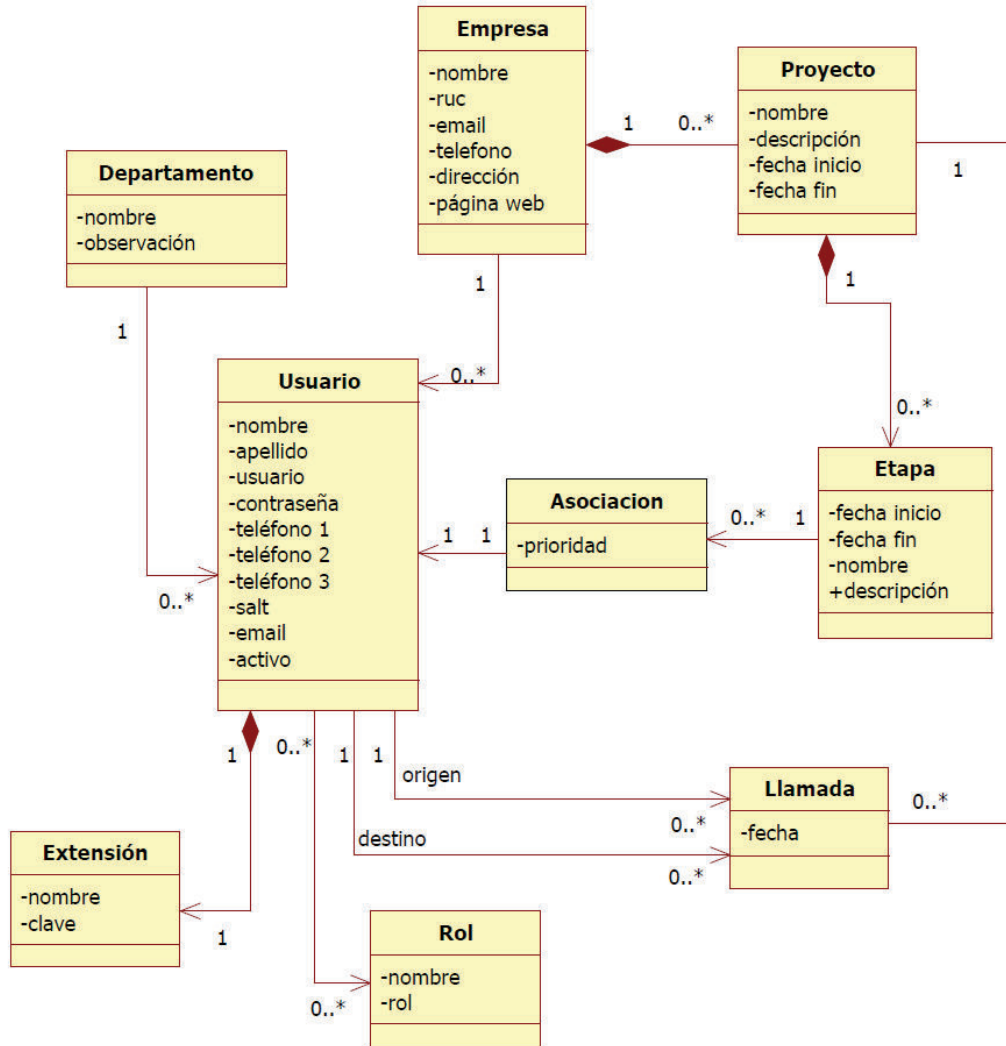
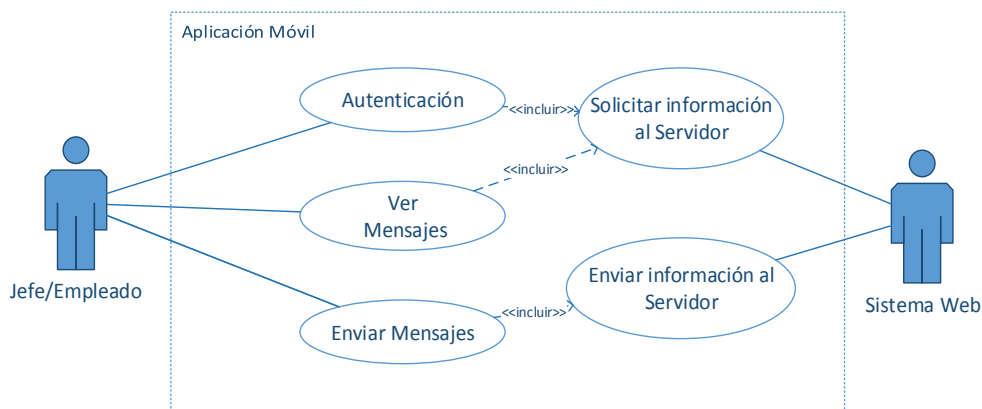


Figura 2.6 Diagrama de clases - Sistema web

## 2.5 DISEÑO APLICACIÓN MÓVIL

### 2.5.1 DIAGRAMA DE CASOS DE USO

En la Figura 2.7 se observa los casos de uso involucrados en la Aplicación Móvil.



**Figura 2.7** Diagrama Casos de Uso – Aplicación Móvil

## 2.5.2 DESCRIPCIÓN CASOS DE USO – APLICACIÓN MÓVIL

En la Tabla 2.21 se observa la descripción del caso de uso Autenticación que incluye Solicitar información al Servidor los cuales forman parte del diagrama de la Figura 2.7.

**Tabla 2.21** Descripción Caso de Uso – Autenticación

Caso de Uso	Autenticación -> Solicitar información al Servidor
Descripción	Autenticación usuario al sistema Web
Actores	Jefe Departamento, Empleado, Sistema Web
Precondiciones	No aplica
Flujo Normal	1) El Jefe de Departamento o Empleado ingresa la información el usuario y contraseña en la pantalla inicial de la aplicación móvil 2) Se presiona el botón Ingresar 3) La aplicación móvil envía la información al Sistema Web y espera una respuesta. 4) El sistema web verifica la información y envía una respuesta positiva 5) La aplicación móvil solicita las conversaciones nuevas y las del usuario autenticado 6) El sistema responde la información solicitada 7) La aplicación muestra un listado de conversaciones
Flujo Alternativo	4A) En caso de que la información no es correcta, el sistema envía una respuesta negativa
Postcondiciones	No aplica

En la Tabla 2.22 se observa la descripción del caso de uso Ver Mensajes que incluye Solicitar información al Servidor los cuales forman parte del diagrama de la Figura 2.7.

**Tabla 2.22** Descripción Caso de Uso – Ver Mensajes

<b>Caso de Uso</b>	<b>Ver Mensajes -&gt; Solicitar información al Servidor</b>
<b>Descripción</b>	Ver mensajes según la conversación escogida
<b>Actores</b>	Jefe Departamento, Empleado, Sistema Web
<b>Precondiciones</b>	Estar autenticado como Jefe de Departamento o Empleado
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El Jefe de Departamento o Empleado da un toque sobre una conversación del listado</li> <li>2) La aplicación móvil solicita al servidor los mensajes asociados a la conversación</li> <li>3) El sistema responde con la información solicitada.</li> <li>4) La aplicación móvil muestra los mensajes en pantalla</li> </ol>
<b>Flujo Alternativo</b>	No aplica
<b>Postcondiciones</b>	No aplica

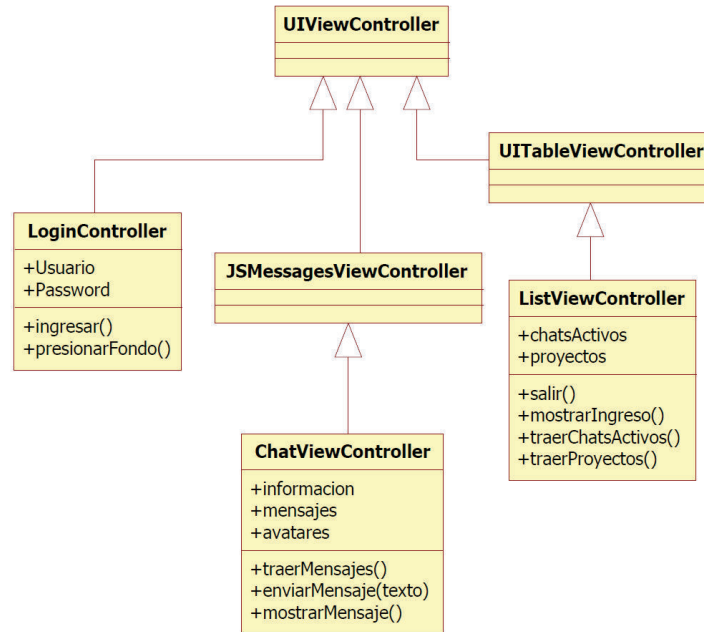
En la Tabla 2.23 se observa la descripción del caso de uso Enviar Mensajes que incluye Enviar información al Servidor los cuales forman parte del diagrama de la Figura 2.7.

**Tabla 2.23** Descripción Caso de Uso – Enviar Mensaje

<b>Caso de Uso</b>	<b>Enviar Mensaje -&gt; Enviar información al servidor</b>
<b>Descripción</b>	Enviar mensajes al Chat de Soporte
<b>Actores</b>	Jefe Departamento, Empleado, Sistema Web
<b>Precondiciones</b>	Estar autenticado como Jefe de Departamento o Empleado Ingresar en una conversación
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) El Jefe de Departamento o Empleado escribe un mensaje</li> <li>2) Presiona el botón Enviar</li> <li>3) La aplicación móvil envía el mensaje al sistema web</li> <li>4) El sistema almacena el mensaje en la base de datos del chat de Soporte</li> </ol>
<b>Flujo Alternativo</b>	No aplica
<b>Postcondiciones</b>	No aplica

### 2.5.3 DIAGRAMA DE CLASES

En la Figura 2.8 se muestran las clases que serán creadas indicando la respectiva clase de la cual heredará. La clase UIViewController y UITableViewController son clases del sistema, mientras que la clase JSMessagesViewController corresponde a la clase del complemento que se utilizará en la implementación del envío y recepción de mensajes y su correspondiente visualización.



**Figura 2.8** Diagrama de Clases – Aplicación móvil

## 2.5.4 DISEÑO DE INTERFACES

### 2.5.4.1 Interfaz de Acceso

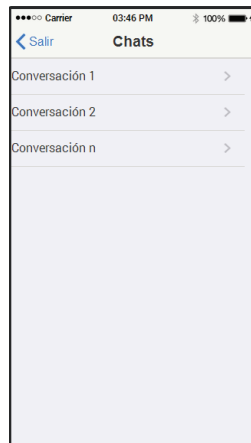
La interfaz de acceso como se observa en la Figura 2.9 permite a los usuarios escribir su nombre de usuario y contraseña para que sean validados con el sistema y permitir el ingreso en la aplicación móvil.

The screenshot shows a simple login screen with a light gray background. At the top, the word "Login" is centered. Below it, there are two input fields: one for "Usuario" and one for "Contraseña". At the bottom, there is a blue button labeled "Ingresar".

**Figura 2.9** Interfaz de acceso

### 2.5.4.2 Interfaz de Conversaciones

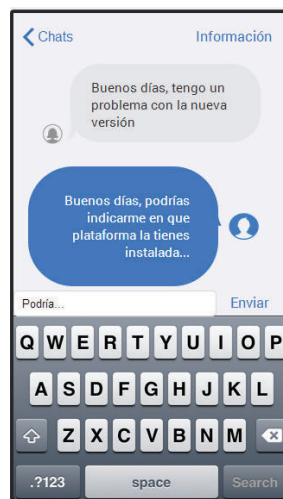
La Figura 2.10 muestra un listado de conversaciones que será cargada como interfaz principal cuando el usuario se encuentra autenticado. En esta interfaz se mostrarán las conversaciones nuevas y las pertenecientes al usuario.



**Figura 2.10** Interfaz de listado de conversaciones

### 2.5.4.3 Interfaz Mensajes

Como se presenta en la Figura 2.11, en esta interfaz aparecen los mensajes de la conversación escogida en la interfaz anterior. De igual forma se puede interactuar con el chat de soporte mediante el envío y recepción de nuevos mensajes.



**Figura 2.11** Interfaz de mensajes

## **2.6 DISEÑO PARA LA GESTIÓN DE LLAMADAS Y REDIRECCIONAMIENTO**

El diseño de la gestión de llamadas restringirá a los usuarios de la plataforma la realización de llamadas a destinatarios no permitidos según el rol asignado a cada usuario. Para esto se configurará en la central telefónica planes de mercado adecuados según las necesidades de una empresa tipo y los roles de los usuarios en la plataforma.

Para mejorar la eficiencia de las llamadas realizadas entre usuarios se diseñarán políticas de redireccionamiento para que los usuarios que no se encuentren conectados o no estén disponibles en la extensión asignada, sean contactados inmediatamente a su respectivo teléfono móvil.

### **2.6.1 PLANES DE MERCADO**

Los planes de mercado en la central telefónica serán enfocados según los permisos y restricciones de mercado que se les dará a cada extensión. Los planes de mercado que serán creados son:

- Nacional
- Celular
- Internacional
- Webphone

A continuación se muestra la Tabla 2.24 con los diferentes permisos según los planes de mercado creados y en la Tabla 2.25 se indica los planes de mercado que serán asignados según el rol del usuario.



**Tabla 2.24** Permisos según planes de marcado

<b>Tipo de Llamada</b>	<b>Plan Nacional</b>	<b>Plan Celular</b>	<b>Plan Internacional</b>	<b>Plan Webphone</b>
Llamadas entre Extensiones	Si	Si	Si	Si
Llamadas Nacionales	Si	Si	Si	No
Llamadas Celulares	No	Si	Si	No
Llamadas Internacionales	No	No	Si	No
Llamada de Emergencias (911) y Servicios (100, 101, 102, etc)	Si	Si	Si	No

**Tabla 2.25** Planes de Marcado según roles

<b>Roles</b>	<b>Plan Nacional</b>	<b>Plan Celular</b>	<b>Plan Internacional</b>	<b>Plan Webphone</b>
Administrador			✓	
Jefe Departamento			✓	
Empleado	✓	✓		
Cliente				✓

### 2.6.2 REDIRECCIONAMIENTO

Los redireccionamientos permitirán a los usuarios de la plataforma facilitar la comunicación cuando estos no se encuentren conectados. Para esto se crearán las redirecciones según las siguientes políticas:

- Las llamadas que no sean contestadas se redireccionarán al teléfono móvil de cada usuario.
- Cuando se ha generado una llamada a un usuario cuya extensión no se encuentra registrado a la central telefónica, automáticamente esta llamada será redireccionada al teléfono móvil del usuario.

- Las llamadas anónimas dirigidas hacia las extensiones asignadas a los usuarios con rol de cliente no serán permitidas.

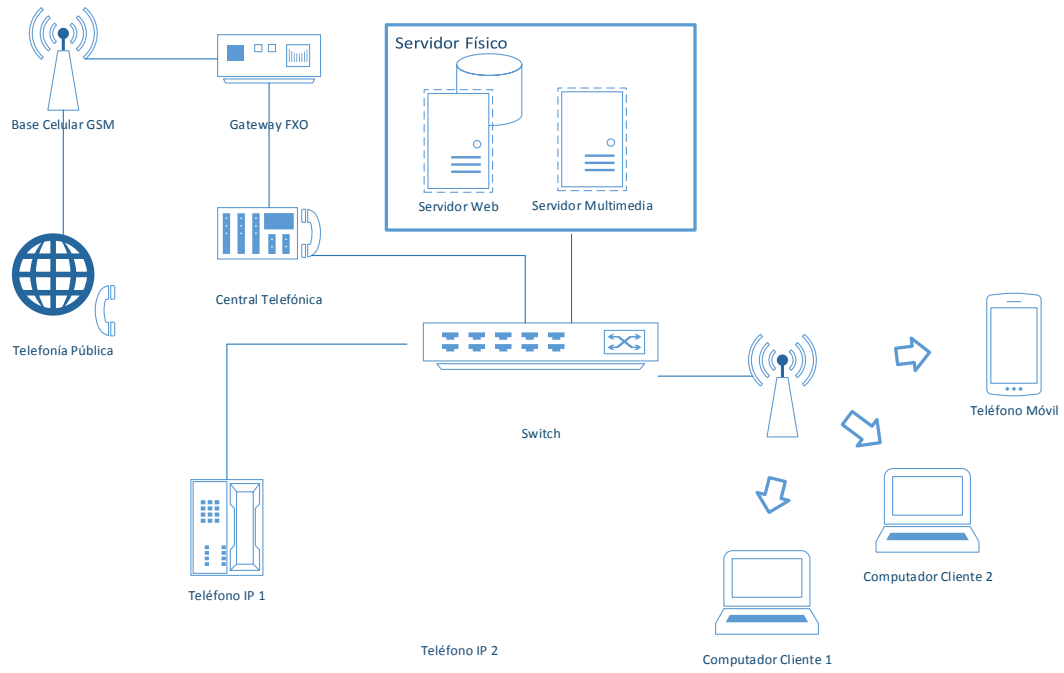
## 2.7 INFRAESTRUCTURA PARA EL PROTOTIPO

Para implementar todas las características necesarias en el prototipo, se utilizarán equipos activos que se encuentran distribuidos según la Figura 2.12.

En la Figura 2.12 se muestra la infraestructura del prototipo para realizar pruebas en un ambiente de laboratorio, para esto se virtualizará el servidor web y el servidor multimedia en un servidor físico con el objetivo de tener una administración centralizada y ocupar menor espacio físico.

La central telefónica se encuentra conectada a la red local para comunicarse con los servidores y los computadores que representarán a los usuarios conectados, de igual forma se encuentra conectada a un Gateway FXO que permitirá redireccionar las llamadas a los teléfonos móviles mediante la base celular GSM.

La función del *switch* y el *access point* son para proveer conectividad entre los equipos. El teléfono móvil conectado a la red permitirá probar las funcionalidades de la aplicación móvil. El teléfono IP físico permitirá evidenciar el funcionamiento de la comunicación entre el sistema web y la telefonía IP interna.



**Figura 2.12** Infraestructura del prototipo

## CAPÍTULO 3

### 3. IMPLEMENTACIÓN Y PRUEBAS

#### 3.1 IMPLEMENTACIÓN

##### 3.1.1 HARDWARE

Para la implementación del sistema se cuenta con un servidor IBM x3100 M4 que tiene las especificaciones detalladas en la Tabla 3.1.

**Tabla 3.1** Especificaciones servidor x3100 M4

CARACTERÍSTICA	DETALLE
Modelo	IBM X3100 M4
Tipo	Torre
Procesador	Intel(R) Xeon(R) CPU E31220 @ 3.10 GHz
Procesadores Físicos	1
Procesadores Lógicos	4
Cache	8MB
Memoria	6 GB
Slots de Expansión	4 PCIe Gen 2 slots (x16, x8, x4, x1)
Interfaces de Red	Ethernet Dual Integrado
Disco Duro	1024 GB
Fuente de Poder	350 W fixed

Entre sus características también cuenta con soporte para virtualización mediante hardware y entre los sistemas operativos compatibles se tiene: Red Hat, Suse, Windows Server, VMware vSphere, entre otros.

### 3.1.2 VIRTUALIZACIÓN DE SERVIDORES

Para la implementación del sistema se requiere de un servidor web para el alojamiento de la aplicación principal y un servidor de medios multimedia que será utilizado por el softphone web que se integrará en la aplicación web para proveer los servicios de comunicación con la central telefónica SIP. En el servidor físico IBM x3100 M4 puesto que cuenta con soporte para virtualización y con los recursos necesarios se virtualizarán los siguientes servidores:

- Servidor de medios multimedia
- Servidor web

Para esto se instalará en el servidor físico el sistema operativo VMware Vsphere Hypervisor 5.5 en donde se activará una licencia gratuita, que tiene las características que se muestran a continuación.

- Licencia para 1 CPU Físico
- Ilimitados núcleos por CPU
- Sin fecha de expiración

Para entornos de producción pequeños no presenta mayores limitaciones. La versión de pago permite algunas funcionalidades avanzadas como administración centralizada, migración en caliente de máquinas virtuales, balanceo de carga automático, respaldos, restauración de máquinas virtuales entre otras. Como se puede apreciar las características adicionales de la versión de pago no son necesarias para el sistema por lo que la versión gratuita es suficiente para la implementación.

La implementación de VMware Vsphere Hypervisor 5 en el servidor, la conexión mediante el cliente de Windows para la administración de servidores virtuales y la creación de un servidor virtual se detalla en el Anexo A.

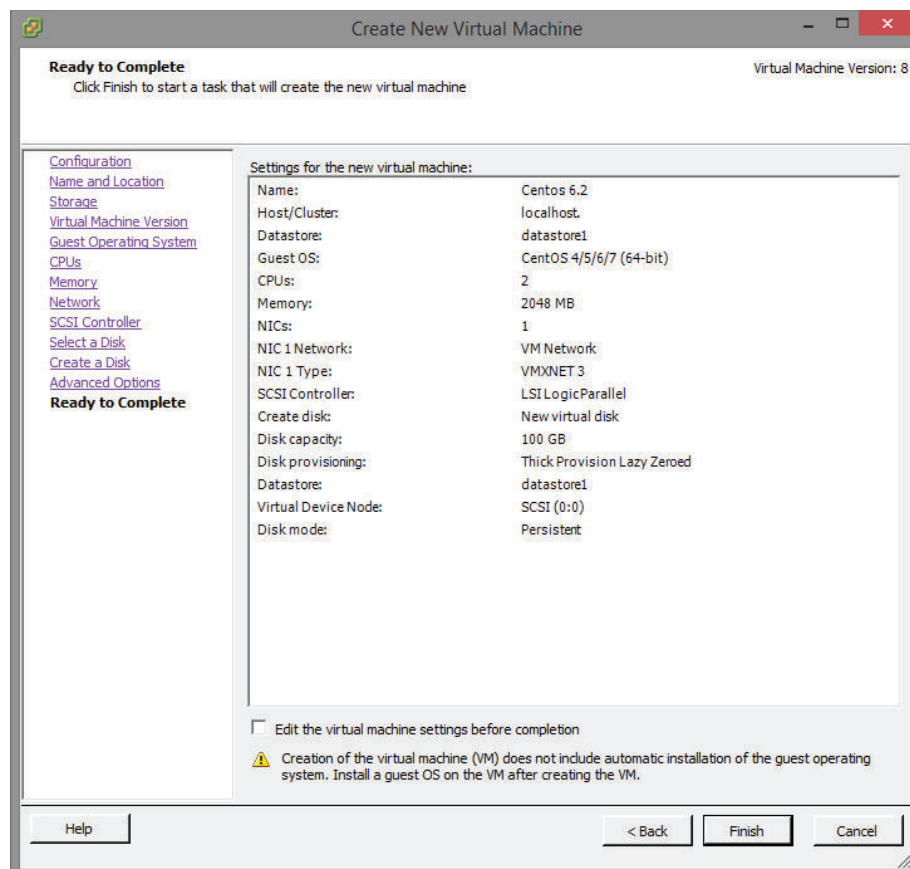
### 3.1.3 IMPLEMENTACIÓN SERVIDOR WEB

Para la implementación del servidor web se deberá crear en VMWare un servidor virtual con las características mostradas en la Tabla 3.2.

**Tabla 3.2** Características servidor web

CARACTERÍSTICA	VALOR
Sistema Operativo	Centos 6.2 64 bits
Número de Núcleos	2
Memoria	2 GB
Disco Duro	100 GB
Interfaces de Red	1

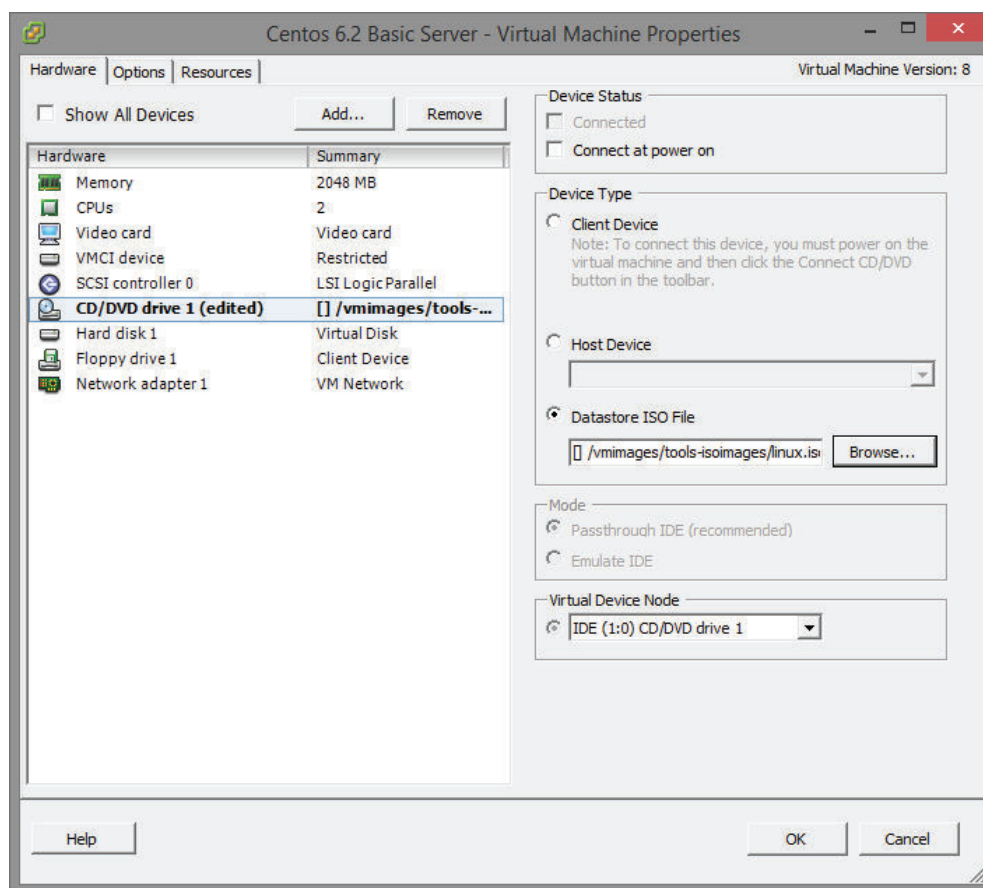
En la Figura 3.1 se muestra un resumen de la configuración para el servidor virtual creado en VMWare.



**Figura 3.1** Resumen configuraciones servidor web

### 3.1.3.1 Instalación Sistema Operativo

Para la instalación del Sistema Operativo se puede utilizar un CD de Instalación o una imagen ISO. Para esto se debe ir a la configuración de *hardware* y en el dispositivo CD/DVD escoger en DataStore ISO File la imagen correspondiente a Centos 6.2 64 bits como se muestra en la Figura 3.2. A continuación se procede a ejecutar la máquina virtual y se mostrará la información inicial de Centos.



**Figura 3.2** Selección ISO para instalación

Se procederá a una instalación típica de Centos 6.2 64 bits utilizando los valores por defecto, se define el nombre del equipo como Soporte, la contraseña de administrador para el usuario *root*, las particiones del disco se dejan en la selección por defecto y en la selección de los paquetes a instalar se debe escoger Basic Server, de esa manera no se instalará herramientas innecesarias como los entornos de escritorio que no se los necesita para el propósito de Servidor Web.

### 3.1.3.2 Implementación Apache

Puesto que en la instalación básica de Centos 6.2 no se instala por defecto Apache se debe proceder a instalarlo. Para realizar esta instalación se debe ejecutar el siguiente comando:

```
yum -y install httpd
```

El servicio instalado corresponde al servicio httpd. Para revisar la versión instalada actualmente ejecutamos el comando `httpd -v` y se obtiene la versión actualmente instalada correspondiente a la 2.2.15.

Para arrancar el servicio de Apache en el servidor es necesario ejecutar el comando `service httpd start`. El servidor arrancará y estará escuchando por defecto en el puerto 80 y 443.

#### 3.1.3.2.1 Host Virtuales

La creación de host virtuales permite alojar varios sitios web con diferentes nombres de dominio en un mismo servidor. Para esto se modifica el archivo `httpd.conf` que se encuentra en el directorio `/etc/httpd/conf/` y se agrega el texto `NameVirtualHost *:80` para definir cualquier IP y que las peticiones serán recibidas por el puerto 80.

Por cada una de las carpetas que se va a alojar se crea un bloque de tipo `VirtualHost` como se muestra en la Figura 3.3, donde el nombre del dominio asociado es `support.lan`.

```
NameVirtualHost *:80

<VirtualHost *:80>
  ServerName support.lan
  DocumentRoot /var/www/html/
</VirtualHost>
```

**Figura 3.3** Configuración host virtual `support.lan`



```
<VirtualHost *:80>
ServerName webim.support.lan
DocumentRoot /var/www/html/webmin
</VirtualHost>
```

**Figura 3.4** Configuración host virtual chat de soporte

La Figura 3.4 muestra la configuración del *virtual host* correspondiente a la aplicación para el chat de soporte, donde el dominio corresponde a `webmin.support.lan`.

Para finalizar la configuración y aplicar los nuevos cambios se debe guardar las modificaciones en el archivo y reiniciar el servicio `httpd` mediante la ejecución en consola del comando `service httpd restart`. Además se deben incluir los dominios asignados a cada *virtual host* en el archivo `hosts` que se encuentra en el directorio `/etc`.

### 3.1.3.3 Implementación PHP

Los repositorios de Centos 6.2 tiene por defecto la versión 5.3.3 de PHP, por lo cual es necesario emplear un repositorio denominado `Webtatic` para su actualización, con la ayuda de este repositorio se puede instalar la versión 5.4 de PHP, la cual es la mínima requerida por la plataforma web.

#### 3.1.3.3.1 Instalación PHP

Para la instalación del repositorio se debe ejecutar en la consola el comando `rpm -Uvh http://mirror.webtatic.com/yum/el6/latest.rpm` que descargará e instalará las referencias del repositorio de `Webtatic`.

Ya con el repositorio disponible para las instalaciones ejecutamos en la consola el comando `yum --enablerepo=webtatic install php54w php54w-mysql php54w-soap php54w-xml php54w-mbstring php54w-cli php54w-common` que se encargará de descargar e instalar PHP y todos los complementos necesarios

para la implementación de la plataforma web. Para finalizar la instalación y aplicar todos los cambios se debe reiniciar el servicio `httpd`.

### 3.1.3.3.2 Configuración PHP

El archivo de configuraciones principales de PHP se encuentra en el directorio `/etc` con el nombre `php.ini`. En este archivo se configurarán las opciones correspondientes al tiempo de ejecución máximo, que permitirá a la aplicación ejecutar procesos durante un mayor lapso de tiempo sin que esta colapse. De igual manera se podrá incrementar la cantidad límite de memoria, la cual puede ocupar la aplicación para la ejecución de tareas más complejas con una mayor carga de memoria. En la Tabla 3.3 se muestran las configuraciones modificadas y el valor con el cual se configuró el archivo.

**Tabla 3.3** Configuraciones `php.ini`

NOMBRE	CONFIGURACIÓN	VALOR
Tiempo Máximo de Ejecución	<code>max_execution_time</code>	600
Límite de Memoria	<code>memory_limit</code>	1024M
Zona Horaria	<code>date.timezone</code>	America\Guayaquil

La modificación de la configuración `date.timezone` permite definir la zona horaria para marcar una referencia para el cálculo de fechas y horas.

### 3.1.3.4 Implementación MySQL

La versión de MySQL necesaria para la implementación debe ser mayor a la 5 lo cual no presenta ningún inconveniente con los repositorios por defecto de Centos 6.2.

#### 3.1.3.4.1 Instalación MySQL

La instalación de MySQL en el servidor debe ser tanto de la aplicación servidor como de la aplicación cliente, para esto se debe ejecutar en la consola el comando

`yum install mysql mysql-server`. Para arrancar el servicio se lo realiza mediante el comando `service mysqld restart` y por defecto va a estar escuchando en el puerto 3306.

#### 3.1.3.4.2 Creación de Bases de Datos

Las bases de datos deben ser creadas para el sistema web principal y para el chat de soporte, esto se lo realiza mediante un gestor de base de datos, se crea una base llamada soporte y la otra webim a cada una de ellas se debe proceder a crear un usuario para asignarles todos los privilegios a cada base de datos respectivamente. La implementación de las tablas se las realizará en otros subcapítulos ya que en el caso del sistema web principal son generadas y mantenidas por el framework y en el caso del chat de soporte se crearán en el momento de la instalación.

### 3.1.4 IMPLEMENTACIÓN SERVIDOR MEDIOS MULTIMEDIA

La implementación de este servidor cumplirá la función de recibirla comunicación realizada por el navegador mediante los protocolos de WebRTC y transmitirlo a la central telefónica mediante el protocolo SIP.

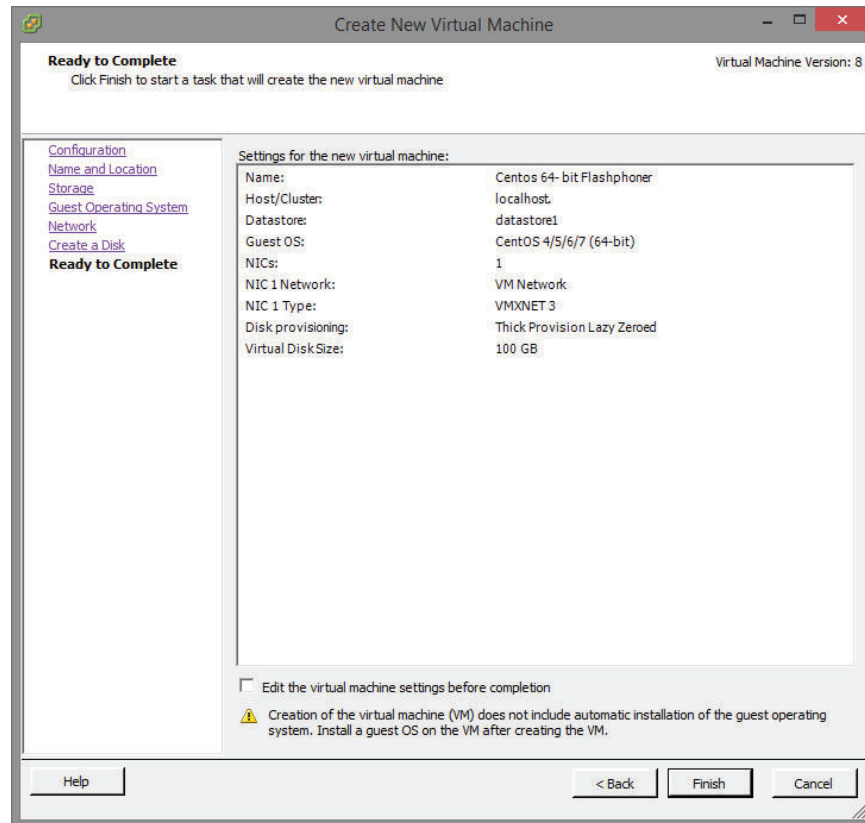
Las características mínimas de funcionamiento para pruebas dadas por el fabricante son de 512 MB de Memoria RAM, 1GB de espacio en disco, 1 core CPU, Sistema Operativo 64 bits y Java Development Kit 1.7.

Para la implementación del Servidor medios multimedia se creará un servidor virtual con las características mostradas en la Tabla 3.4.

**Tabla 3.4** Características servidor medios multimedia

CARACTERÍSTICA	VALOR
Sistema Operativo	Centos 6.4 64 bits
Número de Núcleos	2
Memoria	2 GB
Disco Duro	100 GB
Interfaces de Red	2

En la Figura 3.5 se observa el cuadro de resumen de características de la máquina virtual.



**Figura 3.5** Resumen configuraciones servidor medios multimedia

La instalación del sistema operativo es un proceso similar al realizado para el servidor web ya que de igual forma se utilizará Centos 6.2 64bits. Para la instalación de la misma se utilizarán los valores por defecto y en la selección de paquetes seleccionaremos la opción *Basic Server* de esta manera instalaremos una versión básica de Centos para proceder a instalar sobre este el servidor de Flashphoner Web Call.

#### 3.1.4.1 Instalación Flashphoner Web Call Server

Para empezar la instalación de Flashphoner Web Call Server se debe descargar los archivos correspondientes a la versión 4 que se encuentran en la página

<http://flashphoner.com/web-call-server-4-4/> en la sección Download now y colocarlos en cualquier carpeta del servidor.

A continuación se procede a llenar el formulario que se encuentra la sección Get Trial para obtener una licencia de prueba por 30 días que será enviada al correo electrónico o realizar la compra de alguna licencia del mismo.

Aunque las distribuciones de Linux tienen instalado por defecto *OpenJDK* es recomendación del fabricante del software instalar la última versión de Oracle JDK.

Con los archivos en el servidor se debe proceder de la siguiente manera para instalar en el servidor correctamente:

- Descomprimir archivos mediante el comando `tar -xvzf FlashphonerWebCallServer-4.0.1082.tar.gz`.
- Entrar en el directorio extraído y ejecutar el instalador mediante el comando `./install.sh`.
- Se muestra en la consola el proceso de instalación, al terminar este proceso indicará que la instalación fue satisfactoria.

#### **3.1.4.2 Activación Licencia Flashphoner Server**

Para activar la licencia que fue enviada por correo se debe ir al directorio `/usr/local/FlashphonerWebCallServer/bin/` y ejecutar el comando `./activation.sh` el cual pedirá por consola que se ingrese la licencia obtenida anteriormente y finalizará la activación mostrando un mensaje `Activation Complete` como se muestra en la Figura 3.6.

```

Flashphoner License Activation
-----
Please, enter your License Number: XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXX
11F3E587-C406-4911-AD06-9062C574D5F9
activation date: 2014.03.06
expiration date: 2014.04.04
name: Jonathan Dutã;n
company: Personal
product name: Web Call Server 3
features: [transfer, audio, g729, webrtc, chat, dtmf, video]
licenseNumber: 11F3E587-C406-4911-AD06-9062C574D5F9
licenseType: Trial
hardwareId: EC8298A0A0132967F1C6CE284F18ADC6E710A0DFAA1A98D90AAGF377AF88A7C94F66CF07F8FBD04D70A7903C4957C4430767EA4F
A981A5ACE1FE9C07529183E
support: This license does not include any support plan

activation complete

```

**Figura 3.6** Activación licencia Flashphoner Server

### 3.1.4.3 Ejecución Flashphoner Web Call Server

Para gestionar el servicio Flashphoner Web Call Server se puede realizar mediante el comando `service webcallserver start|stop|restart|status` que permite 4 diferentes opciones:

- `start` para inicializar el servicio.
- `stop` para detener el servicio.
- `restart` para reinicializar el servicio.
- `status` para verificar el estado actual del servicio.

Para configurar el inicio automático del servicio se lo puede activar mediante la ejecución del comando `chkconfig webcallserver on`.

### 3.1.5 IMPLEMENTACIÓN SISTEMA WEB INTEGRAL

Para la implementación del sistema se procede a inicializar el desarrollo mediante la creación de un proyecto nuevo basado en un *framework* MVC que permitirá el uso funciones definidas en el sistema base que ayudarán en la conexión con la base de datos, la implementación de las vistas, el enrutamiento, la depuración de la aplicación, el registro de actividades, entre otras. Para esto se ha elegido a uno de los *frameworks* de PHP denominado Symfony2.

### 3.1.5.1 Implementación Symfony2

Symfony2 se debe instalar sobre la versión de PHP 5.3 o superior, los componentes o *bundles* principales incluidos en la versión inicial corresponden a los que se muestran en la Tabla 3.5.

**Tabla 3.5** Componentes principales Symfony2

COMPONENTE	DETALLE
Config	Gestiona las configuraciones para la aplicación
Console	Permite la creación de interfaces con líneas de comando
Debug	Permite realizar la depuración de código PHP
Filesystem	Provee utilidades para el sistema de archivos
Form	Conjunto de herramientas para la creación de formularios HTML
Security	Provee funciones para autorizaciones en el sistema
Serializar	Transforma los objetos en XML, JSON, Yaml y otros
Templating	Provee herramientas para construir un sistema de plantillas

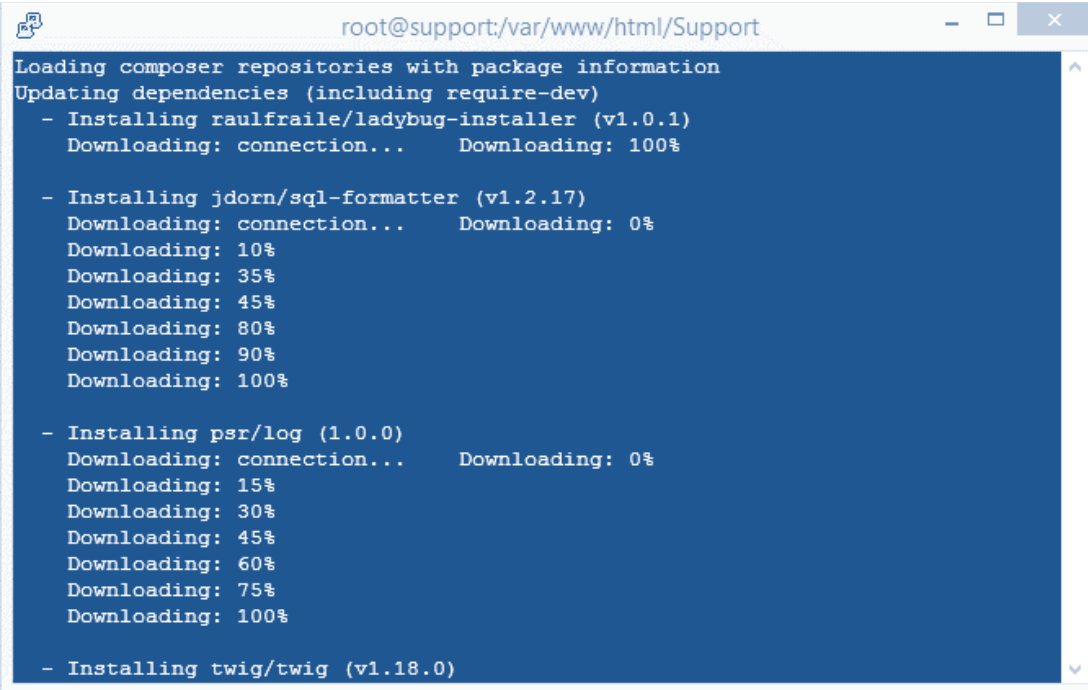
#### 3.1.5.1.1 Instalación Symfony2

Para realizar la instalación de Symfony2 primero se escogerá una versión adecuada para el proyecto, para este caso se utilizará la versión 2.3 LTS(Long Terms Support) ya que tiene un tiempo extendido de soporte. A continuación se necesita descargar el gestor de dependencias Composer en la carpeta `/var/www/html` que corresponde al directorio web por defecto de apache, para esto se ejecuta el comando `curl -sS https://getcomposer.org/installer | php`.

Para empezar la creación del proyecto que será llamado Support y descargar la versión de Symfony escogida se procede a ejecutar el siguiente comando:

```
php composer.phar create-project symfony/framework-standard-edition
Support/ 2.3
```

El directorio `Support` corresponde a la carpeta base de la plataforma web. Al ejecutar el comando mostrará el proceso de descarga de todos los *bundles* o componentes de la instalación básica de Symfony. En la Figura 3.7 se muestra un fragmento del proceso de descarga de componentes.



```
root@support:/var/www/html/Support
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing raulfraile/ladybug-installer (v1.0.1)
  Downloading: connection...   Downloading: 100%

- Installing jdorn/sql-formatter (v1.2.17)
  Downloading: connection...   Downloading: 0%
  Downloading: 10%
  Downloading: 35%
  Downloading: 45%
  Downloading: 80%
  Downloading: 90%
  Downloading: 100%

- Installing psr/log (1.0.0)
  Downloading: connection...   Downloading: 0%
  Downloading: 15%
  Downloading: 30%
  Downloading: 45%
  Downloading: 60%
  Downloading: 75%
  Downloading: 100%

- Installing twig/twig (v1.18.0)
```

**Figura 3.7** Descarga de componentes Symfony

Al culminar el proceso de instalación del *framework* se configura la información de la base de datos creada en la implementación de MySQL, para esto se debe llenar en el archivo de configuración `parameters.yml` que se encuentra en la carpeta `app/config` de la aplicación como se muestra en la Figura 3.8.

```
parameters:
  database_driver: pdo_mysql
  database_host: localhost
  database_port: 3306
  database_name: support
  database_user: support
  database_password:
```

**Figura 3.8** Archivo `parameters.yml`



### 3.1.5.1.2 Instalación Bundles Adicionales

Los componentes o *bundles* permiten al *framework* añadir funcionalidades y complementos adicionales a las que ya se encuentran por defecto en Symfony2. Para agregar nuevos *bundles* en el sistema se debe modificar el archivo `composer.json` y adicionar las líneas correspondientes a cada *bundle* junto con la versión con la cual se quiere instalar. En la Figura 3.9 se muestra el fragmento principal del archivo `composer.json` de la aplicación Support.

```

{
  "name": "symfony/framework-standard-edition",
  "license": "MIT",
  "type": "project",
  "description": "The \"Symfony Standard Edition\" distribution",
  "autoload": {
    "psr-0": { "": "src/" }
  },
  "require": {
    "php": ">=5.3.3",
    "symfony/symfony": "2.3.*",
    "doctrine/orm": "~2.2,>=2.2.3",
    "doctrine/doctrine-bundle": "1.2.*",
    "twig/extensions": "1.0.*",
    "symfony/assetic-bundle": "2.3.*",
    "symfony/swiftmailer-bundle": "2.3.*",
    "symfony/monolog-bundle": "2.3.*",
    "sensio/distribution-bundle": "2.3.*",
    "sensio/framework-extra-bundle": "2.3.*",
    "sensio/generator-bundle": "2.3.*",
    "incenteev/composer-parameter-handler": "~2.0",
    "jms/serializer-bundle": "~0.13",
    "friendsofsymfony/rest-bundle": "dev-master",
    "braincrafted/bootstrap-bundle": "~2.0",
    "twbs/bootstrap": "dev-master",
    "jquery/jquery": "1.10.*",
    "mopa/bootstrap-bundle": "dev-master",
    "raulfraile/ladybug-bundle": "~1.0",
    "laupfprpar/pusher-bundle": "dev-master",
    "parse/php-sdk": "1.0.*"
  }
}

```

Figura 3.9 Archivo `composer.json`

Después se ejecuta el comando `php composer.phar update` que descargará las dependencias agregadas y actualizará los *bundles* actuales del sistema. Para activar el *bundle* es necesario registrarlo en el archivo `AppKernel.php` que se encuentra en el directorio `app` de la aplicación.

A continuación se listarán los principales *bundles* que se agregaron a Symfony con el respectivo motivo de su instalación.

#### **BraincraftedBootstrapBundle**

Permite la integración entre Twitter Bootstrap y Symfony2. Twitter Bootstrap es un *framework front-end* para fácil y rápido desarrollo web. De esta manera se puede realizar un desarrollo fácil utilizando componentes de *front-end* prediseñados.

### ***FOSRestBundle***

Este *bundle* provee varias herramientas para la creación de APIs de tipo REST(Representational State Transfer) para aplicaciones Symfony2. Entre algunas de las características importantes que aporta este *bundle* son un cargador de rutas para generar URLs (Uniform Resource Locator) siguiendo la convención REST y una capa de vista para habilitar la salida en diferentes formatos.

### ***SerializerBundle***

Integra la librería Serializer con Symfony2. Serializer es una librería que permite serializar y deserializar fácilmente información, actualmente soporta formatos XML, JSON y YAML. Además soporta una gran cantidad de tipos PHP como las fechas.

### ***PusherBundle***

Integra Symfony con el SDK(Software Development Kit) de Pusher, facilitando la configuración y acoplamiento del servicio con cualquier aplicación basada en Symfony.

### ***LadyBugBundle***

Facilita la depuración de código en las aplicaciones basadas en Symfony.

#### **3.1.5.2 Creación de Entidades**

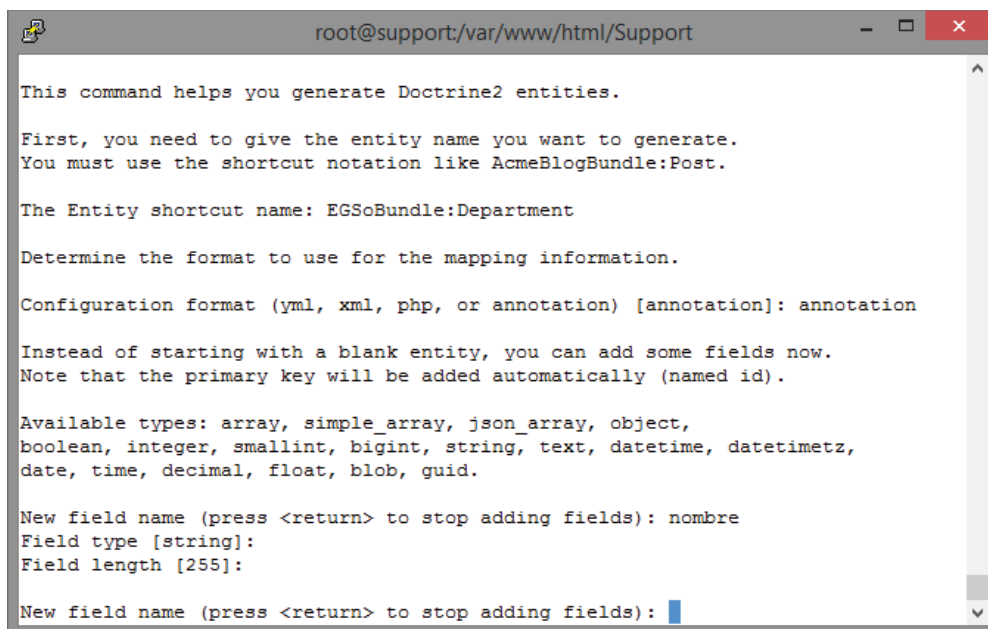
Las entidades del sistema que se implementaron corresponden a las que se muestran en la Figura 2.6, correspondiente al diagrama de clases. Para la creación

de entidades Symfony provee herramientas para el CLI (Command Line Interface) que facilitan la implementación de las mismas.

Esto se realiza mediante el siguiente comando:

```
php app/console doctrine:generate:entity
```

Este comando ejecutará en consola un asistente para la creación de una nueva entidad en donde se agrega información sobre cada uno de los atributos correspondientes a la entidad. En la Figura 3.10 se visualiza la creación de la entidad departamento para mostrar un ejemplo sencillo de la creación de todas las entidades.



```
root@support:/var/www/html/Support

This command helps you generate Doctrine2 entities.

First, you need to give the entity name you want to generate.
You must use the shortcut notation like AcmeBlogBundle:Post.

The Entity shortcut name: EGSoBundle:Department

Determine the format to use for the mapping information.

Configuration format (yaml, xml, php, or annotation) [annotation]: annotation

Instead of starting with a blank entity, you can add some fields now.
Note that the primary key will be added automatically (named id).

Available types: array, simple_array, json_array, object,
boolean, integer, smallint, bigint, string, text, datetime, datetimetz,
date, time, decimal, float, blob, guid.

New field name (press <return> to stop adding fields): nombre
Field type [string]:
Field length [255]:

New field name (press <return> to stop adding fields):
```

**Figura 3.10** Creación entidad Departamento

Las entidades son clases que por defecto crean todos sus atributos como privados, por lo cual es necesario crear los *getters* y *setters* de cada entidad. Al generarla mediante el asistente, estos se crean automáticamente, pero en el caso de que exista alguna modificación de la entidad se puede ejecutar el siguiente comando:

```
php app/console doctrine:generate:entities
EG/SoBundle/Entity/[Nombre de la Entidad]
```

De esta manera se crearán los *getters* y *setters* que sean necesarios, solo se agregarán las funciones para los atributos nuevos. En caso de requerir actualización de atributos se borra las funciones asociadas a cada uno y se ejecuta el comando nuevamente.

En caso de requerir atributos que asocien con otras entidades se los debe crear manualmente y además en caso de ser una relación múltiple se los debe inicializar como `ArrayCollection` en el constructor. De esta manera al momento de crear los *getters* y *setters* se crearán funciones específicas para adicionar o quitar objetos a los `ArrayCollections` correspondientes a las entidades relacionadas.

### 3.1.5.3 Implementación de Base de Datos

Para la implementación y persistencia con la base de datos, Symfony se integra con un conocido ORM denominado Doctrine, para lo cual al momento de crear las entidades también se incorporan en los atributos y en la clase anotaciones con información para el ORM. Esta información corresponde al nombre de la tabla, el nombre de los atributos, el tipo de los atributos, algunas características adicionales de cada atributo, las relaciones entre las entidades en caso de tenerlas, y mucha más información que ayuda al ORM a realizar la persistencia con la base de datos. Para la creación de una entidad es obligatorio incluir un atributo de tipo `id` con un valor autoincremental único. Las anotaciones más comunes utilizadas en las entidades se muestran en la Tabla 3.6.

**Tabla 3.6** Anotaciones ORM para entidades

ANOTACIÓN	DESCRIPCIÓN
<code>@ORM\Entity()</code>	Identifica una entidad y en caso de tener un repositorio adicional se pasa como parámetro
<code>@ORM\Table()</code>	Información sobre la tabla asociada a la Entidad se puede pasar como parámetro el nombre de la tabla

@ORM\Column()	Permite definir las características de las columnas asociadas a los atributos de la entidad mediante la información del nombre y características asociadas a cada columna.
@ORM\OneToOne()	Relación uno a uno con otra entidad se debe pasar como parámetro el nombre de la entidad
@ORM\OneToMany()	Relación uno a muchos con otra entidad se debe pasar como parámetro el nombre de la entidad
@ORM\ManyToMany()	Relación muchos a muchos con otra entidad se debe pasar como parámetro el nombre de la entidad y el nombre de la tabla intermedia

En la Figura 3.11 se muestra un ejemplo de una entidad creada mediante el asistente y modificada manualmente para agregar las relaciones entre entidades.

Después de crear todas las entidades involucradas en el sistema según el diagrama UML del diseño de la aplicación de la Figura 2.6, se procede a la creación de las tablas y esquema de la base de datos. Para realizar este proceso se ejecuta el siguiente comando en la consola:

```
php app/console doctrine:schema:update --force
```

En caso de que se modifique la información de las anotaciones o se incorporen nuevos atributos en las entidades, se ejecuta el mismo comando y este procederá a realizar las actualizaciones correspondientes en la base de datos.

```

namespace EG\SoBundle\Entity;

use Doctrine\ORM\Mapping as ORM;

/**
 * Departamento
 *
 * @ORM\Table("departamento")
 * @ORM\Entity
 */
class Departamento
{
    /**
     * @var integer
     *
     * @ORM\Column(name="id", type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    private $id;

    /**
     * @var string
     *
     * @ORM\Column(name="nombre", type="string", length=255, nullable=false,
     unique=true)
     */
    private $nombre;

    /**
     * @var string
     *
     * @ORM\Column(name="observacion", type="string", length=255, nullable=true)
     */
    private $observacion;

    /**
     * @ORMOneToMany(targetEntity="User", mappedBy="departamento")
     */
    private $users;
}

```

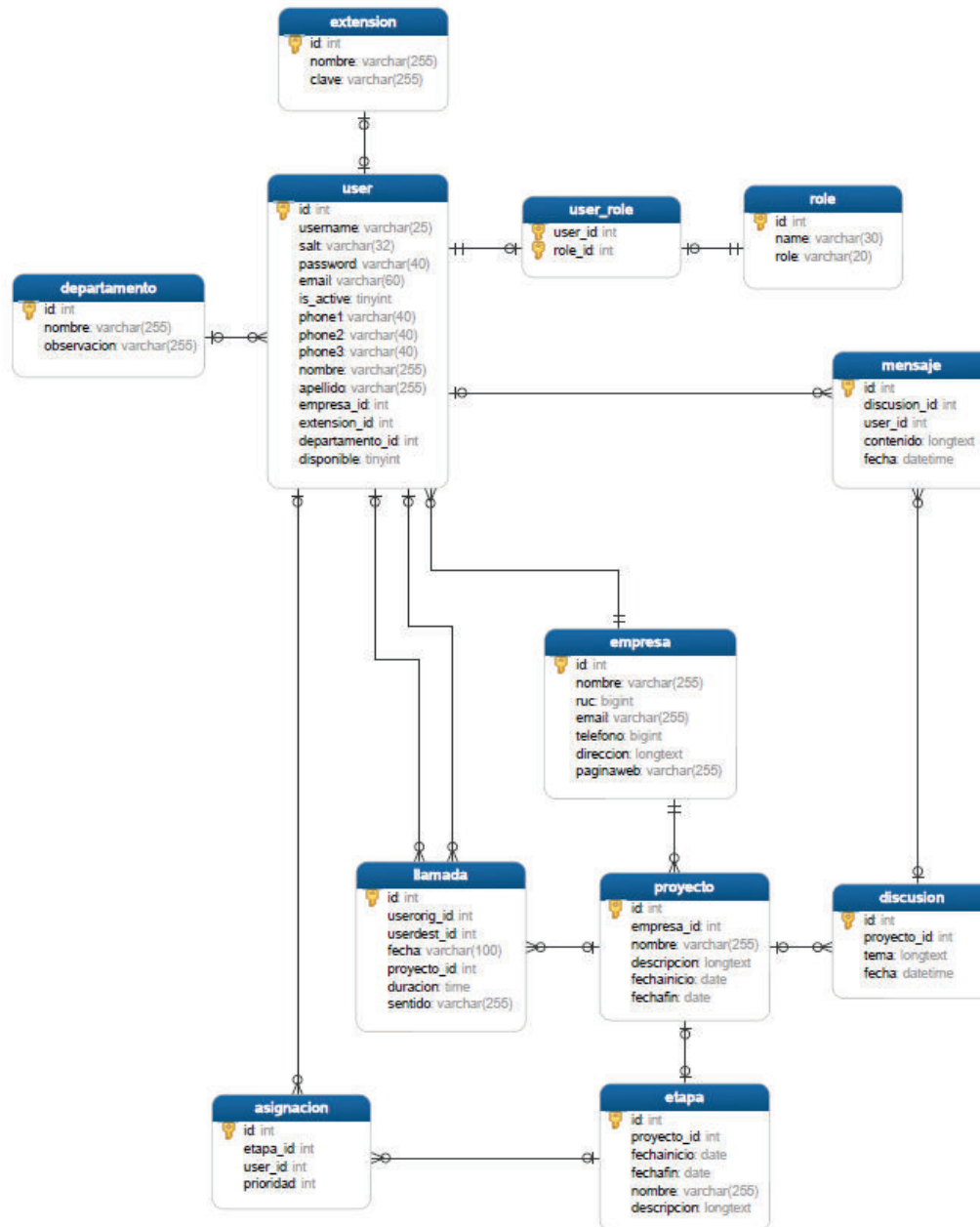
**Figura 3.11** Ejemplo entidad creada

En la Figura 3.12 se visualizan las tablas creadas por el ORM como también las relaciones entre ellas.

Mediante la implementación de las entidades el acceso a las tablas de la base de datos es muy sencilla ya que todo pasa a ser manejado directamente como objetos y Doctrine se encarga de la persistencia.

En el caso de que se requieran consultas complejas o que el ORM no implementa por defecto se las realiza mediante el *Query Builder*, que es parte del DBAL (Database Abstraction Layer) de Doctrine y utiliza un lenguaje propio denominado DQL (Doctrine Query Language) que basa su sintaxis en el SQL convencional pero es orientado a objetos. De esta manera podemos trabajar en un alto nivel de la

abstracción de base de datos, de forma que las consultas no involucren tablas solo objetos.



**Figura 3.12** Diagrama de tablas creadas por el ORM

Para agregar este tipo de consultas y poderlas utilizar en cualquier momento se las creará en el repositorio de cada entidad. En la Figura 3.13 se puede observar un ejemplo del repositorio de la entidad Asignacion.

```

namespace EG\SoBundle\Entity;

use Doctrine\ORM\EntityRepository;

/**
 * AsignacionRepository
 *
 * This class was generated by the Doctrine ORM. Add your own custom
 * repository methods below.
 */
class AsignacionRepository extends EntityRepository
{
    public function getProyectByNombre($arg)
    {
        return $this->getEntityManager()
            ->createQuery(
                "SELECT p FROM EGSoBundle:Proyecto p WHERE p.nombre LIKE '%" . $arg . "%'"
            )
            ->getArrayResult();
    }
}

```

**Figura 3.13** Ejemplo repositorio

### 3.1.5.4 Seguridad

Para aumentar la seguridad de una aplicación web, Symfony provee una capa de seguridad la cual mediante la asignación de usuarios y roles permite configurar los mecanismos para realizar autenticación y autorización a las diferentes secciones de una aplicación.

#### 3.1.5.4.1 Usuarios y Roles

Para la creación de usuarios en el sistema, se utiliza la entidad User que implementa de una interface de Symfony para usuarios denominada AdvancedUserInterface, tiene como atributos principales: id(Identificador), username(Nombre de usuario), password (Contraseña), y isActive(Activo). Estos atributos son empleados por Symfony para la capa de seguridad.

La creación de roles se la realiza mediante la entidad Role que para ser considerada en la capa de seguridad provista por Symfony, debe implementar una interface denominada RoleInterface, esta entidad tiene como atributos principales: id(Identificador), name(Nombre), role (Rol). El atributo role es único y debe ser una cadena de caracteres con el prefijo ROLE\_ como por ejemplo ROLE\_ADMIN o ROLE\_USER.



Para la asignación de roles a cada uno de los usuarios se generó una relación muchos a muchos bidireccional entre estas entidades.

#### 3.1.5.4.2 Autenticación

Para permitir la autenticación mediante los usuarios almacenados en la base de datos es necesario realizar cambios en el archivo `security.yml` que se encuentra en el directorio `app/config`. En la Figura 3.14 se muestra el archivo `security.yml` usado para el sistema.

```
security:
  encoders:
    EG\SoBundle\Entity\User:
      algorithm: sha1
      encode_as_base64: false
      iterations: 1
  role_hierarchy:
    ROLE_ADMIN: ROLE_USER

  providers:
    administrators:
      entity: { class: EGSoBundle:User, property: username }
  firewalls:
    wsse_secured:
      pattern: /rest/.+
      stateless: true
      wsse: true
    admin_area:
      pattern: ^/admin
      http_basic: ~
    dev:
      pattern: ^/(_(profiler|wdt)|css|images|js)/
      security: false

    login:
      pattern: ^/login$
      security: false

    secured_area:
      pattern: ^/
      anonymous: ~
      form_login:
        check_path: login_check
        login_path: login
        default_target_path: /
      logout:
        path: /logout
        target: /login
      remember_me:
        key: "%secret%"
        lifetime: 3600 # 365 days in seconds
        path: /
        domain: ~ # Defaults to the current domain from $_SERVER
        remember_me_parameter: remember_me
```

**Figura 3.14** Archivo `security.yml`

En el archivo `security.yml` se realiza las siguientes configuraciones:

**encoders:** Se configura el algoritmo de codificación que se utilizara para la contraseña de la entidad User. Para este caso se utilizó SHA1, sin codificación en Base64 y una iteración.

**role\_hierarchy:** Se define la jerarquía entre roles, es decir según lo que se muestra en la Figura 3.14 que un usuario que tenga un `ROLE_ADMIN` tiene todos los permisos de un usuario `ROLE_USER`.

**providers:** Se define la fuente que se va a utilizar para validar los usuarios, en este caso la entidad User.

**firewalls:** Se define las diferentes secciones de la aplicación y como se va a aplicar la seguridad según el patrón de la URL. En la Figura 3.14 se puede identificar que existe una sección `secured_area`, que indica que las peticiones dirigidas a cualquier URL requiere autenticación, pero además se define una sección `login` habilitando el acceso a la URL `/login`, la cual dirige al formulario de acceso que se presenta en la Figura 3.15.



**Figura 3.15** Formulario acceso

### 3.1.5.4.3 Autorización

En Symfony la autorización se la realiza mediante los roles asignados a cada usuario, mediante la sección `access_control` del archivo `security.yml` que se muestra en la Figura 3.16 o utilizando cualquier otro atributo del usuario en cualquier sección del código.

```
access_control:
- { path: ^/admin, roles: ROLE_ADMIN }
- { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
- { path: ^/rest, roles: IS_AUTHENTICATED_ANONYMOUSLY }
- { path: ^/, roles: IS_AUTHENTICATED_FULLY }
```

**Figura 3.16** Control de acceso

Los roles existentes en el sistema son los siguientes: `ROLE_ADMIN`, `ROLE_USER`, `ROLE_JEFE` y `ROLE_CLIENTE`, que se utilizarán para gestionar permisos en la aplicación.

Los permisos de la plataforma están organizados según el tipo de rol que tenga asignado, y se describen a continuación:

#### ***ROLE\_CLIENTE***

- Acceso al perfil del Usuario
- Realizar llamadas a usuarios asignados a su proyecto
- Chatear con usuarios de soporte

#### ***ROLE\_USER***

- Acceso a perfil de Usuario
- Llamadas a otros usuarios
- Recibir llamadas de clientes y usuarios
- Chatear con usuarios conectados a la plataforma

- Acceso a la interfaz de atención a clientes del chat de soporte
- Llamadas a la telefonía pública
- Ver proyectos en los que está asignado

### ***ROLE\_JEFE***

- Acceso a perfil de Usuario
- Llamadas a otros usuarios
- Recibir llamadas de clientes y usuarios
- Chatear con usuarios conectados a la plataforma
- Acceso a la interfaz de atención a clientes del chat de soporte
- Llamadas a la telefonía pública
- Administrar y Gestionar Proyectos
- Administrar Etapas
- Asignar Usuarios a Etapas

### ***ROLE\_ADMIN***

- Acceso a perfil de Usuario
- Llamadas a otros usuarios
- Recibir llamadas de clientes y usuarios
- Chatear con usuarios conectados a la plataforma
- Acceso a la interfaz de atención a clientes del chat de soporte
- Llamadas a la telefonía pública
- Administrar y Gestionar Proyectos
- Administrar Etapas
- Asignar Usuarios a Etapas
- Administrar Usuarios y Roles
- Configurar información de la central telefónica

### 3.1.5.5 Enrutamiento

Las rutas permiten dar un direccionamiento a los métodos de los controladores que ejecutan alguna acción y en caso de ser necesario muestran las respuestas mediante una vista.

Para la gestión de rutas mediante Symfony se utiliza el archivo `routing.yml` que se encuentra en el directorio `src/EG/SoBundle/Resources/config`, aquí se agregó la primera ruta denominada anotaciones ubicada en las líneas superiores de la Figura 3.17, que permiten crear rutas mediante anotaciones en cada método de los controladores como se indica en la Figura 3.18. En la Figura 3.17 también se presentan algunas rutas creadas directamente en el archivo `routing.yml`, como son el caso de la ruta denominada `index` y `login` que incluyen en el atributo `defaults` el método y controlador al cual se redireccionan.

```

anotaciones:
  resource: "@EGSoBundle/Controller"
  type:     annotation
index:
  pattern:  /
  defaults: { _controller: EGSoBundle:Default:index }
login:
  pattern:  /login
  defaults: { _controller: EGSoBundle:Security:login }
login_check:
  pattern:  /login_check
logout:
  path:    /logout

```

Figura 3.17 Archivo `routing.yml`

```

/**
 * Perfil de usuario
 * @Route("/perfilUsuario/{id}", name="perfilUsuario")
 * @Method("GET")
 */
public function perfilUsuarioAction($id)
{
    $em = $this->getDoctrine()->getManager();
    $entity = $em->getRepository('EGSoBundle:User')->find($id);
    return $this->render('EGSoBundle:User:userPerfil.html.twig', array('usuario'
=> $entity));
}

```

Figura 3.18 Enrutamiento con anotaciones

### 3.1.5.6 Controladores

En los controladores se encuentran todas las acciones que se ejecutarán cuando un cliente direcciona a una ruta específica, es común que exista un controlador por cada entidad. Las clases controladores tienen una notación que siempre termina con la palabra `Controller` y los métodos que están asociados a alguna ruta deben terminar con la palabra `Action` para que sean reconocidos por el sistema.

Para evidenciar el funcionamiento común de un controlador, se explicará la ejecución del controlador denominado `UserController` que parte de su código se muestra en la Figura 3.19.

```
class UserController extends Controller
{
    /**
     * Perfil de usuario
     * @Route("/perfilUsuario/{id}", name="perfilUsuario")
     * @Method("GET")
     */
    public function perfilUsuarioAction($id)
    {
        $em = $this->getDoctrine()->getManager();
        $entity = $em->getRepository('EGSoBundle:User')->find($id);
        return $this->render('EGSoBundle:User:userPerfil.html.twig', array('usuario'
=> $entity));
    }
}
```

**Figura 3.19** Código UserController

La clase `UserController` para ser un controlador de Symfony debe extender de la clase base `Controller`, el método `PerfilUsuarioAction` que tiene un enrutamiento basado en anotación, acepta peticiones mediante el método `GET` donde el valor `{id}` de la URL `/perfilUsuario/{id}` corresponde al valor `$id` que el método necesita como parámetro de entrada, es decir una URL `/perfilUsuario/1` pasaría al método el parámetro `1`.

En la ejecución del mismo se muestra que realiza una petición al repositorio solicitando una entidad de tipo `User` basada en el parámetro de entrada `$id`. Esta

información es enviada a la vista mediante la función `render` donde se especifica el archivo que se utilizará en este caso `userPerfil.html.twig`, el cual devolverá una respuesta la misma que se enviará al cliente. Este es el funcionamiento más básico y común de los controladores.

### 3.1.5.7 Vistas

Para mostrar la respuestas a las peticiones realizadas por el cliente en Symfony se utiliza twig para lo cual se han creado plantillas que faciliten la visualización de los contenidos. En el sistema existen 2 tipos de plantillas base la plantilla para operadores y la plantilla de clientes.

La plantilla de operadores se denomina `layout.html.twig` que se encuentra en el directorio `src/EG/SoBundle/views`, esta plantilla cuenta con las etiquetas HTML y componentes básicos de CSS y Javascript que se utilizarán en todas las vistas asociadas a los operadores, además se definen los bloques que serán empleados en las vistas que extiendan de esta plantilla para modificarlos o agregarles código particular de cada vista. En esta también se incluye el menú de navegación en el bloque encabezado para que el mismo se pueda reutilizar en todas las vistas, considerando los permisos de los usuarios según sus respectivos roles.

La plantilla de clientes se denomina `layout_cliente.html.twig`, se encuentra en el directorio `src/EG/SoBundle/views`, esta de igual forma que la de operador cumple similares funciones pero con los componentes y permisos propios de un usuario cliente, aunque en estructura el menú y algunas funciones son similares a la de operadores, se ha restringido para solo mostrar el perfil del cliente.

### 3.1.5.8 Implementación Chat en Tiempo Real mediante Websockets

Esta implementación es basada en un chat de código abierto denominada Pusher Chat – JQuery Plugin creada y compartida en Github por el usuario *html5-ninja*. La

adaptación realizada consiste en integrar la aplicación base con Symfony y realizar los cambios necesarios para integrarlo en la interfaz del sistema web, para que pueda ser utilizado como chat interno entre los usuarios de la empresa tipo.

Para esta implementación es necesario crear una cuenta en <http://pusher.com>, en este caso se creó una cuenta gratuita que incluye 100000 mensajes y 20 conexiones simultáneas. En la cuenta de Pusher se crea una nueva aplicación que se llamará *Chat* y Pusher proveerá un identificador de la aplicación y los *tokens* de acceso para que el sistema web se pueda comunicar con el servidor de Pusher.

#### 3.1.5.8.1 Integración con Symfony

Para integrarlo en el sistema web en el lado del Servidor se utiliza un complemento de Symfony denominado `pusher-bundle`, se agrega la información de configuración provista por Pusher en el archivo `config.yml` como se muestra en la Figura 3.20.

```
lopi_pusher:
  app_id: "106939"
  key: '4018c083befca466a14c'
  secret:

  # Default configuration
  debug: false # true if you want use the debug of all requests
  host: http://api.pusherapp.com
  port: 80
  timeout: 30

  # Optional configuration
  auth_service_id: pusher.authentication.service # optional if you want to use
  private or presence channels
```

**Figura 3.20** Configuración parámetros Pusher

Se crea un método `pusherServerAction` en el controlador `Default` que se utilizará para comunicarse con el servidor de Pusher como se muestra en la Figura 3.21. Este se encarga de recibir la información de los mensajes y enviarlo al servidor Pusher para que los reenvíe al destinatario.



```

/**
 * Creando Pusher Server
 *
 * @Route("/pusher_server", name="pusher_server")
 */
public function pusherServerAction(Request $petition){

    $pusher = $this->container->get('lopi_pusher.pusher');

    if ($petition->get('typing') == "false"){
        $pusher->trigger('presence-mychannel', 'send-event', array('message' =>
htmlspecialchars ($petition->get('msg')), 'from' => $petition->get('from'), 'to'
=> str_replace('#', '', $petition->get('to'))));
    }
    else if ($petition->get('typing') == "true")
        $pusher->trigger('presence-mychannel', 'typing-event', array('message'
=> $petition->get('typing'), 'from' => $petition->get('from'), 'to' => str_replace
('#', '', $_POST['to'])));
    else{
        $pusher->trigger('presence-mychannel', 'typing-event', array('message'
=> 'null', 'from' => $petition->get('from'), 'to' => str_replace('#', '',
$petition->get('to')));
    }
    return new Response('');
}

```

**Figura 3.21** Método pusherServerAction

Para la autenticación de los usuarios en Pusher también se creó un método adicional en el controlador Default denominado pusherAuthPathAction que se encarga de autenticar al usuario mediante el identificador del usuario que haga la petición.

#### 3.1.5.8.2 Integración con la Interfaz de Usuario

En el lado del cliente se agregó una función en el código del chat en el archivo *jquery.pusherchat.js* que permita la conexión y desconexión, esto se realizó con el objetivo de que el usuario pueda controlar el estado de su conexión mediante la interfaz. Se inicializó el chat en la plantilla del operador de forma que en caso de estar disponible el chat se muestre en la parte inferior de todas las vistas del operador, como se muestra en la Figura 3.22.

The screenshot displays a web application interface for user management and project tracking. At the top, there is a navigation bar with tabs for 'Soporte', 'Departamentos', 'Proyectos', 'Empresas', 'Llamadas', 'Mensajes', and 'Sistema', along with a user profile for 'Eduardo Gallegos'. The main content area is divided into two sections: 'Perfil Actual' and 'Proyectos de admin'. The 'Perfil Actual' section lists user details such as 'Usuario: admin', 'Nombre: Eduardo', 'Apellido: Gallegos', and contact information. The 'Proyectos de admin' section features a table with columns for 'Proyecto', 'Empresa', and 'Acción'. A search bar and pagination controls are also present. On the right side, there is a yellow sidebar titled '0 En Línea' listing online users.

**Figura 3.22** Chat mediante Pusher para usuarios

### 3.1.5.9 Implementación softphone web

Para la implementación del softphone web es necesario contar con Flashphoner Web Call Server instalado y ejecutándose como se realizó en la implementación del Servidor de Medios Multimedia.

#### 3.1.5.9.1 Marcación por teclado

Las llamadas mediante teclado son permitidas solo para los usuarios operadores y administradores de la plataforma. Para la implementación de esta funcionalidad se creó una ruta `/llamadas/teclado` direccionada al método `tecladoAction` del controlador `LlamadaController` que se encarga de retornar la información de la vista `teclado.html.twig` en donde se realiza toda la lógica de comunicación con Flashphoner Web Call Server.

En la vista se deben incluir algunas dependencias e incluir al API de Javascript provisto por Flashphoner para la comunicación con el servidor multimedia. A

continuación se muestra en la Figura 3.23 los archivos utilizados para la implementación.

```
{% block javascriptfiles %}
  <script type="text/javascript" src="{{ asset('flashphoner/dependencias/jquery/jquery.js') }}"></script>
  <script type="text/javascript" src="{{ asset('flashphoner/dependencias/jquery/jquery-ui.js') }}"></script>
  <script type="text/javascript" src="{{ asset('flashphoner/dependencias/jquery/jquery.websocket.js') }}"></script>
  <script type="text/javascript" src="{{ asset('flashphoner/dependencias/jquery/jquery.json.js') }}"></script>
  <script src="{{ asset('flashphoner/gui/js/jquery.mCustomScrollbar.concat.min.js') }}"></script>

  <script type="text/javascript" src="{{ asset('flashphoner/Flashphoner.js') }}"></script>
  <script type="text/javascript" src="{{ asset('flashphoner/dependencias/swf/swfobject.js') }}"></script>

  <script type="text/javascript" src="{{ asset('flashphoner/Phone-simple.js') }}"></script>
  <script type="text/javascript" src="{{ asset('flashphoner/Phone.js') }}"></script>

  <script type="text/javascript" src="{{ asset('flashphoner/SoundControl.js') }}"></script>
  <script type="text/javascript" src="{{ asset('flashphoner/listener/DefaultListener.js') }}"></script>
  <script type="text/javascript" src="{{ asset('flashphoner/listener/LoadToolListener.js') }}"></script>
  <script type="text/javascript" src="{{ asset('flashphoner/ConfigurationLoader.js') }}"></script>
{% endblock %}
```

**Figura 3.23** Encabezado archivo teclado.html.twig

A continuación en la Tabla 3.7 se detallará la funcionalidad de los archivos principales que se incluyeron para la implementación.

**Tabla 3.7** Descripción archivos para implementación softphone web

ARCHIVO	FUNCIONALIDAD
jquery.js	Integra la aplicación con el framework JQuery
jquery-ui.js	Conjunto de interacciones para la interfaz de usuario basada en JQuery
jquery.websocket.js	Complemento para enviar y recibir objetos JSON con el API de Websockets
jquery.json.js	Complemento para facilitar la conversión a JSON
Flashphoner.js	API de Javascript para Flashphoner Web Call Server
swfobject.js	Complemento para acoplar elementos Flash
Phone-simple.js	Inicialización del API de Flashphoner Web Call Server
Phone.js	Implementación de Flashphoner
SoundControl.js	Funciones para el control de sonido
DefaultListener.js	Prototipo para el control de eventos
LoadToolListener.js	Prototipo para el control de eventos e información de carga
ConfigurationLoader.js	Prototipo para la carga de configuraciones

Para cargar las configuraciones se realiza mediante el archivo `flashphoner.xml` que se muestra en la Figura 3.24, en este archivo se define las configuraciones

básicas para la comunicación con el servidor como son la dirección para la comunicación mediante Websockets y la dirección para comunicarse mediante Flash.

```
<flashphoner>
  <url_ws_server>ws://192.168.1.11:8080</url_ws_server>
  <url_flash_server>rtmp://192.168.1.11:1935</url_flash_server>
  <register_required>true</register_required>
  <use_dtls>true</use_dtls>
  <video_width>640</video_width>
  <video_height>480</video_height>
  <check_validation_callee>true</check_validation_callee>
  <use_enhanced_mic>true</use_enhanced_mic>
  <ring_sound></ring_sound>
  <busy_sound></busy_sound>
  <register_sound></register_sound>
  <finish_sound></finish_sound>
</flashphoner>
```

**Figura 3.24** Archivo flashphoner.xml

En el archivo Phone.js se realiza la sobrescritura de los métodos del prototipo phone creado en el archivo Phone-simple.js de forma que se adapten a la implementación. El método connect se sobrescribió para que utilice la información de la extensión y la clave del usuario actual para conectarse como se observa en la Figura 3.25.

```
Phone.prototype.connect = function () {
  if ($("#outbound_proxy").val() == "") {
    $("#outbound_proxy").val($("#domain").val());
  }

  var connection = new Connection();

  connection.sipLogin = $("#extension").val();
  connection.sipPassword = $("#clave").val();
  connection.sipAuthenticationName = $("#extension").val();
  connection.sipDomain = "192.168.1.10";
  connection.sipOutboundProxy = "192.168.1.10";
  connection.sipPort = "5060";
  connection.useProxy = true;
  connection.appKey = "defaultApp";

  for (var key in connection) {
    Flashphoner.getInstance().setCookie(key, connection[key]);
  }

  var result = Flashphoner.getInstance().connect(connection);
  if (result == 0) {
    trace("Phone - Conectando");
  }
};
```

**Figura 3.25** Sobreescritura método connect

De igual forma se sobrescribió el método callStatusListener que recibe los eventos de los estados de una llamada, con el objetivo de inicializar un temporizador en cada llamada y que al terminar la llamada envíe la información de

la misma al servidor para que sea guardada en la base de datos. La función `guardarLlamada` se muestra en la Figura 3.26 esta se encargará de enviar la información de una llamada al servidor, utiliza la función `post` de JQuery que facilita el envío de información mediante AJAX mediante el método `POST`.

```
// Guardar Llamada
Phone.prototype.guardarLlamada = function(userOrigen, userDestino, proyecto,
duration, date,direction){

    if (proyecto === undefined) proyecto="";
    $.post("/Support/web/llamada/guardarllamada",{
        origen : userOrigen,
        destino : userDestino,
        proyecto : proyecto,
        duracion : duration,
        fecha : date,
        sentido : direction})
        .done(function(data){

        });
    });
};
```

**Figura 3.26** Función para guardar las llamadas desde Javascript

Los métodos que también se sobrescribieron son `connectionStatusListener`, `registrationStatusListener`, `onCallListener` con el fin de ejecutar cambios en la interfaz del usuario y sea mucho más amigable para el usuario. En la Figura 3.27 se muestra la interfaz para la marcación por teclado.



**Figura 3.27** Interfaz marcación por teclado

### 3.1.5.9.2 Marcación por usuario para operadores y administradores

Las llamadas mediante marcación por usuario son permitidas para los usuarios operadores, administradores y clientes. Para usuarios operadores y administradores se agregó al chat interno basado en Pusher botones que permiten realizar llamadas a otros usuarios, contestar llamadas entrantes y colgar las llamadas establecidas.

Para esto se procedió a modificar el archivo `layout.html.twig` en el div con identificador `templateChatBox` pues este es la plantilla para todas las cajas de chat, en este se agregó un div luego del nombre del usuario con el identificador `callButtons` y con las etiquetas correspondientes a los botones llamar, colgar y contestar, como se observa en la Figura 3.28, además se incluyó un div para que el usuario pueda ver el estado de la llamada.

```
<div class="callButtons">
  <button type="button" class="btn btn-default btn-sm
  btn-llamar"><span class="glyphicon glyphicon-earphone"
  aria-hidden="true"></span></button>
  <button type="button" class="btn btn-default btn-sm
  btn-colgar"><span class="glyphicon glyphicon-phone-alt"
  aria-hidden="true"></span></button>
  <button type="button" class="btn btn-default btn-sm
  btn-contestar"><span class="glyphicon glyphicon-open"
  aria-hidden="true"></span></button>
</div>

<div class="estado_llamada">
  Estado:<span class="mensaje_estado"></span>
</div>
```

**Figura 3.28** Etiquetas botones para llamadas

En el archivo `jquery.pusherchat.js` se agregó un método con el nombre `llamadaEntrante` que se encarga abrir una nueva caja de chat según el usuario que se envíe por el parámetro `from`, esto con el objetivo de abrir una nueva ventana de chat cuando ingrese una llamada de un usuario. En la Figura 3.29 se observa la implementación del método.

```

$.fn.pusherChat.llamadaEntrante = function(from) {
    from1 = $("input[value='"+from+"']").parent('a').attr('href');
    if(typeof from1 === 'undefined') {
        var string = "<a href='#"+from+"'"><span>"+from+"</span><input
type='hidden' value='"+from+"'"></a>";
        console.log(string);
        var obj = $('<div/>').html(string).contents();
    }else{
        from = from1.replace("#", "");
        var obj = $('a[href=#'+from+'']');
    }

    createChatBox(obj);
    if ($('title').text().search('New message - ')===-1)
        $('title').prepend('New message - ');
    $.playSound('/Support/web/sounds/new');
    return from;
}

```

**Figura 3.29**Método llamadaEntrante

Para realizar las llamadas se añadieron todos los archivos de la Tabla 3.7 con la diferencia de que el archivo Phone.js se modificó por otro denominado Phone\_bg.js, el cual tiene funcionalidades similares adaptadas a la nueva interfaz. Los cambios más importantes se exponen a continuación:

- En el método onCallListener que espera por nuevas llamadas se agregó la funcionalidad que ejecute el método llamadaEntrante de la Figura 3.29 para abrir una nueva ventana de chat y en el texto del estado cambie a Llamada Entrante.
- En el método callStatusListener que se encarga de gestionar los diferentes estados de las llamadas se agregó en cada evento la actualización del campo estado de la caja del chat.
- Se agregó los eventos para el clic en los botones de llamar, colgar y contestar como se indica en la Figura 3.30. Para el botón llamar se encuentra el destinatario de la llamada denominado callee según el cuadro de chat donde se presionó y se proceder a realizar la llamada al mismo. Para el botón colgar se evalúa si existe una llamada actual y se la cuelga caso contrario se la cancela. Para el botón contestar se verifica la existencia de una llamada actual y se la procede a contestar.



```

// Empezar llamada a usuario
$(document).on('click', 'div .callButtons button.btn-llamar', function(){
    var callee = $(this).closest('.pusherChatBox').find('h2 input').val();
    var mediaProvider = MediaProvider.Flash;
    if (Flashphoner.getInstance().mediaProviders.get(MediaProvider.WebRTC) {
        console.log("WEBRTC");
        mediaProvider = MediaProvider.WebRTC;
    }

    phone.call(callee, false, mediaProvider);
    console.log($(this).closest('.pusherChatBox').attr('id'));
    console.log("Llamando...");

    from = $("input[value='"+callee+"']").parent('a').attr('href');
    from = from.replace("#", "");
    $("#id_"+from+".pusherChatBox").find(".mensaje_estado").text("Llamando");
});

// Colgar llamada a usuario
$(document).on('click', 'div .callButtons button.btn-colgar', function(){
    if (phone.currentCall) {
        phone.hangup(phone.currentCall);
    } else {
        phone.cancel();
    }

    console.log("Colgando...");
});

// Contestar llamada a usuario
$(document).on('click', 'div .callButtons button.btn-contestar', function(){
    if (phone.currentCall) {
        phone.answer(phone.currentCall, false);
    }

    console.log("Contestar");
});

```

**Figura 3.30** Eventos clic para botones llamar, colgar y contestar

### 3.1.5.9.3 Marcación por usuario para clientes

Para este caso se incorporó una nueva vista, que según la fecha actual y el proyecto escogido, se despliegue un listado de usuarios de soporte, ordenados por la prioridad definida en la etapa respectiva. En esta implementación del softphone web como *Click-to-Call*, no se requiere implementar los eventos de llamadas entradas ya que no son necesarios.

De igual forma que en las interfaces anteriores es necesario añadir todas las dependencias de la Tabla 3.7 omitiendo el `Phone_simple.js` el archivo `Phone.js` que en este caso se simplifican a un archivo denominado `Phone_mod.js` en donde se creó una implementación básica para la generación de llamadas.



En el archivo de twig `marcacionContactos.html.twig` como se muestra en la Figura 3.31 al momento de cargar el documento se inicializa el API, se realiza la conexión con la central y se crean los eventos de clic para todos los botones llamar y colgar que se encuentren en la interfaz.

```

<script language="JavaScript">
$(document).ready(function() {
    initAPI();
    connectPhone('{{ app.user.extension.nombre }}', '{{ app.user.extension
.clave }}', '{{ app.session.get("ip_central") }}', '{{ app.session.get("port_central")
}}');
    $('.llamar').click(function(){
        var callee = $(this).closest("tr").find("input[type='hidden']").val();
        call(callee);
    });
    $('.colgar').click(function(){
        hangup();
    });
});
</script>

```

**Figura 3.31** Archivo `marcacionContactos.html.twig`

En la Figura 3.32 se observa la interfaz para la marcación de usuarios en la interfaz de los clientes. Aquí se presentan los usuarios que han sido asignados en la etapa actual, en caso de que un usuario no tenga una extensión no aparecerá en el listado hasta que se le asigne una.

Soporte Proyecto EPN Tesis			
ETAPA			
Nombre: Etapa Final			
Descripción: Etapa Final de la Implementación			
Nombre	Apellido	Departamento	Acciones
Eduardo	Gallegos	Desarrollo	Llamar Colgar
Carlos	Vaca	Ventas	Llamar Colgar

Regresar

**Figura 3.32** Interfaz Click-to-Call para clientes

#### 3.1.5.9.4 Marcación por Departamento

Para mostrar esta interfaz se escoge previamente el proyecto vinculado a la llamada que se va a realizar, en esta se generan botones *Click-to-Call* que muestra

todos los departamentos involucrados en la etapa actual para realizar la comunicación respectiva.

El funcionamiento es similar a la implementación anterior de marcación por usuarios, se utiliza el mismo archivo `Phone_mod.js` para realizar las llamadas, pero en este caso las llamadas a cada departamento serán realizadas al usuario de cada departamento con mayor prioridad en la asignación de la etapa. En la Figura 3.33 se observa la interfaz visual para la marcación a departamentos.

**Soporte Proyecto EPN Tesis**

**ETAPA**

**Nombre:** Etapa Final

**Descripción:** Etapa Final de la Implementación

Departamento	Acciones
Desarrollo	<input type="button" value="Llamar"/> <input type="button" value="Colgar"/>
Ventas	<input type="button" value="Llamar"/> <input type="button" value="Colgar"/>

**Figura 3.33** Marcación por departamento

### 3.1.5.10 Implementación Chat de Soporte

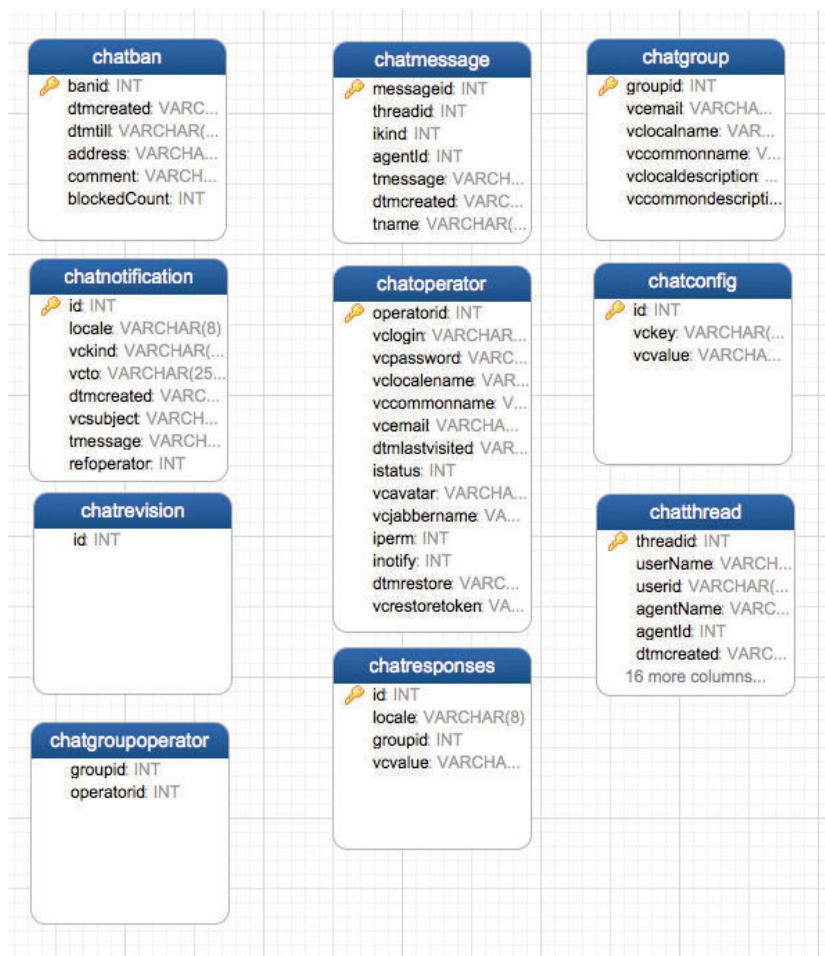
El chat de soporte permite que los usuarios sean atendidos por los agentes asignados a cada departamento, por este motivo para implementar esta funcionalidad al sistema se debe integrar los usuarios y los departamentos del chat de soporte en la gestión de usuarios del sistema web, de igual forma se integró el acceso automático y el acoplamiento de la interfaz del chat de soporte en la interfaz del sistema.

#### 3.1.5.10.1 Instalación Chat de Soporte Open Source

Para la instalación del Chat de Soporte utilizaremos el software Open Source Webim, para esto se descarga el programa de la página <http://mibew.org/download>. A continuación se extrae el contenido del archivo y

se copia los archivos extraídos en el directorio público de Apache/var/www/html/webim para que sea accesible mediante peticiones HTTP y se procede con la modificación del archivo config.php que se encuentra en el directorio chat/libs de la aplicación y se edita la información asociada con la base de datos webim que se creó anteriormente en la implementación de MySQL.

Para la creación de las tablas necesarias se debe ingresar a la URL <http://webim.support.lan/install> en donde se desplegarán alertas en caso de una versión PHP no compatible o falta de permisos y cuando estén resueltas se ejecutarán las consultas a la base de datos para la creación de las tablas. La Figura 3.34 muestra las tablas creadas por el chat de soporte luego de la instalación.



**Figura 3.34** Diagrama de Tablas Chat de Soporte

### 3.1.5.10.2 Integración Usuarios

La creación, edición o actualización de la información de los usuarios se debe omitir del chat de soporte y estas acciones solo quedan permitidas a usuarios con los permisos adecuados desde la interfaz de la plataforma web de soporte. Para esto se incorporan las funciones utilizadas por la gestión de usuarios del chat de soporte en el sistema web de soporte, en la Figura 3.35 se observa el código para la creación de un usuario en el método `createAction` del controlador `UserController`, de esta forma se genera una petición al formulario de creación de usuarios de Webim y se ejecuta cuando se cree un usuario mediante la interfaz del sistema de soporte. De igual manera se implementó para la edición y eliminación de usuarios manteniendo la consistencia de la información.

```

//Crear usuario en webim
if(!$entity->isCliente()) {
    $params = array("action" => "create", "login" =>
$entity->getUsername(), "password" => $password, "email" => $entity->getEmail());
    $query = http_build_query($params);
    $contextData = array(
        'method' => 'POST',
        'header' => "Connection: close\r\n" .
            "Content-Type: application/x-www-form-urlencoded\r\n" .
            "Content-Length: " . strlen($query) . "\r\n",
        'content' => $query);

    $context = stream_context_create(array('http' => $contextData));
    $result = file_get_contents(
        'http://webim.support.lan/operator/egoperator.php',
        false,
        $context);
    $logger = $this->get('logger');
    $logger->info('El resultado del ingreso para webim es:' .
$result);

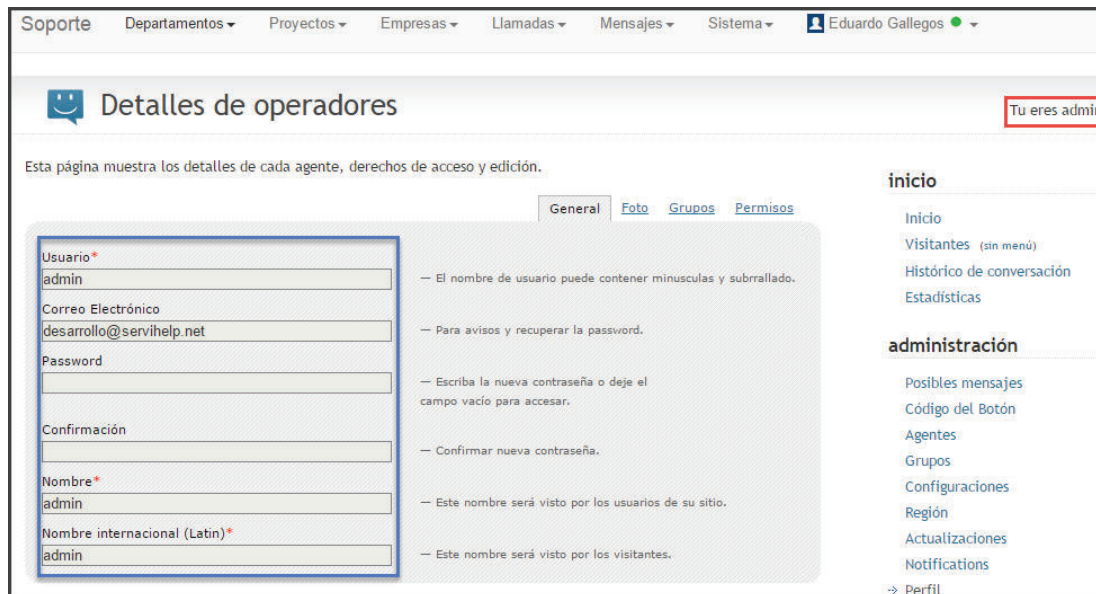
    //Terminado la creación de usuario en webim
    return $this->redirect($this->generateUrl('user_show', array('id'
=> $entity->getId())));
}

```

**Figura 3.35** Creación usuario Webmin desde sistema web

Posteriormente se deshabilitó las acciones de gestión de usuarios en la interfaz del chat de soporte para que de esta manera solo se pueda realizar la gestión desde el sistema integral y evitar la posible creación de inconsistencias, esto se realiza mediante la modificación del código en las vistas, para desactivar la modificación y permitir únicamente la visualización de los usuarios. Como se ilustra en la Figura

3.36 la edición del usuario tiene los cuadros deshabilitados, a pesar de que el usuario registrado es el administrador denominado admin.



**Figura 3.36** Edición de Usuarios Webmin

### 3.1.5.10.3 Integración de Departamentos

En el chat de Soporte los departamentos son relacionados con los grupos, pues cumplen funciones similares agrupando a los usuarios. Para esto se modificó los métodos del controlador DepartamentoController que permiten la creación, actualización y eliminación de Departamentos.

En la Figura 3.37 se muestra la actualización de los departamentos en Webim desde el método actualizarAction del sistema web, en esta sección de código se eliminan todas las relaciones que tenga el departamento en la tabla chatgroup y se procede a crear nuevas relaciones para todos los usuarios que en la interfaz visual se le asignaron al departamento. Los métodos createAction y deleteAction también fueron modificados para agregar la creación y eliminación de departamentos respectivamente.

```

/* Actualizar Departamento y Usuarios Asociados (Inicio) */
$em1 = $this->getDoctrine()->getManager('webim');

$stmt = $em1->getConnection()
->prepare('select groupid from chatgroup where vclocalname="'
.$nombre_old.'"');
$stmt->execute();
$result = $stmt->fetchAll();
foreach ($result as $res){
    $groupid = $res['groupid'];
}

$stmt = $em1->getConnection()
->prepare('delete from chatgroupoperator where groupid='.$groupid);
$stmt->execute();

if(!is_null($empleados)) {
    foreach ($empleados as $empleado) {
        $emp = $em->getRepository('EGSoBundle:User')->find($empleado);
        $stmt = $em1->getConnection()
->prepare('select operatorid from chatoperator where vclogin="'
.$emp->getUsername().'"');
        $stmt->execute();
        $result = $stmt->fetchAll();
        foreach ($result as $res){
            $soperid = $res['operatorid'];
        }

        $stmt = $em1->getConnection()
->prepare('insert into chatgroupoperator (groupid,operatorid)
VALUES ( '.$groupid.', '.$soperid.' )');
        $stmt->execute();
    }
}
}

```

**Figura 3.37** Actualización departamentos en Webmin

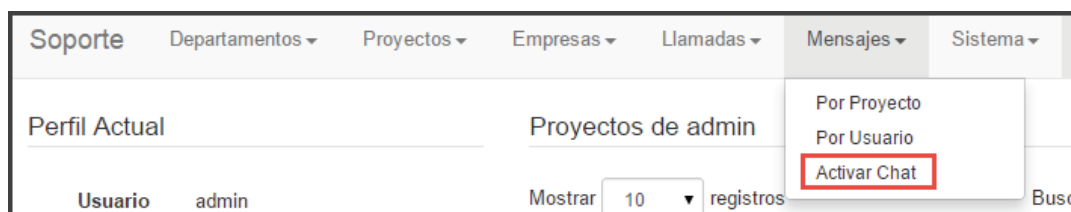
#### 3.1.5.10.4 Integración de Acceso

Con los usuarios sincronizados entre plataformas, el acceso a las mismas se va a centralizar en el formulario de acceso del sistema web, permitiendo que el usuario acceda a la plataforma de soporte web y que sin necesidad de introducir otra vez las credenciales ingrese al chat de soporte.

El formulario de acceso al chat de soporte de Webmin se eliminó y se aumentó un enlace en el menú del sistema que crea una nueva sesión de Webmin y mediante un `iframe` carga en la parte inferior la consola de administración del programa, de tal manera que los usuarios que requieran acceder a la consola de administración



del chat deben ingresar en la plataforma de soporte web y si tiene los permisos necesarios ir al enlace Activar Chat que se encuentra en el Menú como se presenta en la Figura 3.38.



**Figura 3.38** Menú para Acceso al Chat de Soporte

### 3.1.5.11 Implementación Servicios Web

Para la integración con la aplicación móvil se necesita un módulo de servicios web de tipo REST que se encarga de recibir peticiones de tipo http y devolver respuestas en JSON o XML. Se implementa sobre Symfony utilizando las funcionalidades de Doctrine y el complemento FOSRestBundle que se agregó en la instalación inicial.

Para la creación de los métodos que van a responder a las solicitudes realizadas se creó un controlador denominado RestController que extiende de la clase FOSRestController, esto permite que los métodos que empiecen con get solo soporten peticiones realizadas por GET, aquellos que empiezan por post solo soportan peticiones tipo POST, entre otros. En la

Tabla 3.8 se muestran todos los métodos creados y su respectiva funcionalidad.

**Tabla 3.8** Funciones REST implementadas

FUNCIÓN	MÉTODO	DESCRIPCIÓN
getChatsbyidstate	GET	Devuelve los chats según el id del agente y el estado del chat
getActivechatsbyid	GET	Devuelve chats que están asignados al id del agente o no tienen asignación
getLoginAction	GET	Confirma que el usuario puede acceder a la aplicación
getMessagesbythread	GET	Devuelve los mensajes de un chat por el id
postSendMessage	POST	Recibe un mensaje y lo añade a la base de datos

Las rutas son creadas automáticamente como se visualiza en la Figura 3.39 que muestra las rutas mediante el comando `php app/console route:debug`, en el final de cada ruta permite especificar el tipo de formato que puede ser XML o JSON, para esta implementación solo se utilizará JSON.

<code>get_chatsbyidstate</code>	GET	ANY	ANY	<code>/rest/chatsbyidstates/{id}</code>
<code>{_format}</code>				
<code>get_activechatsbyid</code>	GET	ANY	ANY	<code>/rest/activechatsbyids/{id}</code>
<code>{_format}</code>				
<code>get_login</code>	GET	ANY	ANY	<code>/rest/logins/{user}{_format}</code>
<code>get_messagesbythread</code>	GET	ANY	ANY	<code>/rest/messagesbythreads/{id}{_format}</code>
<code>post_login</code>	POST	ANY	ANY	<code>/rest/logins{_format}</code>
<code>post_send_mensaje</code>	POST	ANY	ANY	<code>/rest/sends/mensajes{_format}</code>

**Figura 3.39** Rutas para funciones REST

#### 3.1.5.11.1 Seguridad

La seguridad de los servicios web basados en REST se realizó mediante WSSE (Web Service Security), puesto que a diferencia de las aplicaciones web que se mantienen autenticadas mediante una variable de sesión en los servicios web es necesario enviar información sobre la autenticación en cada solicitud. Utilizando WSSE se envía en la cabecera la información de autenticación que se indica a continuación:

- El nombre de usuario.
- Una cadena de caracteres aleatoria denominada nonce.
- Un sello de tiempo de la fecha de creación.
- Un resumen de la contraseña creada a partir del nonce, la fecha de creación y la contraseña.

En la Figura 3.40 se muestra una captura de una solicitud que lleva en la cabecera la información indicada anteriormente para autenticarse en el sistema web.



```

Frame 29404: 573 bytes on wire (4584 bits), 573 bytes captured (4584 bits) on interface 0
Ethernet II, Src: Apple_06:2b:d1 (4c:7c:5f:06:2b:d1), Dst: Tp-LinkT_0c:b7:36 (e8:de:27:0c:b7:36)
Internet Protocol Version 4, Src: 192.168.1.101 (192.168.1.101), Dst: 192.168.1.106 (192.168.1.106)
Transmission Control Protocol, Src Port: 50622 (50622), Dst Port: 8081 (8081), Seq: 689, Ack: 2656, Len: 507
Hypertext Transfer Protocol
  GET /Support/web/app_dev.php/rest/logins/1.json?username=admin HTTP/1.1\r\n
  Host: 192.168.1.106:8081\r\n
  Accept: */*\r\n
  X-WSSE: UsernameToken Username="admin", PasswordDigest="DpOcr/HPG7HARLH1g/HLd2sC4iU=", Nonce="NEYwNku0MDhBMEQ2MzJBRT"
  Cookie: m0baw_tocate=es; PHPSESSID=dnb4jnmj6m1n0z08pgsue1zK4\r\n
  User-Agent: Aplicacion/1.0 CFNetwork/711.2.23 Darwin/14.0.0\r\n
  Accept-Language: es-es\r\n
  Accept-Encoding: gzip, deflate\r\n
  Connection: keep-alive\r\n
\r\n
[Full request URI: http://192.168.1.106:8081/Support/web/app_dev.php/rest/logins/1.json?username=admin]
[HTTP request 3/3]
[Prev request in frame: 29397]
[Response in frame: 29413]

```

**Figura 3.40** Captura Wireshark solicitud mediante WSSE

La implementación de este tipo de autenticación en Symfony corresponde a crear un proveedor de autenticación personalizado que se encargue de verificar que la información enviada en la cabecera de la solicitud es correcta. La configuración correspondiente a todas las rutas que solicitaran la información de autenticación mediante WSSE se realiza en el archivo `security.yml` en la sección `firewalls` como se muestra en la Figura 3.41, aquí se crea uno nuevo denominado `wsse_secured` y en el atributo `pattern` se define la URL base `/rest/.*` para que aplique a todas las funciones REST.

```

firewalls:
  wsse_secured:
    pattern: /rest/.*
    stateless: true
    wsse: true

```

**Figura 3.41** Configuración WSSE en `security.yml`

### 3.1.6 IMPLEMENTACIÓN APLICACIÓN MÓVIL

La aplicación será implementada para teléfonos móviles iPhone con iOS 7 o superior, para desarrollar este tipo de aplicaciones solo se lo puede realizar mediante un equipo Mac instalado el IDE(Integrated Development Enviroment) Xcode y el simulador de iOS. De igual forma es necesario un iPhone con soporte para iOS versión 7 u 8, en el cuál se realizarán pruebas reales de la aplicación pues el simulador por sí solo no permite realizar pruebas con notificaciones *push*, para ejecutar estas pruebas es necesaria la adquisición de una suscripción a iOS

Developer Program en la página de Apple. La aplicación móvil se desarrolló completamente en lenguaje nativo Objective-C.

Para realizar esta implementación se contará con los recursos que se muestran en la Tabla 3.9.

**Tabla 3.9** Recursos para desarrollo aplicación móvil

<b>Hardware</b>	
Referencia	Características
Macbook Air	Procesador 1.8 Ghz Intel Core i5, Memoria 4GB, SSD 128 GB
iPhone 6	iOS 8
<b>Software</b>	
Referencia	Versión
OS X Mavericks	10.9
Xcode	6
<b>Recursos Adicionales</b>	
Suscripción iOS Developer Program	

### 3.1.6.1 Librerías Adicionales

Para facilitar la implementación de algunas opciones se optó por utilizar algunas librerías adicionales a las provistas por defecto en Objective-C.

#### 3.1.6.1.1 AFNetworking

Es un *framework* para Objective-C que facilita la interacción con URLs y la gestión de comunicaciones con servidores que usan protocolos de internet como HTTP, HTTPS o FTP. En la aplicación se utiliza para la gestión de peticiones y respuestas con la aplicación web mediante servicios Web. También gestiona la notificación de uso de internet en el sistema operativo.

#### 3.1.6.1.2 *JSMessagesViewController*

Es una librería con la implementación de la interfaz visual para chats mediante mensajes cortos. Está soportada para chats grupales, por lo que cuenta con todas las características necesarias para mostrar los mensajes en la aplicación.

#### 3.1.6.1.3 *Parse*

Es una librería provista por el servicio Parse que nos ayuda a interactuar con el servicio para el envío y recepción de notificaciones *push*, además con la gestión de datos almacenados en la plataforma Parse.

#### 3.1.6.1.4 *CocoaWSSE*

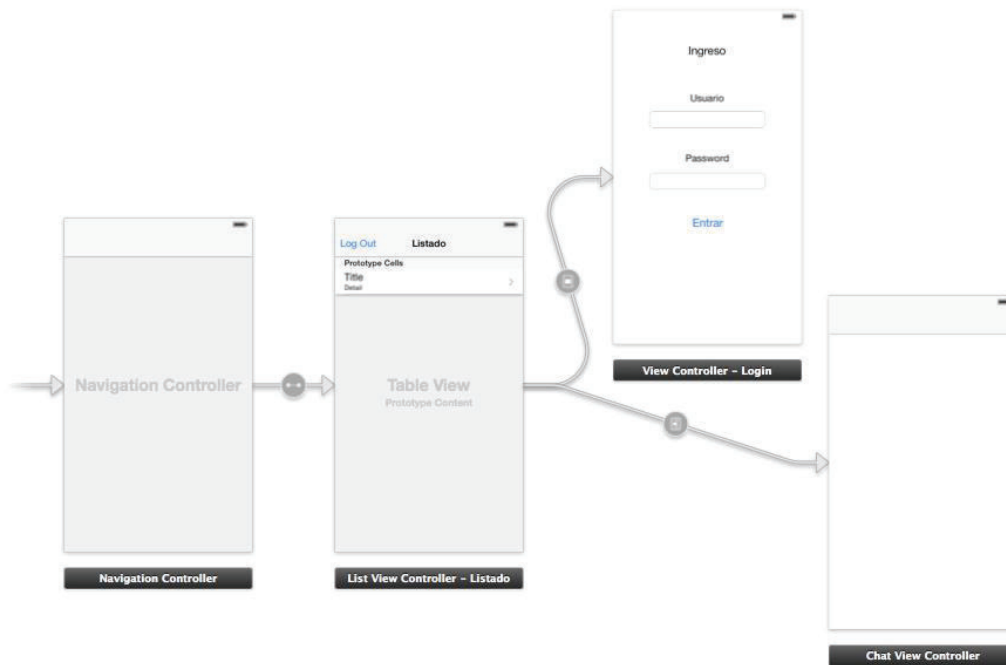
Esta librería facilita la creación de cabeceras de autenticación para WSSE, que serán utilizadas para comunicarse con las funciones REST del sistema.

### 3.1.6.2 Interfaz de Usuario

Para construir la interfaz de usuario Xcode provee la herramienta Interface Builder que permite hacerlo de manera visual y con gran facilidad. La aplicación consta de las siguientes interfaces:

- Acceso
- Listado de Chats
- Mensajes

En la Figura 3.42 se muestra la interacción entre las interfaces de usuario.



**Figura 3.42** Diagrama de Interacción Aplicación Móvil

#### 3.1.6.2.1 Acceso

Esta pantalla permite el acceso a la aplicación mediante el ingreso del nombre de usuario y la contraseña. Esta información es utilizada para crear la cabecera necesaria en la autenticación WSSE. El servidor verifica que los datos en la cabecera sean correctos y la función llamada `loginAction` confirmar que el usuario que está accediendo corresponde a un usuario administrador y no a un cliente, permitiendo el ingreso y almacenando las variables del nombre de usuario, la contraseña cifrada y el *salt* en variables de usuario y en los `AppDelegate`, de igual forma se registra en la base de datos de Parse el usuario asociado al móvil actual para enviar las notificaciones *push* correspondientes. En la Figura 3.43 se muestra el proceso principal de autenticación en la aplicación móvil.

```

NSString *url = [[NSString alloc] initWithFormat:@"http://192.168.1.106:8081/Support/web/app_dev.php/rest/
logins/1.json?username=%@", [self.txtUsername text]];

NSMutableURLRequest* req = [NSMutableURLRequest requestWithURL:[NSURL URLWithString:url]];
[req setValue:[CocoaWSSSE headerWithUsername:[self.txtUsername text] password:hashedString]
forHTTPHeaderField:@"X-WSSSE"];

[NSURLConnection sendAsynchronousRequest:req queue:[NSOperationQueue mainQueue] completionHandler:^(
NSURLResponse *res, NSData *data, NSError *err) {

    NSHTTPURLResponse *response = (NSHTTPURLResponse *) res;
    NSLog(@"response status code: %ld", (long)[response statusCode]);

    NSLog(@"Response code: %ld", (long)[response statusCode]);
    if ([response statusCode] >=200 && [response statusCode] < 300)
    {
        NSString *responseData = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];
        NSLog(@"Response ==> %@", responseData);

        NSError *error = nil;
        NSDictionary *jsonData = [NSJSONSerialization JSONObjectWithData:data options:
        NSJSONReadingMutableContainers error:&error];
        NSLog(@"%@", jsonData);
        success = [jsonData[@"success"] integerValue];
        iden = jsonData[@"id"];
        if(success == 1)
        {
            AppDelegate *delegate = [UIApplication sharedApplication].delegate;
            delegate.hashed = hashedString;
            NSLog(@"Login SUCCESS");

            [[NSUserDefaults standardUserDefaults] setObject:self.txtUsername.text forKey:@"userID"];
            [[NSUserDefaults standardUserDefaults] setObject:iden forKey:@"id"];
            [[NSUserDefaults standardUserDefaults] setObject:hashedString forKey:@"hashedString"];
            [[NSUserDefaults standardUserDefaults] setObject:@"true" forKey:@"logged"];
            [[NSUserDefaults standardUserDefaults] synchronize];

            PFInstallation *installation = [PFInstallation currentInstallation];
            [installation setObject:self.txtUsername.text forKey:@"usuario"];
            [installation saveInBackground];
        }
    }
}

```

**Figura 3.43** Código para autenticación de usuarios

### 3.1.6.2.2 Listado de Chats

La lista de chats se muestra en función del usuario que ingreso a la aplicación. Con la información del usuario la aplicación hace una petición al servidor en busca de todos los chats que se encuentran activos en espera de un agente o asociados al usuario actual. Esta solicitud es enviada al servidor cuando la pantalla es cargada, por tal motivo en caso de que el usuario cierre la aplicación sin salir de su usuario, al momento de ejecutar nuevamente la aplicación se cargaran los chats actualizados. En la Figura 3.44 se muestra la carga de los chats en el tableView que es el contenedor principal de la interfaz del listado de chats.

```

NSString *user = [[NSUserDefaults standardUserDefaults] objectForKey:@"userID"];
NSString *hashedString = [[NSUserDefaults standardUserDefaults] objectForKey:@"hashedString"];
NSURL *url = [NSURL URLWithString:[NSString stringWithFormat:@"http://192.168.1.106:8081/Support/web/app_dev.php/rest/
messagesbythreads/%@.json", iden]];

NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:url];
[request setValue:[CocoaWSE headerWithUsername:user password:hashedString] forHTTPHeaderField:@"X-WSE"];

AFHTTPRequestOperation *operation = [[AFHTTPRequestOperation alloc] initWithRequest:request];
operation.responseSerializer = [AFJSONResponseSerializer serializer];
[operation setCompletionBlockWithSuccess:^(AFHTTPRequestOperation *operation, id responseObject)
{
    self.mensajes = [responseObject objectForKey:@"result"];
    NSString *ident = [[NSUserDefaults standardUserDefaults] objectForKey:@"id"];
    NSString *iden = [NSString alloc] initWithFormat:@"%@", iden];
    NSMutableArray *mensajesTemp = [NSMutableArray alloc] initWithCapacity:self.mensajes.count];
    for(id msg in self.mensajes){
        NSLog(@"%@",msg);
        if([msg[@"agentId"] isEqualToString:@"1"]){
            NSLog(@"a:%@!=%b:",msg[@"agentId"],iden);
        }
        if([@"1" isEqualToString:iden]){
            NSLog(@"b:%@!=%a:",msg[@"agentId"],iden);
        }

        if([msg[@"ikind"] isEqualToString:@"2"] && [msg[@"agentId"] isEqualToString:iden]){
            JSMessage *aux = [JSMessage alloc] initWithText:msg[@"tmessage"] sender:kSubtitleMe date:[NSDate date];
            [mensajesTemp addObject:aux];
            NSLog(@"%@",@"1");
        }else if ([msg[@"ikind"] isEqualToString:@"2"] && ![msg[@"agentId"] isEqualToString:iden]){
            JSMessage *aux = [JSMessage alloc] initWithText:msg[@"tmessage"] sender:kSubtitleAgente date:[NSDate date];
            [mensajesTemp addObject:aux];
            NSLog(@"%@",@"2");
        }else if ([msg[@"ikind"] isEqualToString:@"1"]){
            JSMessage *aux = [JSMessage alloc] initWithText:msg[@"tmessage"] sender:kSubtitleCliente date:[NSDate date];
            ;
            [mensajesTemp addObject:aux];
            NSLog(@"%@",@"3");
        }else{
            JSMessage *aux = [JSMessage alloc] initWithText:msg[@"tmessage"] sender:kSubtitleInfo date:[NSDate date];
            [mensajesTemp addObject:aux];
            NSLog(@"%@",@"4");
        }
    }
    self.messages = mensajesTemp;
    [self.tableView reloadData];
    [self scrollToBottomAnimated:NO];
}failure:^(AFHTTPRequestOperation *operation, NSError *error){
    NSLog(@"Request Failed: %@", error, error.userInfo);
}];

```

Figura 3.44 Código carga de chats en el tableView

### 3.1.6.2.3 Mensajes

La interfaz de mensajes permite interactuar con el cliente asociado al chat que se haya seleccionado en el listado de Chats. Para esta interfaz se integró JSMessagesViewController, que se encarga de mostrar los mensaje en burbujas asociadas cada una a un avatar específico que representa el usuario que generó ese mensaje como se visualiza en la Figura 3.45, se crearon 4 tipos de avatar correspondientes a usuario cliente, otro agente, usuario actual y avisos del chat con el objetivo de diferenciar los diferentes actores de los mensajes que se muestra en la interfaz de Webim.



**Figura 3.45** Interfaz de Mensajes

En la zona inferior como se observa en la Figura 3.45 se muestra un recuadro que al presionarlo mostrará el teclado virtual, permitiendo escribir un mensaje para ser enviado. Al presionar el botón Send, el mensaje será enviado mediante POST a la función de REST `postSendMessage` que se encargará de incluirlo en la base de datos para que el cliente sea notificado.

### 3.1.6.3 Notificaciones Push

La implementación de notificaciones *push* se va a utilizar el servicio ofrecido por Parse que es un servicio gratuito que nos brinda una base de datos en la nube con un límite de 30 peticiones por segundo, 20 GB de capacidad y el envío ilimitado de notificaciones *push* hasta para 1 millón de dispositivos. El uso de este servicio facilita en gran medida la implementación de las notificaciones, ya que caso contrario sería necesario crear un servidor de notificaciones propio que permita gestionar los móviles activos y el envío de notificaciones a cada uno de ellos.



Para utilizar Parse para el envío de notificaciones *push* es necesario crear una aplicación en el servicio y adjuntarle los certificados creados en Apple, con esto el servicio Parse nos entregará un identificador de la aplicación creada, una llave para el API REST y una llave maestra que serán utilizadas para inicializar la instancia de la clase `ParseClient` y comunicarse con el servicio.

### 3.1.6.3.1 Envío de notificaciones *push*

El envío de notificaciones en el sistema se lo va a generar cuando se ha enviado un nuevo mensaje por parte del cliente o cuando el cliente ha empezado una conversación. Para crear estas notificaciones Parse tiene un SDK de PHP que se integró al código de Webmin, como se observa en la Figura 3.46 se agregó en el archivo `client.php` el código de envío de una notificación *push* a todos los móviles que se encuentren registrados con un usuario, en el momento que el cliente ha creado una nueva ventana de conversación.

```
$data = array("alert" => "Nuevo Chat Iniciado");
$query = ParseInstallation::query();
$query->notEqualTo("usuario", "");
ParsePush::send(array(
    "where" => $query,
    "data" => $data
));

header("Location: $mibewroot/client
.php?thread=$threadid&token=$token&level=$level".($chatstyle ?
"&style=$chatstyle" : ""));
```

**Figura 3.46** Envío Notificación *push* nuevo chat

En la Figura 3.47 se visualiza el código insertado en el archivo `thread.php` de Webmin, que envía una notificación *push* cuando el cliente ha enviado un mensaje, el código envía una notificación al usuario agente que se encuentre operando la conversación y en caso de no existir alguno se envía a todos los usuarios que se encuentren activos en la aplicación móvil. También junto con el mensaje de la notificación se envía información como el `threadid` que corresponde al identificador del chat, para que en la recepción de la notificación se cargue la vista de mensajes de esa conversación.



```

$postedid = post_message_($threadid,$kind,$message,$link,$from,null,$isuser ? null
: $operator['operatorid'] );
if($kind == $kind_user){
    if(!is_null($thread['agentName'])) {
        $data = array("alert" => $thread['userName'] . ":" . $message,
            "threadid" => $threadid,
            'title' => "Nuevo Mensaje",
            "titulo" => $thread['userName'],
            'agentName' => $thread['agentName'],
            'p' => 'ok'
        );
        $query = ParseInstallation::query();
        $query->equalTo("usuario", $thread['agentName']);
        ParsePush::send(array(
            "where" => $query,
            "data" => $data
        ));
    }else{
        $data = array("alert" => "Usuario esperando respuesta");
        $query = ParseInstallation::query();
        $query->notEqualTo("usuario", "");
        ParsePush::send(array(
            "where" => $query,
            "data" => $data
        ));
    }
}
}

```

**Figura 3.47** Envío notificaciones push nuevo mensaje

### 3.1.6.3.2 Recepción notificaciones push

La recepción de las notificaciones se realiza en la aplicación móvil, en donde existen dos circunstancias la primera que la aplicación sea abierta mediante la notificación y la segunda que la notificación sea recibida cuando la aplicación este en ejecución. En el primer caso la información se recibe en el método `didFinishLaunchingWithOptions` de la clase `AppDelegate` como se muestra en la Figura 3.48 y le indica a la aplicación móvil que debe cargar el chat asociado al mensaje nuevo.

```

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [[AFNetworkActivityIndicatorManager sharedManager] setEnabled:YES];

    NSDictionary *notificationPayload = launchOptions[UIApplicationLaunchOptionsRemoteNotificationKey];

    if ([[notificationPayload objectForKey:@"p"] isEqualToString:@"ok"]){
        ChatViewController *chatViewController =
        (ChatViewController*)[[ChatViewController alloc] init];
        chatViewController.informacion = [[NSDictionary alloc] initWithObjectsAndKeys:[notificationPayload
        objectForKey:@"titulo"], @"titulo",[notificationPayload objectForKey:@"threadid"], @"id",[notificationPayload
        objectForKey:@"agentName"], @"agentname", nil];
        [(UINavigationController *)self.window.rootViewController pushViewController:chatViewController animated:YES];
    }
}

```

**Figura 3.48** Recepción notificación push 1

En el segundo caso la información se recibe en el método `didReceiveRemoteNotification` de la clase `AppDelegate` como se muestra en la Figura 3.49 y de igual forma que el anterior le indica a la aplicación móvil que debe cargar el chat asociado al mensaje recibido.

```

- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo{
    if ([[userInfo objectForKey:@"p"] isEqualToString:@"ok"]){
        ChatViewController *chatViewController =
        (ChatViewController*)[[ChatViewController alloc] init];
        chatViewController.informacion = [[NSDictionary alloc] initWithObjectsAndKeys:[userInfo objectForKey:@"titulo"],
        @"titulo",[userInfo objectForKey:@"threadid"] , @"id",[userInfo objectForKey:@"agentName"], @"agentname", nil];
        [[UINavigationController *)self.window.rootViewController pushViewController:chatViewController animated:YES];
    }
}

```

**Figura 3.49** Recepción notificación push 2

### 3.1.7 CONFIGURACIÓN CENTRAL TELEFÓNICA

#### 3.1.7.1 Extensiones

Cada usuario de la aplicación para poder realizar llamadas y recibirlas debe tener una extensión con la cual se registrará en el webphone automáticamente en el sistema web. Las extensiones no necesitan una configuración diferente al resto de extensiones y serán configuradas manualmente en la central telefónica según los usuarios creados en el sistema web. El registro SIP de la extensión con la central telefónica se lo puede realizar desde varias instancias al mismo tiempo, por este motivo el usuario puede estar registrado con la plataforma y con el teléfono IP físico sin ningún problema. En la Figura 3.50 se muestra la interfaz para la creación de extensiones en la central telefónica utilizada en la implementación.

**Create New Accounts**

Accounts are numbers that you can call on the PBX. Please select the account type that you would like to create.  
**Tip:** You may enter more than one name. If you use a space between the names, the PBX will set up several accounts for you. For example, "123/theo 124/fred" will set up two accounts: the first with the name 123 and the alias name theo and the second with the names 124 and fred.

Account Type:

Dial Plan:


Plug and Play:

	Number	First Name	Last Name	SIP Pass.	Web Pass.	PIN	Email
1	1001	Soporte	Desarrollo				
2							
3							
4							

**Figura 3.50** Interfaz Creación Extensiones en la Central Telefónica

### 3.1.7.2 Troncales

Para fines de la implementación del prototipo se utilizará un Gateway FXO que permita comunicarse con una base celular y realizar llamadas a la Telefonía Pública a través de la misma. Por este motivo se creó una troncal denominada PSTN Gateway y se la configuró con la información del Gateway como se observa en la Figura 3.51.



**Edit Trunk PSTN Gateway**

Name:	PSTN Gateway
Type:	SIP Gateway
Direction:	Inbound and outbound
Account:	
Domain:	192.168.1.200
Username:	1478
Password:	.....
Password (repeat):	.....
Outbound Proxy:	192.168.1.200
CO Lines:	
Permissions to monitor this account:	

**Figura 3.51** Configuración Troncal

### 3.1.7.3 Planes de Mercado

Los planes de mercado permiten a los administradores definir la ruta que tomarán las llamadas cuando sean realizadas desde una extensión según el número marcado. Los planes de mercado que se agregaron a la central telefónica según el diseño del capítulo anterior son los siguientes:

### 3.1.7.3.1 Plan Webphone

Permite a los usuarios utilizar la plataforma para hacer llamadas entre usuarios y extensiones de la central telefónica pero no permite realizar llamadas a números externos. Este plan es creado para los usuarios clientes.

**Edit Dial Plan Webphone**

Quick Usage: Use simple patterns for matching the input (for example, "9\*" or "911"), and just leave the replacement empty. Please see the more information on how to use the advanced features of the dial plan.

Name:

Global:  yes  no

Pref	Trunk	Pattern	Replacement
<input type="text" value="100"/>	<input type="text" value="Unassigned"/>	<input type="text"/>	<input type="text"/>

**Figura 3.52** Plan de Marcado Webphone

Como se muestra en la Figura 3.52 para no permitir llamadas a números externos no se debe crear ninguna ruta y de esta manera solo se permitirán llamadas a extensiones.

### 3.1.7.3.2 Plan Nacional

El plan de marcado debe permitir llamadas a números convencionales locales y nacionales. No tiene permiso para llamadas internacionales o celulares. Este plan es creado para usuarios que realizan soporte y necesitan comunicarse con los clientes.

**Edit Dial Plan Nacional**

Quick Usage: Use simple patterns for matching the input (for example, "9\*" or "911"), and just leave the replacement empty. Please see the more information on how to use the advanced features of the dial plan.

Name:

Global:  yes  no

Pref	Trunk	Pattern	Replacement
<input type="text" value="100"/>	<input type="text" value="Unassigned"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="100"/>	<input type="text" value="PSTN Gateway"/>	<input type="text" value="^(0[3,4,5,6,7][0-9]*)@*"/>	<input type="text"/>
<input type="text" value="200"/>	<input type="text" value="PSTN Gateway"/>	<input type="text" value="^([2,3,4,5,6][0-9]*)@*"/>	<input type="text"/>

**Figura 3.53** Plan de Marcado Nacional

Como se muestra en la Figura 3.53 se creó dos rutas por la troncal PSTN Gateway correspondientes a números locales y números nacionales.

Los patrones utilizados son los siguientes:

**Locales:** `^[2,3,4,5,6][0-9]*.@*`

**Nacional:** `^(0[3,4,5,6,7][0-9]*).@*`

### 3.1.7.3.3 Plan Celular

Este plan debe permitir similares opciones al plan Nacional pero se le adicionará una ruta correspondiente a llamadas a números celulares. Será creado para usuarios de soporte que según el administrador de la plataforma crea conveniente la activación de llamadas a celulares.

**Edit Dial Plan Celular**

Quick Usage: Use simple patterns for matching the input (for example, "9\*" or "911"), and just leave the replacement empty. Please see more information on how to use the advanced features of the dial plan.

Name:

Global:  yes  no

Pref	Trunk	Pattern	Replacement
<input type="text" value="100"/>	<input type="text" value="Unassigned"/>	<input type="text" value=""/>	<input type="text" value=""/>
<input type="text" value="100"/>	<input type="text" value="PSTN Gateway"/>	<input type="text" value="^(09[0-9]*).@*"/>	<input type="text" value=""/>
<input type="text" value="100"/>	<input type="text" value="PSTN Gateway"/>	<input type="text" value="^(0[3,4,5,6,7][0-9]*).@*"/>	<input type="text" value=""/>
<input type="text" value="100"/>	<input type="text" value="PSTN Gateway"/>	<input type="text" value="^[2,3,4,5,6][0-9]*.@*"/>	<input type="text" value=""/>

**Figura 3.54** Plan de Mercado Celular

Como se indica en la Figura 3.54 se creó 3 rutas para los diferentes números de destino.

Los patrones utilizados son los siguientes:

**Locales:** `^[2,3,4,5,6][0-9]*.@*`

**Nacional:** `^(0[3,4,5,6,7][0-9]*).@*`

**Celular:** `^(09[0-9]*).@*`

### 3.1.7.3.4 Plan Internacional

Este plan tiene permisos similares al plan de marcado Celular pero adicionalmente permite llamadas a destinos internacionales. Este plan será asignado a los usuarios de soporte que lo requieran.

Quick Usage: Use simple patterns for matching the input (for example, "9\*\*" or "911"), and just leave the replacement empty. Please see more information on how to use the advanced features of the dial plan.

Pref	Trunk	Pattern	Replacement
100	Unassigned		
100	PSTN Gateway	^[2,3,4,5,6][0-9]*.@*	
100	PSTN Gateway	^[0][3,4,5,6,7][0-9]*.@*	
100	PSTN Gateway	^[09][0-9]*.@*	
100	PSTN Gateway	^[00][1-9]*.@*	

Save

**Figura 3.55** Plan de Marcado Internacional

En la Figura 3.55 se puede observar que para este plan se crearon 4 rutas por la troncal PSTN Gateway.

Los patrones utilizados son los siguientes:

**Locales:** `^[2,3,4,5,6][0-9]*.@*`

**Nacional:** `^[0][3,4,5,6,7][0-9]*.@*`

**Celular:** `^[09][0-9]*.@*`

**Internacional:** `^[00][1-9]*.@*`

## 3.2 MANUAL DEL USUARIO

El manual de usuario está realizado específicamente para el sistema web integral, en donde se puede ver información sobre los siguientes módulos:

- Usuarios y Roles
- Departamentos

- Proyectos
- Llamadas
- Chat
- Cliente
- Aplicación Móvil

Este manual se encuentra en el Anexo B.

### **3.3 PRUEBAS**

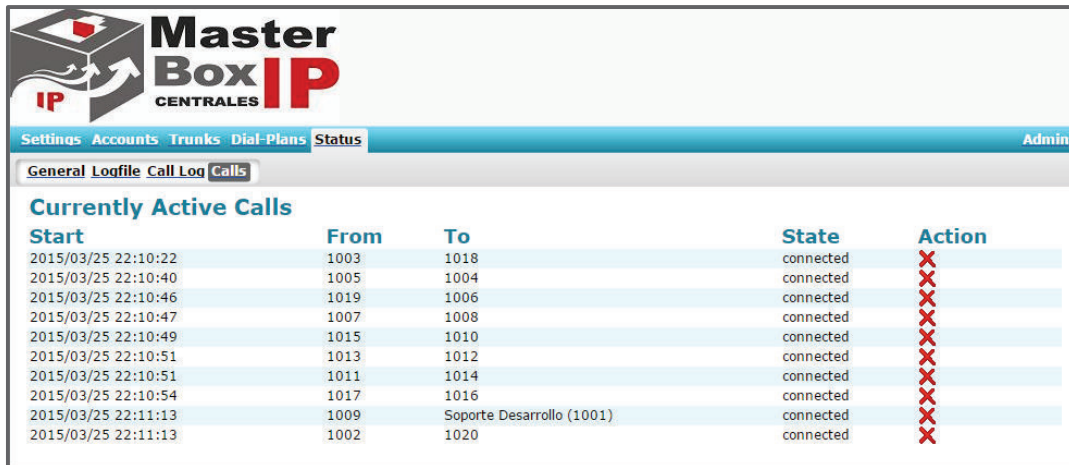
Las pruebas se realizarán en el entorno utilizado para la implementación, de esta forma se podrá determinar la carga que requiere el sistema para dimensionar los recursos que garanticen el correcto funcionamiento según las especificaciones del prototipo.

#### **3.3.1 PRUEBAS FLASHPHONER WEBCALL SERVER**

Con la consideración de que el prototipo debe dimensionarse para un total de 15 usuarios de soporte de ventas y técnico más los usuarios clientes, el escenario más crítico corresponde a 15 llamadas concurrentes, pues significaría todos los usuarios de soporte estarían ocupados, por este motivo en las pruebas se tomará en cuenta escenarios alrededor del valor crítico para evaluar su funcionamiento. Los escenarios que se utilizarán en las pruebas son los siguientes:

- 10 llamadas concurrentes
- 20 llamadas concurrentes
- 30 llamadas concurrentes

La Figura 3.56 corresponde a una captura del registro de la central telefónica de 10 llamadas concurrentes realizadas con su respectiva hora.

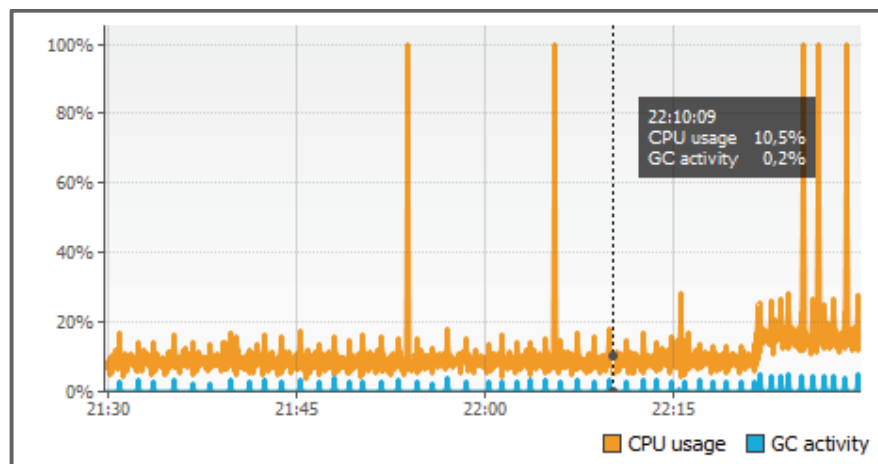


The screenshot shows the MasterBox IP interface with a navigation menu (Settings, Accounts, Trunks, Dial-Plans, Status, Admin) and a sub-menu (General, Logfile, Call Log, Calls). The main content area displays a table titled 'Currently Active Calls' with the following data:

Start	From	To	State	Action
2015/03/25 22:10:22	1003	1018	connected	X
2015/03/25 22:10:40	1005	1004	connected	X
2015/03/25 22:10:46	1019	1006	connected	X
2015/03/25 22:10:47	1007	1008	connected	X
2015/03/25 22:10:49	1015	1010	connected	X
2015/03/25 22:10:51	1013	1012	connected	X
2015/03/25 22:10:51	1011	1014	connected	X
2015/03/25 22:10:54	1017	1016	connected	X
2015/03/25 22:11:13	1009	Soporte Desarrollo (1001)	connected	X
2015/03/25 22:11:13	1002	1020	connected	X

**Figura 3.56** Captura 10 llamadas central telefónica

La Figura 3.57 muestra un gráfico que relaciona el consumo de procesador y el tiempo. Se puede observar que desde las 21:30 hasta las 22:20 se tiene un consumo cercano a 11%, esto corresponde al consumo registrado cuando existen 10 llamadas simultáneas.



**Figura 3.57** Consumo CPU 10 llamadas

En la Figura 3.58 se indica el consumo de memoria y se puede observar que en el lapso de tiempo cuando se realizaron las 10 llamadas concurrentes el consumo de memoria promedio aproximado corresponde a 90 MB.



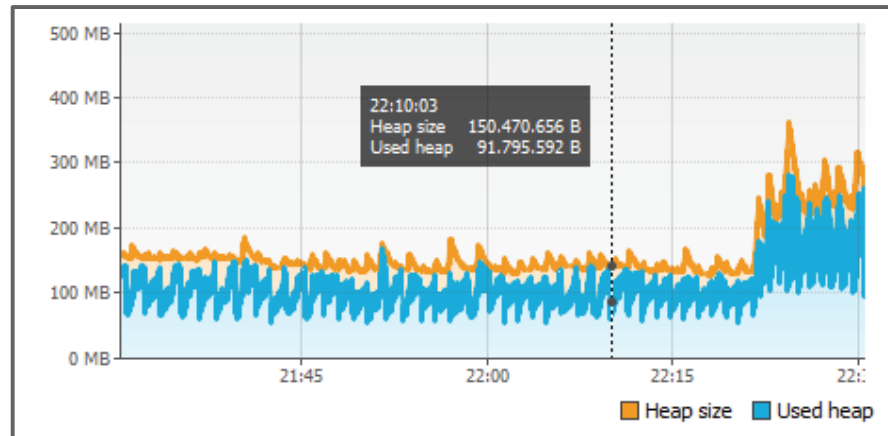


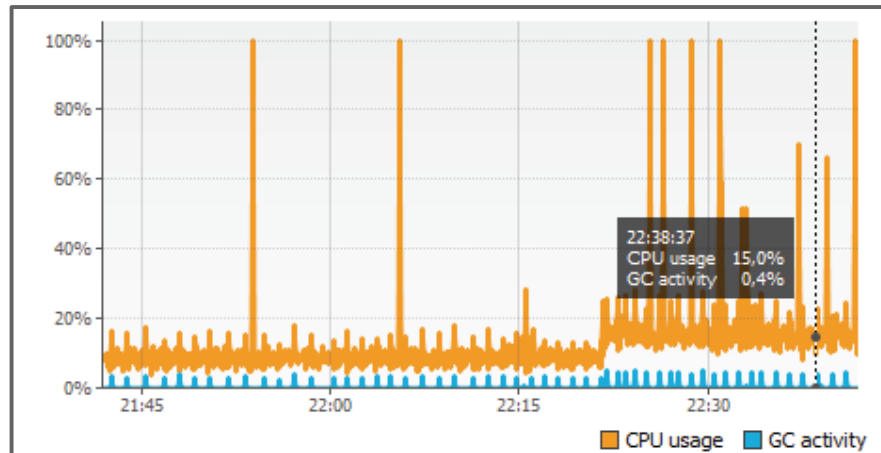
Figura 3.58 Consumo Memoria 10 llamadas

En la Figura 3.59 se visualiza una captura de la central telefónica con 20 llamadas concurrentes y se indica la hora en que las llamadas son realizadas.

Master Box IP CENTRALES		Settings Accounts Trunks Dial-Plans Status Admin		
General Logfile Call Log Calls				
Currently Active Calls				
Start	From	To	State	Action
2015/03/25 22:37:21	1009	1008	connected	X
2015/03/25 22:38:00	1029	1020	connected	X
2015/03/25 22:38:01	1002	1022	connected	X
2015/03/25 22:38:05	1027	1004	connected	X
2015/03/25 22:38:06	1003	1024	connected	X
2015/03/25 22:38:07	1007	1026	connected	X
2015/03/25 22:38:09	1005	1028	connected	X
2015/03/25 22:38:10	1023	1006	connected	X
2015/03/25 22:38:17	1037	1008	connected	X
2015/03/25 22:38:21	1017	Soporte Desarrollo (1001)	connected	X
2015/03/25 22:38:22	1011	1010	connected	X
2015/03/25 22:38:26	1015	1012	connected	X
2015/03/25 22:38:26	1021	1014	connected	X
2015/03/25 22:38:27	1033	1018	connected	X
2015/03/25 22:38:31	1039	1030	connected	X
2015/03/25 22:38:32	1019	1016	connected	X
2015/03/25 22:38:33	1031	1032	connected	X
2015/03/25 22:38:34	1013	1034	connected	X
2015/03/25 22:38:34	1025	1036	connected	X
2015/03/25 22:38:39	1035	1034	connected	X

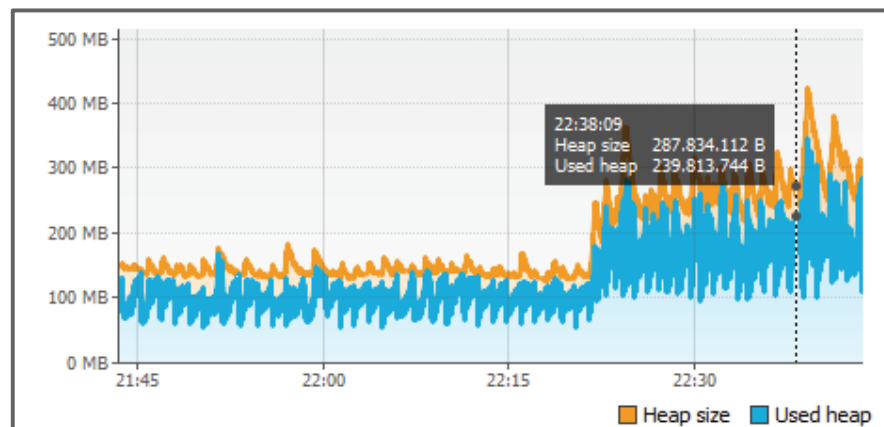
Figura 3.59 Captura 20 llamadas central telefónica

La Figura 3.60 presenta un gráfico relacionando el consumo de procesador y el tiempo. Se puede observar que desde las 22:20 hasta las 22:45 cuando existen 20 llamadas concurrentes, se tiene un consumo aproximado de 15%.



**Figura 3.60** Consumo CPU 20 llamadas

En la Figura 3.61 se presenta el consumo de memoria evaluado en el lapso de tiempo cuando se realizaron 20 llamadas concurrentes, mostrando un consumo de memoria promedio aproximado de 200 MB.



**Figura 3.61** Consumo memoria 20 llamadas

En la Figura 3.62 se indica la captura del monitoreo de la central telefónica cuando existen 30 llamadas concurrentes.

Settings Accounts Trunks Dial-Plans Status Admin				
General Logfile Call Log Calls				
Currently Active Calls				
Start	From	To	State	Action
2015/03/25 23:09:55	1002	1006	connected	X
2015/03/25 23:10:01	1027	1010	connected	X
2015/03/25 23:10:02	1039	1022	connected	X
2015/03/25 23:10:05	1043	1024	connected	X
2015/03/25 23:10:06	1037	1026	connected	X
2015/03/25 23:10:06	1025	1028	connected	X
2015/03/25 23:10:08	1021	1030	connected	X
2015/03/25 23:10:08	1041	1032	connected	X
2015/03/25 23:10:09	1031	1036	connected	X
2015/03/25 23:10:09	1029	1034	connected	X
2015/03/25 23:10:11	1023	1038	connected	X
2015/03/25 23:10:12	1053	1040	connected	X
2015/03/25 23:10:12	1033	1042	connected	X
2015/03/25 23:10:15	1035	1044	connected	X
2015/03/25 23:10:16	1055	1048	connected	X
2015/03/25 23:10:16	1057	1046	connected	X
2015/03/25 23:10:19	1059	Soporte Desarrollo (1001)	connected	X
2015/03/25 23:10:20	1003	1004	connected	X
2015/03/25 23:10:21	1045	1008	connected	X
2015/03/25 23:10:25	1049	1012	connected	X
2015/03/25 23:10:26	1013	1014	connected	X
2015/03/25 23:10:27	1051	1016	connected	X
2015/03/25 23:10:27	1011	1020	connected	X
2015/03/25 23:10:28	1015	1050	connected	X
2015/03/25 23:10:29	1005	1058	connected	X
2015/03/25 23:10:30	1017	1016	connected	X
2015/03/25 23:10:36	1007	1052	connected	X
2015/03/25 23:10:40	1019	1018	connected	X
2015/03/25 23:10:44	1047	1054	connected	X
2015/03/25 23:10:46	1009	1056	connected	X

Figura 3.62 Captura 30 llamadas central telefónica

La Figura 3.63 presenta la relación entre el consumo de procesador y el tiempo. Se observa que desde las 23:10 hasta las 23:17 cuando se registran 30 llamadas concurrentes, se tiene un consumo aproximado de 21%.

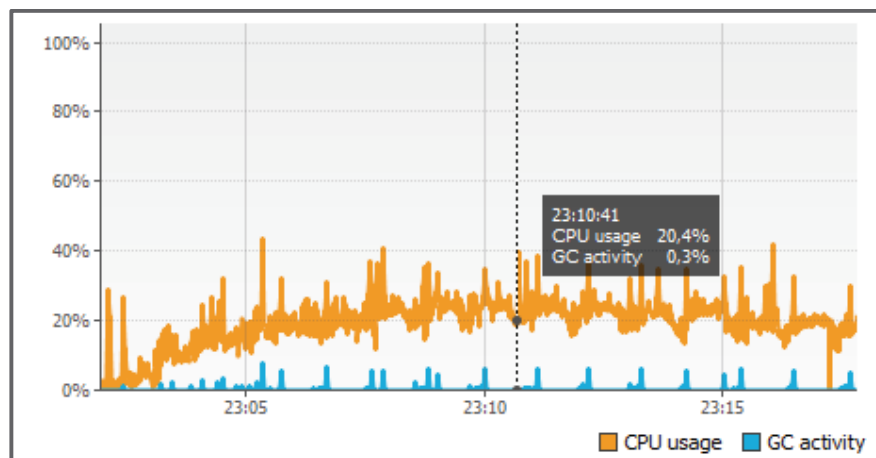
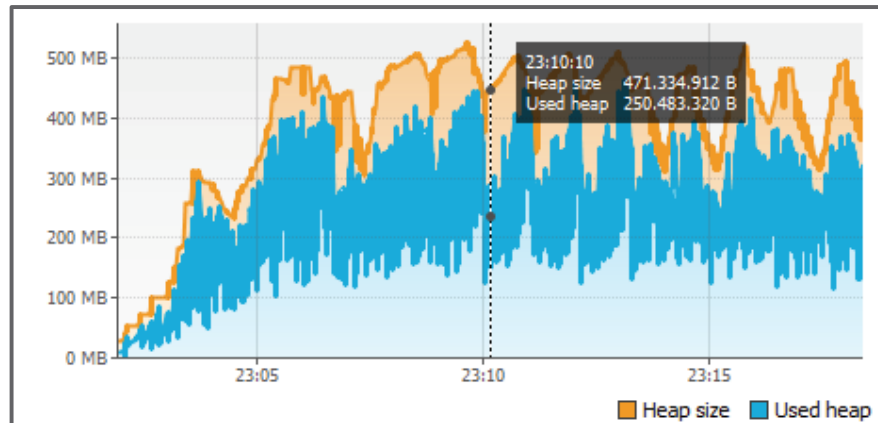


Figura 3.63 Consumo CPU 30 llamadas

El consumo de memoria promedio cuando existen 30 llamadas concurrentes corresponde aproximadamente a 300 MB como se indica en la Figura 3.64.



**Figura 3.64** Consumo memoria 30 llamadas

En la Tabla 3.10 se muestra el resumen de las pruebas realizadas al servidor.

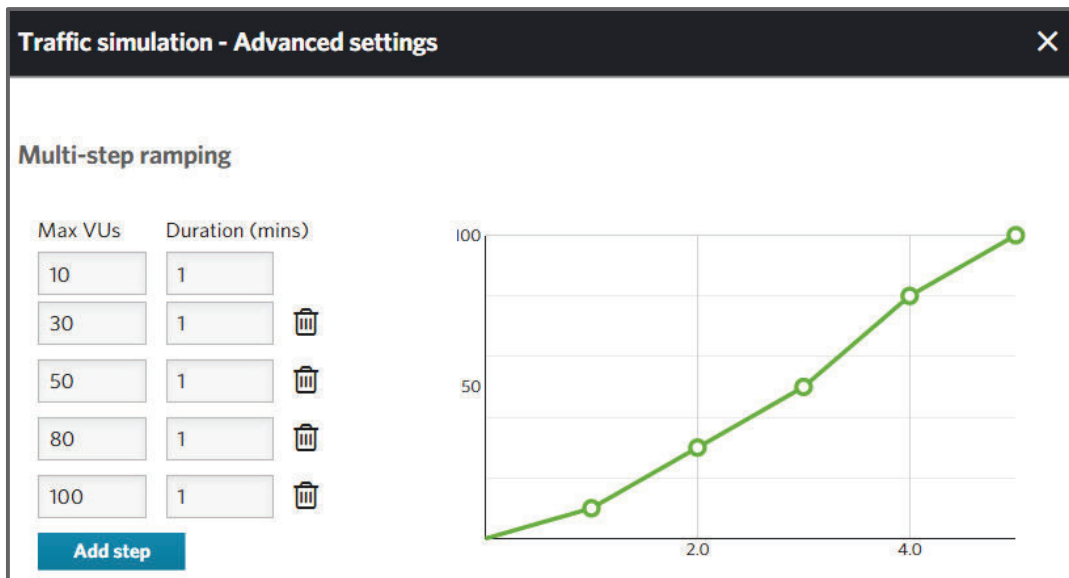
**Tabla 3.10** Resumen pruebas Web Call Server

Llamadas Concurrentes	Consumo Memoria	Consumo Procesador
10	90 MB	11%
20	200 MB	15%
30	300 MB	21%

### 3.3.2 PRUEBAS SERVIDOR WEB

Las pruebas del servidor web se realizaron mediante el servicio ofrecido por Load Impact, que permite realizar pruebas de carga a sistemas web. El servicio es ofrecido gratuitamente con un límite máximo de 100 clientes virtuales, 5 minutos por prueba y 5 pruebas al mes.

Para la prueba se ha utilizado las condiciones mostradas en la Figura 3.65, en esta se presenta que las cargas de clientes virtuales irán aumentando progresivamente cada minuto en el siguiente orden 10, 30, 50, 80 y 100.

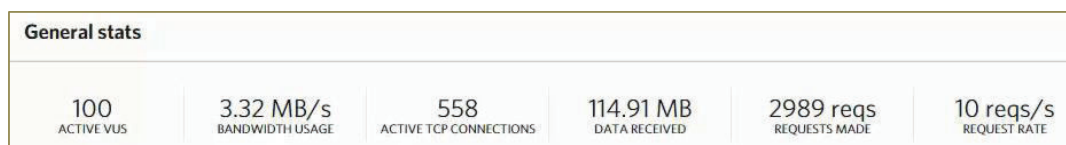


**Figura 3.65** Configuración simulación de tráfico

Cada prueba ejecuta llamadas a 4 páginas cada una que involucran las siguientes acciones:

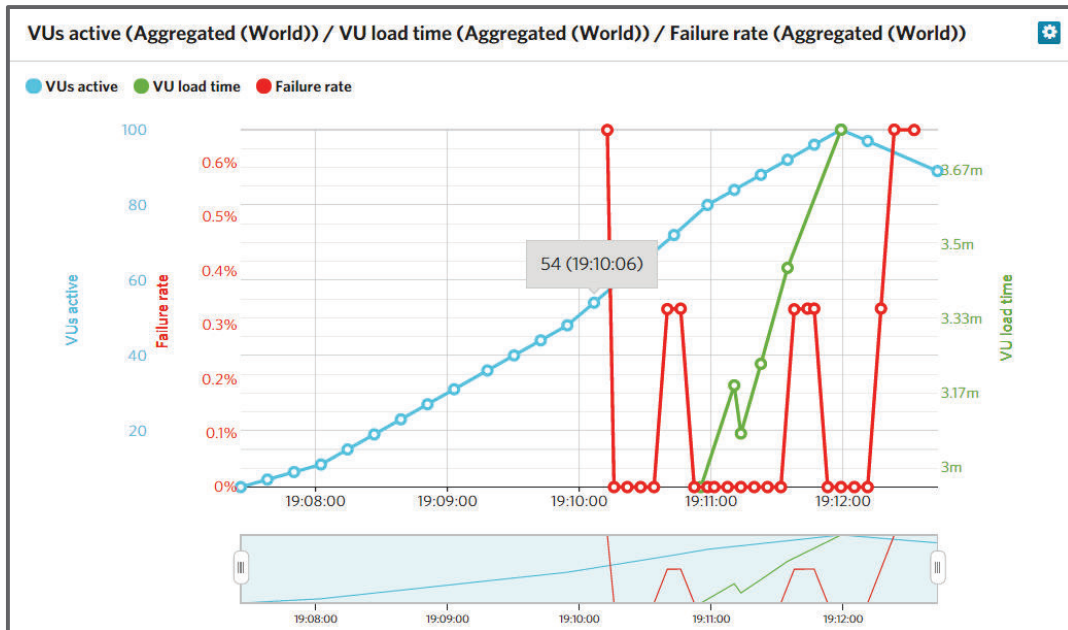
- Acceso al sistema
- Verificación de información
- Perfil del usuario
- Mostrar listado de departamentos

En la Figura 3.66 se observan las estadísticas generales de la prueba de carga.



**Figura 3.66** Estadísticas de prueba de carga

La Figura 3.67 muestra una gráfica con los datos recopilados de la prueba correspondiente a los clientes virtuales conectados, los tiempos de carga promedio según la cantidad de clientes y la cantidad de fallos ocurridos.



**Figura 3.67** Gráfica de resultados prueba de carga

De la Figura 3.67 se concluye que el sistema web opera con normalidad hasta 54 clientes virtuales concurrentes, luego de esto el sistema empieza a generar fallas.

### 3.3.3 ANÁLISIS WEBSOCKETS

En la Figura 3.68 se observa la captura de la comunicación por Websockets mediante la herramienta para desarrolladores de Google Chrome, en el recuadro rojo se muestra la URL de la petición correspondiente a `ws://192.168.1.11:8080` que es la dirección del servicio de Websockets del Flashphoner Web Call Server, esto se realiza con el objetivo de mantener un canal de comunicaciones para intercambiar los mensajes de señalización de las llamadas.

En el recuadro azul de la Figura 3.68 se indican algunos de los mensajes intercambiados y se tiene seleccionado un mensaje correspondiente a una notificación de llamada entrante que como se puede observar está en formato JSON<sup>11</sup> y tiene toda la información correspondiente a la llamada, de esta manera se realiza el intercambio de información entre el servidor y el cliente.

<sup>11</sup>JSON Javascript Object Notation – Formato ligero para intercambio de datos





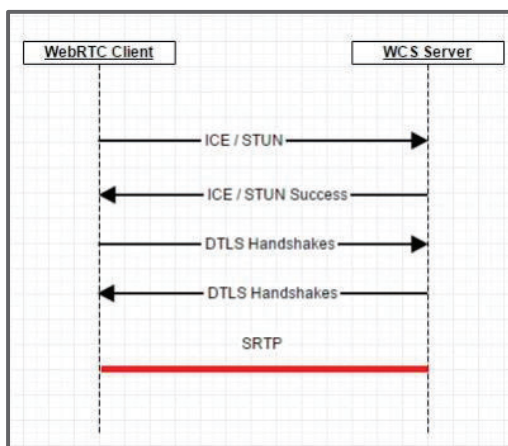
**Tabla 3.11** Campos paquete WebSocket

Campo - Valor	Extensión	Descripción
<b>Fin: True</b>	1 bit	Indica que es el fragmento final de un mensaje, en este caso Verdadero o 1
<b>Reserved: 0x00</b>	3 bits	Deben estar en cero, para uso de extensiones al protocolo.
<b>Opcode: Text(1)</b>	4 bits	Define la interpretación de los datos, en este caso Texto
<b>Mask: False</b>	1 bit	Identifica si los datos van enmascarados o no, en este caso en Falso o 0
<b>Payload Length : 126</b>	7 bits	Longitud de la carga útil, si es 126 o 127 existe una extensión de la longitud de la carga útil. En este caso 126 los siguientes 2 bytes representan la longitud de la carga útil.
<b>Extended Payload Length: 511</b>	16 bits	Extensión de la longitud de la carga útil, en este caso 511 bits son de carga
<b>Payload</b>	511 bits	Carga útil, en este caso notificación en formato JSON de una llamada entrante.

### 3.3.4 ANALISIS WEBRTC

Como se muestra en la Figura 3.70, los protocolos que utiliza WebRTC para su comunicación son los siguientes:

- Interactive Connectivity Establishment (ICE) / Session Traversal Utilities for NAT (STUN)
- Datagram Transport Layer Security (DTLS)
- Secure Real-Time Transport Protocol (SRTP)



**Figura 3.70** Modelo comunicación WebRTC[22]



El establecimiento de conexiones en WebRTC se efectúa con el protocolo ICE, que se encarga de establecer un túnel entre los clientes, para este caso utiliza el protocolo STUN que permite identificar la IP y puerto de cada uno. El cliente envía peticiones STUN al Web Call Server y este responde confirmando que la comunicación se realizó satisfactoriamente.

Como se presenta en la Figura 3.71 se han capturado los paquetes STUN en Wireshark que muestran peticiones de confirmación para la comunicación mediante la IP y puerto de un usuario y la misma es respondida con la comprobación de la comunicación. Este proceso se realiza para encontrar la ruta de conexión entre clientes.

No.	Time	Source	Destination	Protocol	Length	Info
260	8.326740	192.168.1.141	192.168.1.11	STUN	146	Binding Request user: ah5eu:9zResj2MgcvhI5Y0
261	8.327155	192.168.1.11	192.168.1.141	STUN	150	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.1.141:55265 user:
263	8.346323	192.168.1.11	192.168.1.141	STUN	158	Binding Request user: 9zResj2MgcvhI5Y0:ah5eu
265	8.347764	192.168.1.141	192.168.1.11	STUN	106	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.1.11:31290
271	8.366283	192.168.1.11	192.168.1.141	STUN	86	Binding Indication
274	8.375524	192.168.1.141	192.168.1.11	STUN	146	Binding Request user: ah5eu:9zResj2MgcvhI5Y0
275	8.375745	192.168.1.11	192.168.1.141	STUN	150	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.1.141:55265 user
330	8.856563	192.168.1.141	192.168.1.11	STUN	146	Binding Request user: ah5eu:9zResj2MgcvhI5Y0
331	8.856843	192.168.1.11	192.168.1.141	STUN	150	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.1.141:55265 user:
416	9.337591	192.168.1.141	192.168.1.11	STUN	146	Binding Request user: ah5eu:9zResj2MgcvhI5Y0
417	9.337842	192.168.1.11	192.168.1.141	STUN	150	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.1.141:55265 user:
469	9.818639	192.168.1.141	192.168.1.11	STUN	146	Binding Request user: ah5eu:9zResj2MgcvhI5Y0
470	9.818969	192.168.1.11	192.168.1.141	STUN	150	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.1.141:55265 user:
549	10.299849	192.168.1.141	192.168.1.11	STUN	146	Binding Request user: ah5eu:9zResj2MgcvhI5Y0

**Figura 3.71** Captura Wireshark protocolo ICE/STUN

Los datos enviados por WebRTC deben estar encriptados para esto tiene que intercambiar certificados, este proceso se realiza mediante DTLS ya que cumple la función de TLS pero para UDP. En la Figura 3.72 se observa la captura de pantalla de Wireshark con el intercambio de mensajes DTLS entre el cliente y el servidor para asegurar la conexión para la transmisión de los datos. DTLS añade fragmentación y secuencia para el proceso de *handshake* como se muestra en los cuadros rojos de la Figura 3.72.

No.	Time	Source	Destination	Protocol	Length	Info
946	635.198585000	192.168.1.106	192.168.1.11	DTLSv1.0	170	Client Hello
955	635.217671000	192.168.1.106	192.168.1.11	DTLSv1.0	168	Client Hello
958	635.230417000	192.168.1.11	192.168.1.106	DTLSv1.0	853	Server Hello, Certificate,
965	635.246602000	192.168.1.106	192.168.1.11	DTLSv1.0	823	Certificate, Client Key Ex
970	635.255367000	192.168.1.11	192.168.1.106	DTLSv1.0	133	change cipher spec, Encryp

User Datagram Protocol, Src Port: 59315 (59315), Dst Port: 3127 (3127)

Datagram Transport Layer Security

DTLSv1.0 Record Layer: Handshake Protocol: Client Hello

Content Type: Handshake (22)

Version: DTLS 1.0 (0xfeff)

Epoch: 0

Sequence Number: 0

Length: 113

Handshake Protocol: Client Hello

Handshake Type: Client Hello (1)

Length: 101

Message Sequence: 0

Fragment Offset: 0

Fragment Length: 101

Version: DTLS 1.0 (0xfeff)

**Figura 3.72** Captura Wireshark protocolo DTLS

El tráfico de audio es enviado mediante SRTP el cual es reconocido en Wireshark como RTP puesto que es una extensión del mismo. En la Figura 3.73 se muestra los flujos de tramas RTP y se encuentran seleccionados los flujos correspondientes al envío de información entre el cliente WebRTC con la IP 192.168.1.106 y el Web Call Server con la IP 192.168.1.11.

Src addr	Src port	Dst addr	Dst port	SSRC	Payload	Packets	Lost
192.168.1.106	51703	192.168.1.11	31912	0x505C115	g711A	277	0 (0.0%)
192.168.1.11	31632	192.168.1.30	58282	0x6ED4B76A	g711U	458	0 (0.0%)
192.168.1.11	31796	192.168.1.30	56196	0x686A22EF	g711U	277	0 (0.0%)
192.168.1.11	31912	192.168.1.106	51703	0x5E36D78C	g711A	276	0 (0.0%)
192.168.1.30	58282	192.168.1.11	31632	0xAF555984	g711U	464	0 (0.0%)
192.168.1.30	56196	192.168.1.11	31796	0x6ACB1960	g711U	282	0 (0.0%)

Forward: 192.168.1.106:51703 -> 192.168.1.11:31912, SSRC=0x505C115  
Reverse: 192.168.1.11:31912 -> 192.168.1.106:51703, SSRC=0x5E36D78C

**Figura 3.73** Flujo de tramas RTP

### 3.3.5 ANÁLISIS SIP

El intercambio de tráfico SIP se lo realiza entre el Web Call Server (WCS) con la IP 192.168.1.11 y la central telefónica con la IP 192.168.1.30 puesto que todos los clientes web se comunican mediante el WCS. En la Figura 3.74 se presentan las etapas de una llamada y son detalladas a continuación:

- El WCS envía un *INVITE SDP* inicia la comunicación con la central telefónica.
- La central telefónica responde con *100 Trying* informando que acepta la invitación.
- La central telefónica envía un mensajes 401 solicitando autenticación.
- El WCS responde con un *ACK* confirmando que recibió la solicitud.
- EL WCS reenvía un *INVITE SDP* con la información de autenticación incluida.
- La central telefónica responde con *100 Trying* informando que acepta la invitación.
- La central telefónica inicia el proceso de timbrado con el mensaje *183 Ringing SDP*.
- La central telefónica envía datos mediante RTP que corresponde a un mensaje pregrabado de la central telefónica indicando la extensión destino.
- La central telefónica continúa el proceso de timbrado enviando mensajes *183 Ringing SDP*.
- La central telefónica envía un mensaje *200 OK SDP* cuando el cliente contesta.
- El WCS envía una *ACK* confirmando que recibió el mensaje *200 OK* del servidor.
- Se empieza la comunicación de datos mediante RTP.
- Cuando un cliente desea terminar la llamada se envía un mensaje *BYE*.
- EL WCS envía un mensaje *200 OK SDP* confirmando la liberación de la llamada.

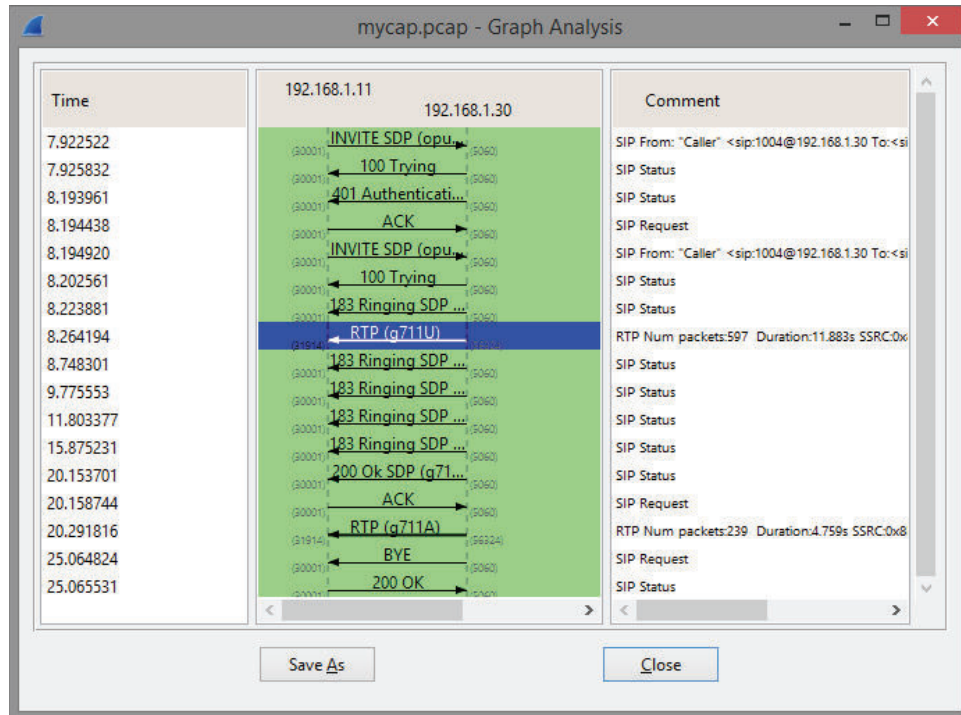


Figura 3.74 Captura Wireshark flujo SIP

### 3.4 DIMENSIONAMIENTO DE RECURSOS Y SERVIDORES

Según los requerimientos del prototipo se debe cumplir con las siguientes condiciones:

- Debe soportar mínimo 10 usuarios de soporte técnico.
- Debe soportar mínimo 5 usuarios de ventas.
- Orientado para empresas tipo integradoras de servicio.
- Pruebas en ambientes de laboratorio.

#### 3.4.1 DIMENSIONAMIENTO SERVIDOR WEB

El servidor web del prototipo permitirá una cantidad de usuarios mínima igual a 15 usuarios de soporte entre personal técnico y de ventas más los usuarios cliente que sean necesarios. El servidor virtual web utilizado para la implementación según las

pruebas realizadas soporta hasta 54 usuarios web sin mostrar fallos, lo que cubre los 15 usuarios de soporte y permite un crecimiento de hasta 39 usuarios adicionales de soporte o clientes. Por este motivo el servidor web que se utilizará en el prototipo corresponde al mismo de la implementación y desarrollo de la aplicación web integral con las características indicadas en la Figura 3.1.

### 3.4.2 DIMENSIONAMIENTO SERVIDOR MULTIMEDIA

El servidor multimedia corresponde al Web Call Server que es el que gestiona todas las llamadas realizadas y recibidas mediante la interfaz Web. Para realizar el dimensionamiento de las llamadas concurrentes necesarias para una empresa tipo integradora de servicios se realizó un estudio de la cantidad máxima de llamadas concurrentes analizadas cada minuto en una empresa con estas características.

Para el estudio se consideró la información del registro de llamadas de la empresa de una semana y se analizó cada minuto la cantidad de llamadas concurrentes para obtener la mayor cantidad registrada. Los resultados obtenidos en este estudio se muestran en la Tabla 3.12.

**Tabla 3.12** Llamadas concurrentes empresa tipo

<b>Etapa</b>	<b>Llamadas Simultáneas</b>	<b>Repeticiones</b>
<b>Diario</b>	2	28
	3	10
	4	2
<b>Semanal</b>	2	138
	3	36
	4	8

Con los resultados se puede concluir que la mayor cantidad de llamadas concurrentes corresponde a 4, que se puede considerar como el límite mínimo de llamadas concurrentes que debe soportar el servidor.

Con las pruebas realizadas en el WCS empleado en la implementación se pudo definir que soportó un total de 30 llamadas concurrentes consumiendo una cantidad menor al 30% de los recursos del servidor. De esta manera en el prototipo será usado el servidor virtual utilizado en la implementación con las características expuestas en la Figura 3.5.

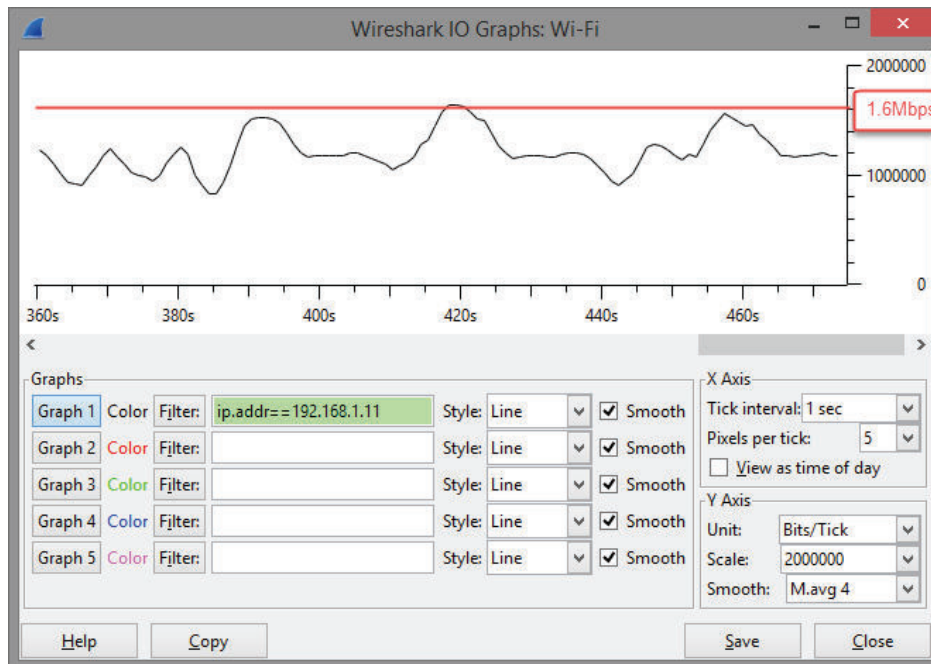
### **3.4.3 DIMENSIONAMIENTO CONEXIÓN A INTERNET**

Para las pruebas efectuadas en el ambiente de laboratorio la tasa de transferencia de datos no es considerable puesto que los servicios de Pusher para la mensajería y Parse para el envío de notificaciones consumen recursos esporádicamente y son mínimos.

En un ambiente de producción es necesario considerar la tasa de transferencia requerida pues los usuarios deberían poder conectarse a través de internet a una IP Pública, en consecuencia el tráfico de llamadas debe ser reservado según la cantidad de llamadas concurrentes que se va a garantizar.

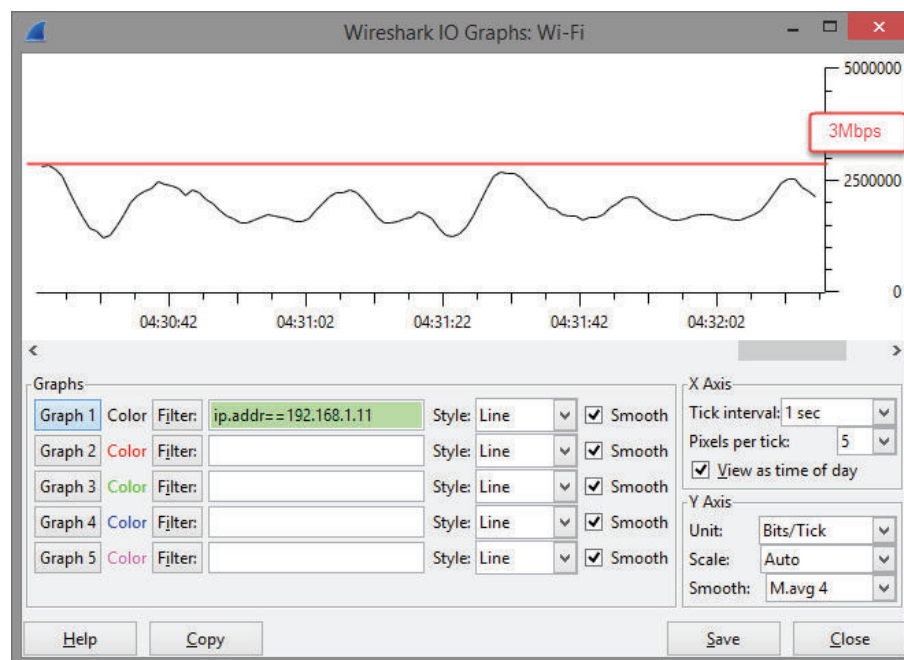
Para dimensionar la tasa de transferencia en un ambiente de producción se han tomado mediciones de las tasas de transferencia involucradas en la generación de llamadas concurrentes.

En la Figura 3.75 se muestra la tasa de transferencia utilizada en 10 llamadas concurrentes, de esta se puede definir que el punto más alto de la gráfica se encuentra cercano a los 1.6Mbps.



**Figura 3.75** Gráfica tasa de transferencia 10 llamadas

En la Figura 3.76 se indica la tasa de transferencia utilizada en 20 llamadas concurrentes, de esta se puede definir que el punto más alto de la gráfica se encuentra cercano a los 2.5Mbps.



**Figura 3.76** Gráfica tasa de transferencia 20 llamadas

De las mediciones realizadas se puede evidenciar que una buena aproximación de la tasa de transferencia involucrada en una llamada es de 155Kbps, con este valor se puede calcular la tasa de transferencia total necesaria para cualquier cantidad de llamadas simultaneas para un ambiente de producción.

### **3.5 REQUERIMIENTOS MÍNIMOS**

Los requerimientos mínimos serán considerados en un entorno donde se pueda garantizar el funcionamiento de la plataforma, mas no una cantidad específica de usuarios simultáneos conectados a la misma.

#### **3.5.1 CENTRAL TELEFÓNICA SIP**

El único requerimiento básico del sistema corresponde a la compatibilidad de la central telefónica con el protocolo SIP, puesto que Flashphoner es un softphone web que está basado en este protocolo. En cuanto a la cantidad de extensiones, debe permitir la creación de las extensiones que se requieran para usuarios operadores, más las de los clientes a los que se va a brindar el servicio.

La cantidad de troncales soportadas de igual forma depende de cada empresa, pero es necesaria por lo menos una troncal para poder redireccionar las llamadas telefónicas hacia los móviles en caso de no estar registrados ese momento mediante SIP.

La central telefónica debe permitir la creación de por lo menos 4 planes de marcado para cumplir con los requisitos de la implementación.

Las especificaciones técnicas dependen de los recursos y funcionalidades de cada central telefónica, estas son comercializadas en función de la cantidad de usuarios soportados.



### **3.5.2 SERVIDOR WEB**

Para el funcionamiento de la plataforma web y el chat de soporte es necesario un servidor web que tenga acceso a la red local y a internet para utilizar los servicios de Pusher y Parse que son basados en la Nube. Para una implementación básica debe contar con soporte para PHP 5.4 o superior y base de datos MySQL 5 o superior. Debe contener por lo menos una interfaz de red para comunicarse en la red local. El sistema operativo debe ser una distribución de Linux como Centos, Red Hat, Ubuntu o similares.

### **3.5.3 SERVIDOR DE MEDIOS MULTIMEDIA**

Este servidor debe tener conectividad con la central telefónica y con la red local para comunicarse con la aplicación cliente. Este servidor debe tener instalada una distribución de Linux de 64 bits como Centos, Red Hat, Ubuntu o similares y Java Development Kit 1.7 o superior.

Las especificaciones técnicas mínimas según el manual de FlashPhoner Web Call Server 4 son las siguientes:

- Memoria RAM 512 MB
- 1Gb de espacio libre en Disco
- 1 núcleo de procesador

### **3.5.4 CONEXIÓN A INTERNET**

La conexión a internet es necesaria para la implementación del sistema, pues hace uso de servicios como Pusher y Parse que están alojados en internet. El dimensionamiento de este recurso depende del tráfico generado por los usuarios, que para consumir los servicios alojados en internet es mínimo pero para generar y recibir llamadas en el caso de que el servidor este publicado en internet si es

considerable, por este motivo para los requerimientos mínimos solo se considera una conexión a internet y no una tasa de transferencia específica para el sistema.

### 3.5.5 CONEXIÓN TELEFONÍA PÚBLICA

En una configuración básica la comunicación de la central telefónica con la telefonía pública no es determinante ya que con el uso de la plataforma web las llamadas son entre extensiones, pero para habilitar funcionalidades adicionales como las llamadas a números externos desde la plataforma es necesaria la contratación de una línea telefónica que a través de un Gateway nos comunique con números externos.

### 3.5.6 LICENCIA FLASHPHONER

El softphone web Flashphoner tiene planes de licencias en función del tipo de cliente, las características incluidas y las llamadas simultáneas que se pueden realizar. La Tabla 3.13 ilustra las 2 licencias básicas ofrecidas por Flashphoner, la licencia simple es la versión mínima requerida que aunque tiene todas las características incluidas tiene sus limitaciones en cuanto a soporte, la cantidad de nodos y se debe incluir información de Flashphoner en cada implementación.

**Tabla 3.13** Licencias básicas Flashphoner

CARACTERÍSTICA	LICENCIA SIMPLE	LICENCIA CORPORATIVA
Foro Técnico	Si	Si
Soporte individual email o ticket	No	Si
Respuesta de Soporte Garantizado	No	Si, Siguiendo día laborable
Costo Licencia	\$999	Depende de características y llamadas simultaneas
Debe incluir marca de Flashphoner	Si	No
Nodos de Servidor	1	1-16
Características Incluidas	Todas	Limitadas según requerimientos
Número máximo de conexiones simultaneas y llamadas simultaneas	Ilimitado	Limitado según requerimientos

### 3.6 COSTOS DE LA IMPLEMENTACIÓN

Para este análisis se tomarán en cuenta todos los costos relacionados a la implementación del prototipo incluido hardware, software, honorarios por desarrollo y servicios en el lapso de un año.

La licencia de Flashphoner es indefinida solo en caso de requerir soporte adicional a los 3 meses es necesario la contratación de este servicio por lo cual se tomará en cuenta como un solo pago.

El costo de la conexión a internet para este sistema es crítico por lo cual se hizo un análisis para dimensionar las características del servicio y se lo incluirá en los costos para el lapso de un año. Los costos adicionales son rubros que se cubrieron durante el desarrollo del sistema. En la

Tabla 3.14 se muestra un resumen de los costos involucrados en la implementación.

**Tabla 3.14** Resumen de costos

<b>Hardware</b>		
Servidor IBM 3100 M4	\$1.200,00	Dólares
Memoria 4 GB	\$400,00	Dólares
<b>Costo Total</b>	<b>\$1.800,00</b>	<b>Dólares</b>
<b>Software</b>		
Licencia FlashPhoner	\$999,00	Dólares
<b>Costo Total</b>	<b>\$999,00</b>	<b>Dólares</b>
<b>Internet</b>		
Costo Mensual Conexión 1000 Mbps Compartición 1:1 con IP Pública	\$250,00	Dólares
Tiempo	12	Meses
<b>Costo Total</b>	<b>\$3.000,00</b>	
<b>Honorarios Desarrollador de Software</b>		
Remuneración Mensual Promedio Empresa Pública[23]	\$1.500,00	Dólares
Tiempo Estimado	5	Meses
<b>Costo Total</b>	<b>\$7.500,00</b>	<b>Dólares</b>
<b>Adicionales</b>		
Suscripción iOS Developer Program	\$100,00	Dólares
<b>Costo Total</b>	<b>\$100,00</b>	<b>Dólares</b>
<b>COSTO TOTAL IMPLEMENTACIÓN</b>	<b>\$13.399,00</b>	<b>Dólares</b>

## CAPÍTULO 4

### 4. CONCLUSIONES Y RECOMENDACIONES

#### 4.1 CONCLUSIONES

- I. El sistema integral del presente proyecto permite planificar los usuarios de cada departamento que brindarán soporte según la etapa en la cual se encuentre cada proyecto.
- II. Mediante la integración de un chat de soporte en la aplicación se facilita la comunicación del cliente con la empresa, dando una alternativa de contacto cuando los usuarios asignados a determinado proyecto no se encuentren disponibles, permitiendo comunicarse con otro usuario de un determinado departamento.
- III. La inclusión de un chat interno permite a los usuarios comunicarse entre ellos para intercambiar información, mientras se está brindando soporte a un cliente sea por voz o mediante el chat de soporte.
- IV. El uso de las nuevas características de HTML5 como Websockets en la aplicación del presente proyecto garantizan eficiencia en la transmisión de información de la comunicación cliente-servidor pues los beneficios de estas tecnologías son mayores cuando la cantidad de usuarios aumenta.
- V. La virtualización de servidores provee gran escalabilidad a los servidores permitiendo una administración centralizada para gestionar o aumentar sus recursos en caso de ser necesario, además de poderlos migrar fácilmente en una actualización del servidor base.

- VI. Existen diversidad de servicios como Pusher y Parse que proveen funcionalidades gratuitas y/o de bajo costo muy útiles para el desarrollo de proyectos, pues de esta manera el tiempo de implementación disminuye en gran medida facilitando la implementación y reduciendo costos significativamente.
- VII. La aplicación no requiere instalación de ningún complemento, funciona en los navegadores más comunes, facilitando el acceso a todos los clientes para promover el uso de las llamadas mediante la interfaz web y en consecuencia disminuir el tráfico en las interconexiones con la telefonía pública.
- VIII. La integración con aplicaciones de terceros como Webim que no proveen una API para este objetivo resulta compleja, es necesario un análisis a fondo del código fuente involucrado para que las modificaciones no alteren el normal funcionamiento de la misma y que las características queden acopladas óptimamente a la nueva aplicación.
- IX. La utilización de un framework MVC para el desarrollo de la aplicación facilita la creación de funcionalidades comunes en un sistema web como autenticación, seguridad, gestión de usuarios, administración de base de datos, entre otras reduciendo el tiempo de implementación de todo el proyecto.
- X. La aplicación del presente proyecto mediante la integración con la telefonía IP de una empresa tipo mejora la accesibilidad de los clientes con el soporte de la empresa, simplificando y agilizando la comunicación entre ambas partes para proveer un servicio eficaz y eficiente en respuesta a los requerimientos que se generen.
- XI. La implementación de una aplicación móvil reduce el tiempo de respuesta en la interacción con los clientes pues garantiza la continua disponibilidad de los operadores.

- XII. El desarrollo de aplicaciones Web mediante PHP provee una gran versatilidad en la selección del servidor que alojará la misma pues permiten ser instaladas en sistemas operativos Windows, Linux, OSX, FreeBSD, entre otros.
- XIII. El uso de WebRTC para llamadas de audio mejora el rendimiento y facilita la comunicación entre usuarios pues con HTML5 el soporte es nativo y anteriormente solo se podía realizar mediante complementos como Flash.
- XIV. El protocolo Websockets se encarga de enviar mensajes denominados ping/pong para mantener abierta la conexión, de esta forma las aplicaciones como el chat interno mantienen la lista de clientes conectados para interactuar con ellos.
- XV. El protocolo Websockets introduce una cabecera muy pequeña en comparación con HTTP como se pudo observar en el análisis realizado del protocolo, esto demuestra la eficiencia en la información transmitida por WebSocket vs HTTP.

## **4.2 RECOMENDACIONES**

- I. Para seleccionar aplicaciones basadas en las nuevas características de HTML5 como Websockets se recomienda aquellas que tienen mecanismos alternos que permitan una mayor compatibilidad en aquellos navegadores que no soportan todas las características.
- II. El desarrollo de aplicaciones que tienen como uno de sus objetivos alcanzar la mayor cantidad de personas, debe permitir la coexistencia en navegadores compatibles con HTML5 y aquellos que no tengan completa compatibilidad.

- III. La configuración del cortafuegos debe ser tomada en cuenta en un ambiente de producción para lograr una interconexión satisfactoriamente los diferentes servidores de la implementación.
- IV. Se debe realizar un constante mantenimiento de la aplicación de tal forma que se encuentre actualizada para garantizar el adecuado funcionamiento e integración con los servicios utilizados.
- V. Para un ambiente de producción se deben encontrar con acceso público las IPs asignadas a la central telefónica, el servidor web y el servidor multimedia, de forma que sean accesibles para los clientes.
- VI. Para un ambiente de producción la aplicación móvil debe ser publicada en el App Store de Apple pues de otra manera los usuarios no podrán instalar la aplicación en el móvil.
- VII. Con el aumento de usuarios se deben tomar en consideración los límites de los servicios Pusher y Parse y en caso de superarlos proceder a actualizar a un plan de pago que aumente las capacidades.
- VIII. Para escoger un framework de desarrollo se debe considerar el soporte, la documentación y se recomienda utilizar una versión LTS (Long Term Support) pues de esta manera se tiene un soporte extendido, evitando que la versión quede obsoleta en poco tiempo y tener que migrar la aplicación a una nueva versión.
- IX. Para la prueba de aplicaciones de iOS en dispositivos iPhone es necesario adquirir la Suscripción iOS Developer Program, sin esta no es posible probar las notificaciones push.
- X. Es recomendable utilizar servicios como Parse y Pusher en la implementación de aplicaciones debido a que disminuyen el tiempo de desarrollo significativamente y proveen una gran escalabilidad a las aplicaciones pues se puede aumentar las limitaciones contratando un servicio de pago.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] H. Sinnreich and A. B. Johnston, Internet Communications Using SIP, Wiley Publishing, Inc., 2006.
- [2] "The WebSocket Protocol - RFC 6455," [Online]. Available: [http://datatracker.ietf.org/doc/rfc6455/?include\\_text=1](http://datatracker.ietf.org/doc/rfc6455/?include_text=1). [Accessed 25 Agosto 2014].
- [3] V. Wang, F. Salim and P. Moskovits, The Definitive Guide to HTML5 WebSocket, New York: Apress, 2013.
- [4] "WebSocket.org - What is WebSocket?," [Online]. Available: <http://www.websocket.org/>. [Accessed 15 Septiembre 2014].
- [5] "Heise Developer," 2015. [Online]. Available: <http://www.heise.de/developer/artikel/WebSocket-Annaeherung-an-Echtzeit-im-Web-1260189.html?artikelseite=2>.
- [6] P. Lubbers and F. Greco, "HTML5 Web Sockets: A Quantum Leap in Scalability for the Web," [Online]. Available: <https://www.websocket.org/quantum.html>. [Accessed 25 Agosto 2014].
- [7] "WebRTC API," [Online]. Available: <https://docs.webplatform.org/wiki/apis/webrtc>. [Accessed 28 Agosto 2014].
- [8] "WebRTC," [Online]. Available: <http://www.webrtc.org/home>. [Accessed 25 08 2014].
- [9] "Techtarget - Softphone," 9 Enero 2015. [Online]. Available: <http://searchunifiedcommunications.techtarget.com/definition/softphone>.
- [10] "Pusher," [Online]. Available: <https://pusher.com>.  
]
- [11] "Parse," [Online]. Available: <https://parse.com/>.  
]
- [12] "PHP: Hypertext Preprocessor," [Online]. Available: <http://php.net/>.  
] [Accessed 10 Agosto 2014].
- [13] "W3Techs - World Wide Web Technology Surveys," [Online]. Available: <http://w3techs.com/>. [Accessed 10 Enero 2015].
- [14] C. Pitt, Pro PHP MVC, New York: Apress, 2012.  
]
- [15] "Symfony," [Online]. Available: <http://symfony.com/>. [Accessed 29 Agosto  
] 2014].
- [16] F. Potencier, "What is Symfony," 25 Octubre 2011. [Online]. Available: <http://fabien.potencier.org/article/49/what-is-symfony2>.
- [17] Doctrine Project, "Welcome to the Doctrine Project," [Online]. Available: <http://www.doctrine-project.org/>. [Accessed 28 Agosto 2014].



- [18 "Acerca de Javascript - Javascript MDN," [Online]. Available:  
] [https://developer.mozilla.org/es/docs/JavaScript/Acerca\\_de\\_JavaScript](https://developer.mozilla.org/es/docs/JavaScript/Acerca_de_JavaScript).  
[Accessed 21 Septiembre 2014].
- [19 "JavaScript Tutorial," [Online]. Available: <http://www.w3schools.com/js/>.  
] [Accessed 21 Septiembre 2014].
- [20 "jQuery," [Online]. Available: <http://jquery.com/>. [Accessed 21 Septiembre  
] 2014].
- [21 "Redmonk," [Online]. Available:  
] <http://redmonk.com/sogrady/category/programming-languages/>. [Accessed  
Enero 2015].
- [22 Flashphoner, "Web Call Server 4 - Admin Guide," [Online]. Available:  
] [http://flashphoner.com/docs/wcs4/wcs\\_docs/pdf/en/wcs-admin-guide.pdf](http://flashphoner.com/docs/wcs4/wcs_docs/pdf/en/wcs-admin-guide.pdf).
- [23 "CFN - Remuneración mensual por cargo," [Online]. Available:  
] [http://www.cfn.fin.ec/index.php?option=com\\_content&view=article&id=80&Itemid=389](http://www.cfn.fin.ec/index.php?option=com_content&view=article&id=80&Itemid=389).
- [24 R. Rai, Birmingham: Packt Publishing Ltd., 2013.  
]

**ANEXOS**

## **ANEXO A**

### **IMPLEMENTACIÓN VMWARE VSPHERE HYPERVISOR 5**

## Implementación VMware vSphere Hypervisor 5

Como primer paso para adquirir una licencia de VMware vSphere es necesario crear una cuenta en <https://my.vmware.com>, con lo cual se obtiene una licencia gratuita de VMware vSphere Hypervisor en su versión 5, se procede con la descarga de la imagen ISO del CD de Instalación para el servidor y con la aplicación cliente para la administración que se encuentra disponible para Windows.

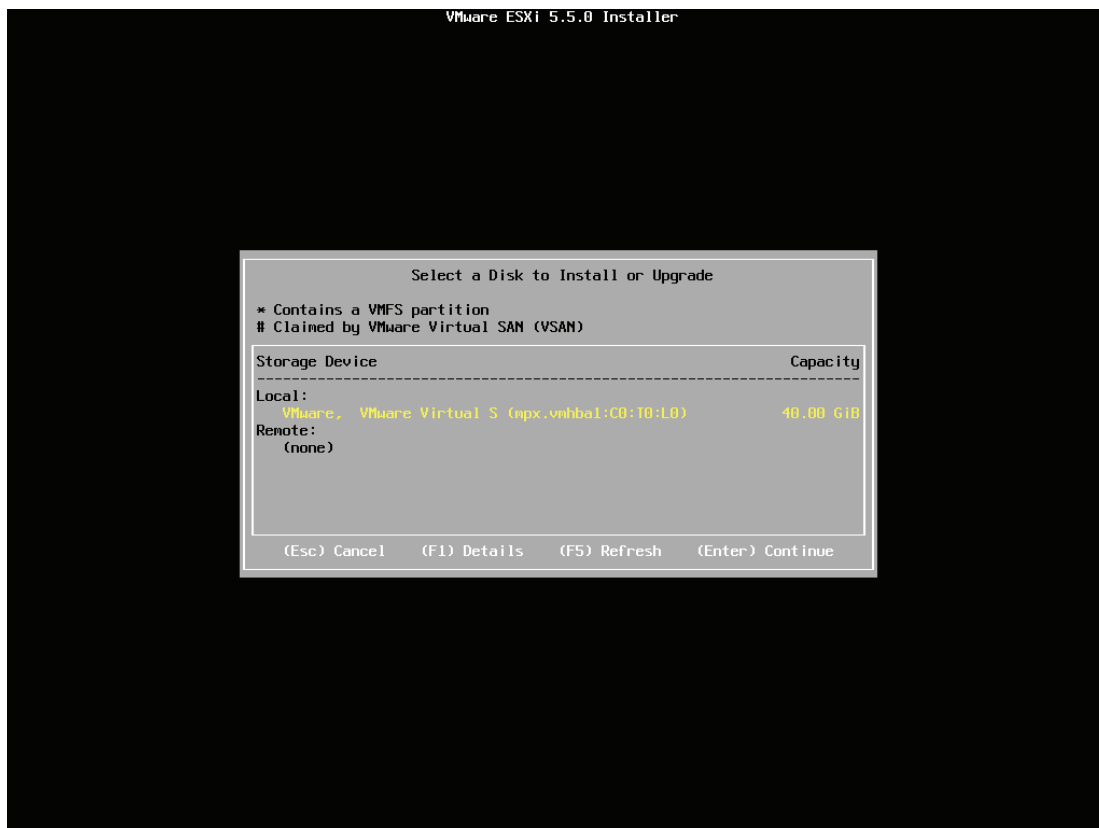
### *Instalación y Configuración VMware vSphere Hypervisor 5*

Para la instalación se debe ejecutar el CD de Instalación de VMware vSphere en el servidor. Al ejecutarse correctamente mostrará una pantalla similar a la Figura 1 en la cual muestra la versión que va a ser instalada, en este caso la versión 5.5.0 y la carga de archivos en la barra inferior.



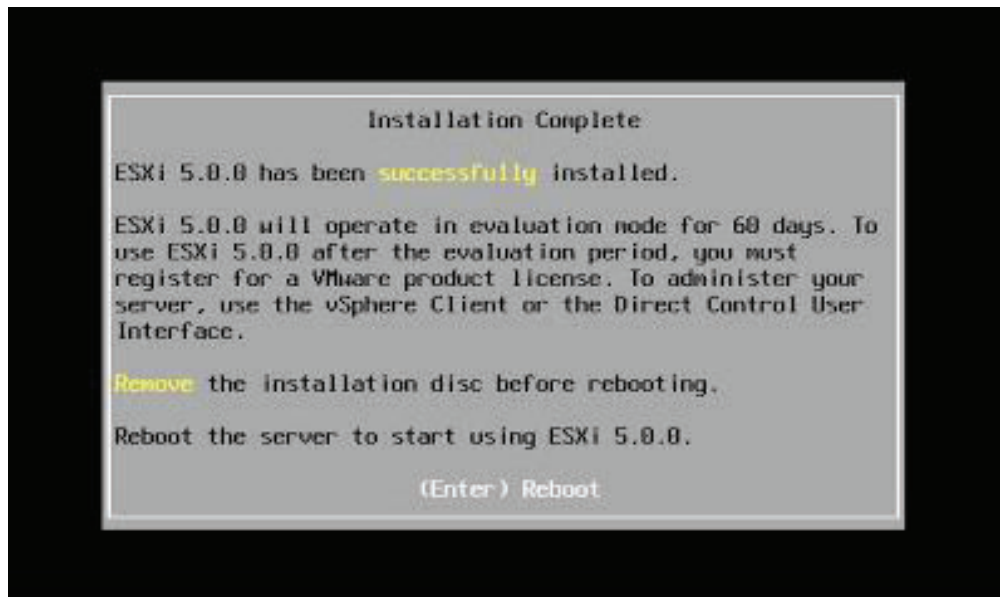
**Figura 1** Pantalla Inicial Instalación VMware vSphere Hypervisor 5

A continuación se mostrarán pantallas informativas sobre las versiones de sistemas operativos compatibles y sobre la aceptación del acuerdo de licencia. El siguiente paso es escoger el disco duro en el cual se instalará el sistema y mostrará una pantalla similar a la Figura 2 en donde se seleccionará el único disco que se encuentra en el servidor físico.



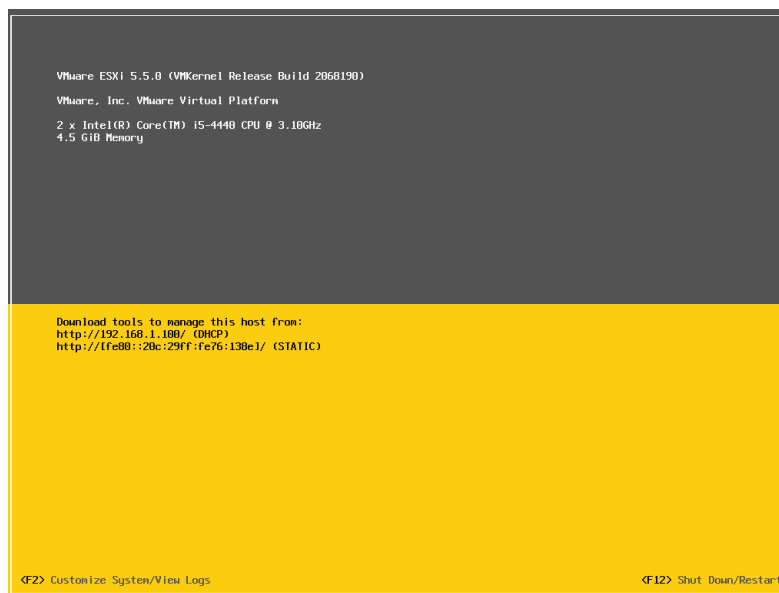
**Figura 2** Selección de Disco Duro para Instalación VMware vSphere Hypervisor

Se procede con la creación de una contraseña para el usuario *root* del sistema y la confirmación de los parámetros de instalación. Al finalizar se mostrará en la pantalla que la instalación se ha completado y pedirá que se remueva el CD de Instalación y se reinicie el sistema como indica la Figura 3.



**Figura 3** Finalización Instalación VMware vSphere Hypervisor 5

Cuando el sistema complete su carga se mostrará en pantalla la información correspondiente al procesador y memoria del equipo además de la IP asociada para su administración como indica la Figura 4. Desde esta pantalla se debe ingresar a la interfaz de administración presionando la tecla F2 para hacer el cambio de IP y colocar con una dirección estática para facilitar su administración.

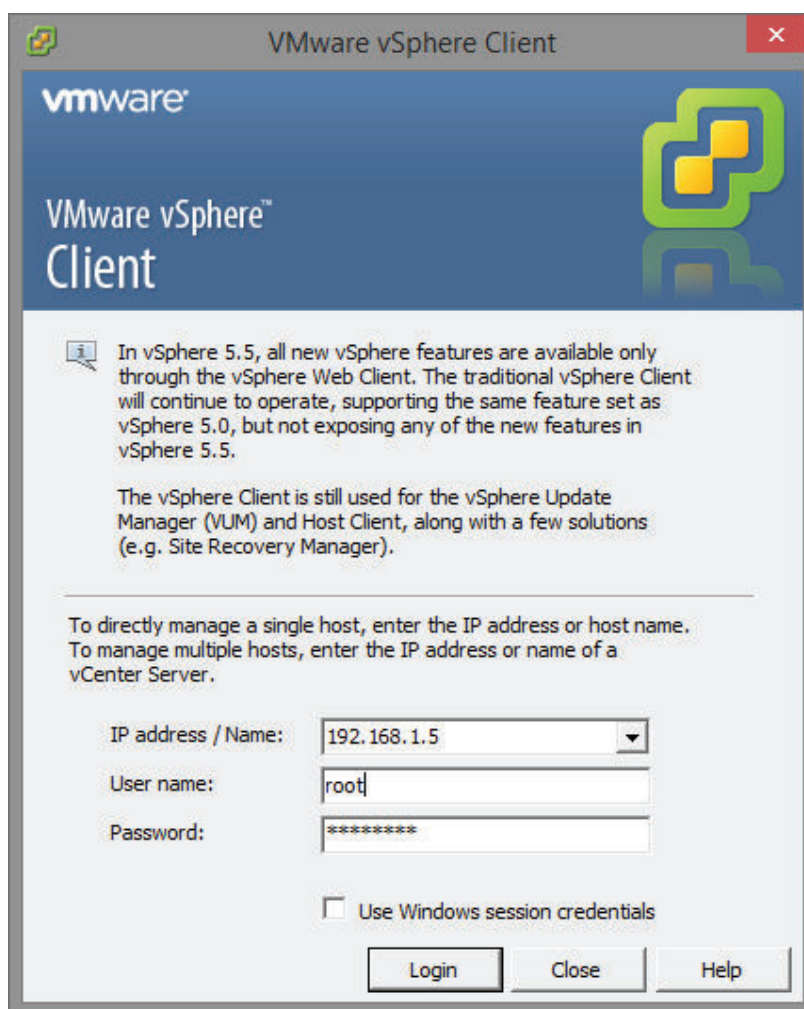


**Figura 4** Pantalla Inicial VMware vSphere Hypervisor

### *Instalación y Configuración VMware vSphere Client*

Para la instalación del cliente se ejecuta en un entorno Windows el archivo descargado desde la página web denominado VMware vSphere Client. Aquí aparecerá un asistente de instalación que guiará la instalación mostrando la Patente para el Usuario Final y los términos del Acuerdo de Licencia. A continuación se debe introducir un Nombre de Usuario y el Nombre de la Organización, se indica la carpeta destino y se finaliza con la instalación.

Al iniciar la aplicación como se muestra en la Figura 5, pedirá configurar la información correspondiente al servidor, para esto se introduce la IP asignada al servidor como también el usuario y contraseña correspondiente.



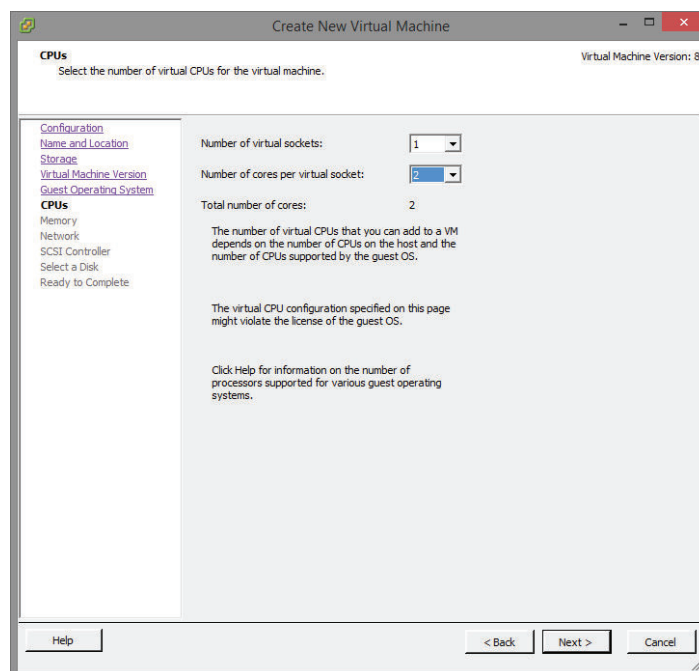
**Figura 5** Ingreso VMware vSphere Client

Luego de presionar el botón *Login* empezará la comunicación con el servidor y mostrará la ventana de administración de las máquinas virtuales que se encuentran en el servidor.

Se debe agregar el número de licencia que fue provisto por VMWare para activar las características de la versión gratuita, pues por defecto tiene una licencia de prueba, eso se lo debe realizar en la configuración del servidor.

### *Ejemplo Creación Máquina Virtual*

En la aplicación para administración de VMware vSphere se debe dar clic derecho sobre el icono de la máquina física y seleccionar *New Virtual Machine*, se mostrará una ventana para escoger el tipo de configuración y para este caso se debe seleccionar *Custom*. A continuación definiremos el nombre de la Máquina virtual, el lugar de almacenamiento y el sistema operativo. Una de las opciones más importantes al momento de configurar la máquina virtual es la cantidad de *cores* o núcleos que como se muestra en la Figura 6.

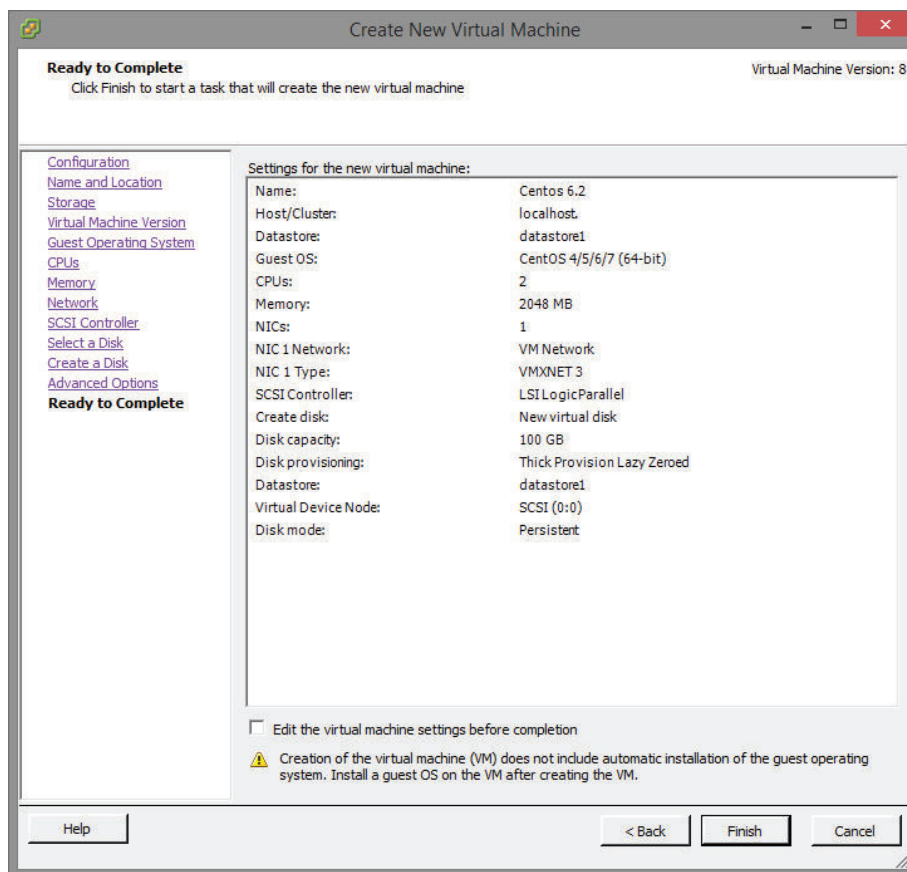


**Figura 6** Configuración CPU Servidor Web Virtual



Para continuar con la creación se define la cantidad de memoria RAM designada para el servidor virtual, el controlador SCSI que seleccionaremos el que se encuentra por defecto.

En la siguiente pantalla se debe seleccionar *Create new virtual disk* y se define el tamaño del disco duro según sea necesario. Para finalizar se mostrará un resumen de todas las configuraciones como se muestra en la Figura 3.1.



**Figura 7** Resumen configuraciones Servidor Virtual

## **ANEXO B**

### **MANUAL DEL USUARIO SISTEMA WEB INTEGRAL SOPORTE**



# Sistema Web Integral Soporte

Manual de Usuario

Autor: Eduardo Gallegos

# 1 Introducción

---

Este manual tiene como objetivo guiar a los usuarios para gestionar y utilizar las funcionalidades del sistema. Cada una de las funcionalidades es restringida según el rol del usuario autenticado por este motivo se indicará si una determinada funcionalidad tiene restricciones para un rol específico. Los roles con los cuales cuenta el sistema son los siguientes:

- ✓ Administrador
- ✓ Jefe de Departamento
- ✓ Usuario o Empleado

## 2 Acceso

---

Para el acceso a la aplicación se debe dirigir mediante un navegador a la siguiente url:

*http://[dirección\_del\_servidor]/Support/web/login*

En esta url se mostrará el recuadro de la Figura 1 que nos permite ingresar el usuario y contraseña asignados previamente por el Administrador del sistema.



*Figura 1 - Acceso a la Plataforma*



La casilla **Recordar** permite a los usuarios que su información de acceso quede almacenada por 1 hora aunque la ventana sea cerrada. De esta manera no tendrá que ingresar sus datos de autenticación cada vez que ingrese a la aplicación.

## 3 Perfil de Usuario

---

Luego de que el usuario acceda a la plataforma se mostrará una pantalla con el perfil del usuario como se muestra en la Figura 2 que cuenta las siguientes secciones:

1. Menú
2. Información del Usuario
3. Información de la Empresa
4. Información de Proyectos

## 5. Chat Interno

The screenshot displays the user profile page for 'Eduardo Gallegos'. The page is divided into several sections:

- Perfil Actual:** Shows user details such as 'Usuario: admin', 'Nombre: Eduardo', 'Apellido: Gallegos', and contact information like 'Tif. Principal: 3815930' and 'Email: edugallegos@gmail.com'.
- Datos de la Empresa:** Shows company information including 'Nombre: Servihelp', 'RUC: 1001123155004', and 'Email Institucional: edugallegos@gmail.com'.
- Proyectos de admin:** A table with columns 'Proyecto', 'Empresa', and 'Acción'. It shows one record for 'Ecuador' under 'Servihelp' with a 'Detalle' button. A search bar and pagination controls are also present.
- Dropdown Menu:** A list of users is shown, with 'Alan Endara' selected.

Figura 2 Perfil del Usuario

**Menú**

El menú contiene enlaces que permite acceder a los diferentes módulos de la plataforma que es restringido según el rol del usuario.

- *Departamentos* [Administrador, Jefe de Departamento]
  - Listado de Departamentos [Administrador, Jefe de Departamento]
  - Nuevo Departamento [Administrador]
- *Proyectos* [Todos]
  - Listado de Proyectos [Todos]
- *Empresa* [Administrador]
  - Listado de Empresas [Administrador]
  - Nueva Empresa [Administrador]
- *Llamadas* [Todos]
  - Marcación por Teclado [Todos]
  - Registro de Llamadas [Administrador, Jefe de Departamento]
  - Registro Llamadas por Usuario [Administrador, Jefe de Departamento]
- *Mensajes* [Todos]
  - Activar Chat [Todos]
- *Sistema* [Administrador]
  - Administración de Usuario [Administrador]
  - Configuraciones [Administrador]

- {Nombre de Usuario} [Todos]
  - Perfil [Todos]
  - Desconectar [Todos]
  - Salir [Todos]

### Información del Usuario

En esta sección se muestra toda la información del usuario actual, entre estos datos se encuentra el nombre, apellido, extensión, departamento, teléfonos y email.

### Información de la Empresa

En esta sección se muestra la información de la Empresa a la cual pertenece el usuario.

### Información de Proyectos

En esta sección se muestran los proyectos que se encuentran asignados en alguna etapa al usuario.

### Chat Interno

Esta sección contiene un chat interno que permite comunicarse y saber el estado de los otros usuarios de la plataforma.

## 4 Departamentos

Este módulo permite gestionar las diferentes áreas o departamentos de una empresa para que los usuarios sean asociados a cada uno de ellos.

### Listado de Departamentos



Esta sección es restringida para usuarios con roles Administrador y Jefe de Departamentos

Para acceder a esta sección se lo realiza mediante el menú *Departamentos* submenú *Listado de Departamentos* como se muestra en la Figura 3.

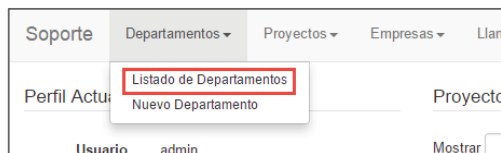


Figura 3 Enlace Listado de Departamentos

En la Figura se muestra el listado de los departamentos creados. Este listado permite realizar una búsqueda por su nombre. Los botones de acciones dependen de las restricciones del rol del usuario y se revisaran en cada acción.

Listado de Departamentos		
Mostrar	10 registros	Buscar: <input type="text"/>
Nombre	Observación	Acciones
Administrativo	Departamento Administrativo	Ver Editar
Desarrollo	Desarrollo	Ver Editar
RRHH	RRHH	Ver Editar
Ventas	Departamento de Ventas	Ver Editar
Mostrando registros del 1 al 4 de un total de 4 registros		Anterior 1 Siguiente
<a href="#">Crear Nuevo Departamento</a>		

Figura 8 Listado de Departamentos

## Creación de Departamentos



Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el menú *Departamentos* submenú *Nuevo Departamento* como se muestra en la Figura 4 o mediante el botón *Crear Nuevo Departamento* que se muestra en el listado de departamentos.

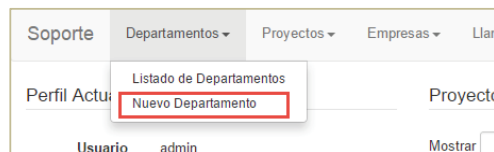


Figura 4 Enlace Nuevo Departamento

En la Figura 5 se muestra el formulario para la creación de un departamento en el cual se deben ingresar los siguientes datos:

**Nombre:** Nombre del departamento

**Observación:** Descripción del departamento

Creación Departamento	
<b>Nombre</b>	<input type="text"/>
<b>Observación</b>	<input type="text"/>
<a href="#">Crear</a>	<a href="#">Regresar a Lista</a>

Figura 5 Formulario Creación Departamento

Luego de ingresar los datos se procede a presionar el botón *Crear* y el nuevo departamento será creado.

## Editar Departamento

 Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el botón *Editar* en las acciones del listado de departamentos que se muestra en la Figura 6.



Nombre	Observación	Acciones
Administrativo	Departamento Administrativo	Ver Editar
Desarrollo	Desarrollo	Ver Editar
RRHH	RRHH	Ver Editar
Ventas	Departamento de Ventas	Ver Editar

Mostrando registros del 1 al 4 de un total de 4 registros

Anterior 1 Siguiete

Crear Nuevo Departamento

Figura 6 Acción Editar Departamento

En la Figura 7 sección 1 se muestra el formulario para la edición de un departamento en el cual se pueden modificar los siguientes datos:

**Nombre:** Nombre del departamento

**Observación:** Descripción del departamento

En la sección 2 se puede añadir o quitar usuarios del departamento, en este solo se podrán asociar usuarios que no tengan un departamento.



**Editar Departamento**

Nombre: Administrativo

Observación: Departamento Administrativo

Listado de Usuarios

Mostrar: 10 registros

Nombre	Apellido	Extensión
Ningún dato disponible en esta tabla		

Mostrando registros del 0 al 0 de un total de 0 registros

Añadir Usuarios Quitar Seleccionados

Editar Regresar Eliminar

Figura 7 Formulario Editar Departamento

Luego de realizar las modificaciones requeridas se procede a presionar el botón *Editar* y el departamento será actualizado.

## Ver Departamento

 Esta sección es restringida para usuarios con roles Administrador y Jefe de Departamento



Para acceder a esta sección se lo realiza mediante el botón *Ver* en las acciones del listado de departamentos que se muestra en la Figura 8.

Nombre	Observación	Acciones
Administrativo	Departamento Administrativo	Ver Editar
Desarrollo	Desarrollo	Ver Editar
RRHH	RRHH	Ver Editar
Ventas	Departamento de Ventas	Ver Editar

Mostrando registros del 1 al 4 de un total de 4 registros

Anterior 1 Siguiente

Crear Nuevo Departamento

Figura 8 Acción Ver Departamento

En la Figura9 sección 1 se muestra la información de un departamento.

En la sección 2 se muestran los usuarios asociados al departamento.

Información del Departamento

Nombre	Administrativo
Observación	Departamento Administrativo

Listado de Usuarios

Nombre	Apellido	Extensión
Ningún dato disponible en esta tabla		

Mostrando registros del 0 al 0 de un total de 0 registros

Anterior Siguiente

Regresar Editar Eliminar

Figura 9 Formulario Ver Departamento

## Eliminar Departamento



Esta sección es restringida para usuarios con rol Administrador

Para eliminar un departamento se lo realiza mediante el botón *Eliminar* que se muestra en el formulario *Editar Departamento* de cada departamentocomo indica la Figura 10.

Figura 10 Enlace Editar Departamento

## 5 Empresas

Este módulo permite gestionar las diferentes empresas que interactúan en el sistema, la primera empresa creada corresponde a la empresa propietaria.

### Listado de Empresas



Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el menú *Empresa* submenú *Listado de Empresas* como se muestra en la Figura 11.

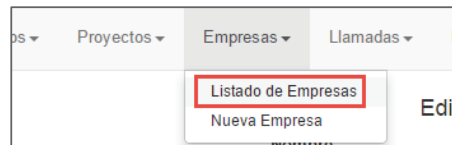


Figura 11 Enlace Listado de Empresas

En la Figura12 se muestra el listado de las empresas creadas. Este listado permite realizar una búsqueda por su nombre. Los botones de acciones dependen de las restricciones del rol del usuario y se revisaran en cada acción.

Listado de Empresas						
Mostrar	10	registros	Buscar: <input type="text"/>			
Nombre ▲	RUC	Email	Teléfono	Dirección	Página Web	Acciones
GYP SAN	1717130164	admin@gypsan.com	56591423		www.gypsan.com	Ver Editar
MAGAP	1001123155001	soporte@magap.com		Eloy Alfaro y Amazonas		Ver Editar
Sinohydro	48571216665	synohydro@prefd.ch				Ver Editar
TESIS EPN	4855656565	eprn@gmail.com	2147483	Alfon	www.eprn.edu.ec	Ver Editar

Mostrando registros del 1 al 4 de un total de 4 registros

Anterior 1 Siguiente

[Crear Nueva Empresa](#)

Figura 12 Listado de Empresas

## Creación de Empresas



Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el menú *Empresas* submenú *Nueva Empresa* como se muestra en la Figura13 o mediante el botón *Crear Nueva Empresa* que se muestra en el listado de empresas.

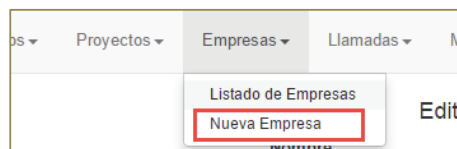


Figura 13 Enlace Nueva Empresa

En la Figura14 se muestra el formulario para la creación de una empresa en el cual se deben ingresar los siguientes datos:

**Nombre:** Nombre de la empresa

**RUC:** RUC de la empresa

**Email:** Email principal de la empresa

**Teléfono:** Teléfono principal de la empresa

**Dirección:** Dirección Matriz de la empresa

**Página Web:** Página Web de la empresa



Creación de Empresa

Nombre

RUC

Email

Teléfono

Dirección

Página Web

Figura 14 Formulario Creación Empresa

Luego de ingresar los datos se procede a presionar el botón Crear y la nueva empresa será creada.

## Editar Empresa



Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el botón *Editar* en las acciones del listado de empresas que se muestra en la Figura 15.



Listado de Empresas

Mostrar 10 registros

Nombre	RUC	Email	Teléfono	Dirección	Página Web	Acciones
GYPSAN	1717130164	admin@gypsan.com	56591423		www.gypsan.com	Ver Editar
MAGAP	1001123155001	soporte@magap.com		Eloy Alfaro y Amazonas		Ver Editar
Sinohydro	48571216665	synohydro@prefd.ch				Ver Editar
TESIS EPN	4855656565	epn@gmail.com	2147483	Alfon	www.epn.edu.ec	Ver Editar

Mostrando registros del 1 al 4 de un total de 4 registros

Anterior 1 Siguiente

Figura 15 Acción Editar Empresa

En la Figura16 se muestra el formulario para la edición de una empresa en el cual se pueden modificar los datos de la empresa que se indicaron en la creación.

Luego de realizar las modificaciones requeridas se procede a presionar el botón *Actualizar* y la empresa será actualizada.

**Editar Empresa**

**Nombre**  
GYPSAN

**RUC**  
1717130164

**Email**  
admin@gypsan.com

**Teléfono**  
56591423

**Dirección**

**Página Web**  
www.gypsan.com

*Figura 16 Formulario Editar Empresa*

## Ver Empresa

 Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el botón *Ver* en las acciones del listado de empresas que se muestra en la Figura 17.

**Listado de Empresas**

Mostrar  registros Buscar:

Nombre	RUC	Email	Teléfono	Dirección	Página Web	Acciones
GYPSAN	1717130164	admin@gypsan.com	56591423		www.gypsan.com	<input type="button" value="Ver"/> <input type="button" value="Editar"/>
MAGAP	1001123155001	soporte@magap.com		Eloy Alfaro y Amazonas		<input type="button" value="Ver"/> <input type="button" value="Editar"/>
Sinohydro	48571216665	synohydro@prefd.ch				<input type="button" value="Ver"/> <input type="button" value="Editar"/>
TESIS EPN	485566565	epn@gmail.com	2147483	Alfon	www.epn.edu.ec	<input type="button" value="Ver"/> <input type="button" value="Editar"/>

Mostrando registros del 1 al 4 de un total de 4 registros

*Figura 17 Acción Ver Empresa*

En la Figura 18 se muestra la información de una empresa.

**Información de la Empresa**

<b>Nombre</b>	GYPSAN
<b>RUC</b>	1717130164
<b>Email</b>	admin@gypsan.com
<b>Teléfono</b>	56591423
<b>Dirección</b>	
<b>Página Web</b>	www.gypsan.com

*Figura 18 Formulario Ver Empresa*

## Eliminar Empresa

 Esta sección es restringida para usuarios con rol Administrador

Para eliminar una empresa se lo realiza mediante el botón *Eliminar* que se muestra en el formulario *Editar Empresa* de cada empresa como indica la Figura 19.

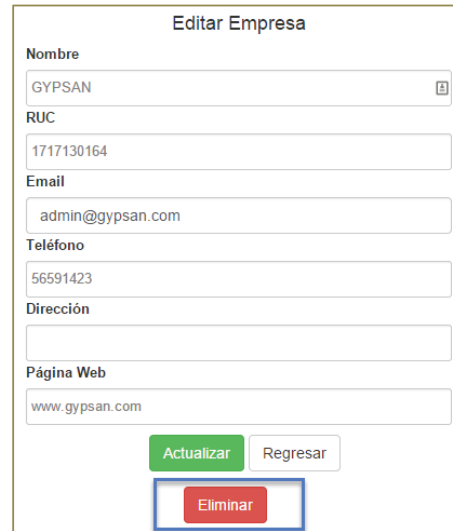



Figura 19 Eliminar Empresa

## 6 Proyectos

Este módulo permite gestionar proyectos, las etapas y las asignaciones de usuarios.

### Listado de Proyectos

 Esta sección es restringida para usuarios con rol Administrador, Jefe de Departamento, Usuario

Para acceder a esta sección se lo realiza mediante el menú *Proyectos* submenú *Listado de Proyectos* como se muestra en la Figura 20.

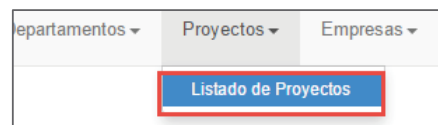


Figura 20 Enlace Listado de Proyectos

En la Figura 21 se muestra el listado de proyectos creados. Este listado permite realizar una búsqueda por su nombre. Los botones de acciones dependen de las restricciones del rol del usuario y se revisaran en cada acción.

Lista de Proyectos					
Mostrar	10	registros	Buscar: <input type="text"/>		
Nombre	Fecha Inicio	Fecha Fin	Empresa	Acciones	
Ecuador	01-02-2015	01-03-2015	TESIS EPN	Ver	Editar Etapas
EPN Tesis	05-12-2014	30-04-2015	GYPSAN	Ver	Editar Etapas
Proyecto Testing	01-12-2014	31-12-2014	MAGAP	Ver	Editar Etapas
GPS Nivel Nacional			MAGAP	Ver	Editar Etapas

Mostrando registros del 1 al 4 de un total de 4 registros

Anterior 1 Siguiente

[Crear Nuevo Proyecto](#)

Figura 21 Listado de Proyectos

## Creación de Proyectos



Esta sección es restringida para usuarios con rol Administrador o Jefe de Departamento

Para acceder a esta sección se lo realiza mediante el botón *Crear Nuevo Proyecto* que se muestra en el listado de proyectos en la Figura 22.

Lista de Proyectos					
Mostrar	10	registros	Buscar: <input type="text"/>		
Nombre	Fecha Inicio	Fecha Fin	Empresa	Acciones	
Ecuador	01-02-2015	01-03-2015	TESIS EPN	Ver	Editar Etapas
EPN Tesis	05-12-2014	30-04-2015	GYPSAN	Ver	Editar Etapas
Proyecto Testing	01-12-2014	31-12-2014	MAGAP	Ver	Editar Etapas
GPS Nivel Nacional			MAGAP	Ver	Editar Etapas

Mostrando registros del 1 al 4 de un total de 4 registros

Anterior 1 Siguiente

[Crear Nuevo Proyecto](#)

Figura 22 Acción Crear Nuevo Proyecto

En la Figura 23 se muestra el formulario para la creación de una empresa en el cual se deben ingresar los siguientes datos:

**Nombre:** Nombre del proyecto

**Descripción:** Descripción del proyecto

**Fecha Inicio:** Fecha inicial de la planificación del proyecto

**Fecha Fin:** Fecha final de la planificación del proyecto (*La fecha final no puede ser menor a la fecha inicial*)

**Empresa:** Empresa a la cual pertenece el proyecto



**Creación de Proyecto**

**Nombre**

**Descripción**

**Fecha Inicio**

**Fecha Fin**

**Empresa**

Figura 23 Formulario Creación Proyecto


Luego de ingresar los datos se procede a presionar el botón Crear y el proyecto será creado.

## Editar Proyecto



Esta sección es restringida para usuarios con rol Administrador o Jefe de Departamento

Para acceder a esta sección se lo realiza mediante el botón *Editar* en las acciones del listado de proyectos que se muestra en la Figura 24.



Lista de Proyectos

Mostrar 10 registros      Buscar:

Nombre	Fecha Inicio	Fecha Fin	Empresa	Acciones
Ecuador	01-02-2015	01-03-2015	TESIS EPN	Ver <input type="button" value="Editar"/> <input type="button" value="Etapas"/>
EPN Tesis	05-12-2014	30-04-2015	GYP SAN	Ver <input type="button" value="Editar"/> <input type="button" value="Etapas"/>
Proyecto Testing	01-12-2014	31-12-2014	MAGAP	Ver <input type="button" value="Editar"/> <input type="button" value="Etapas"/>
GPS Nivel Nacional			MAGAP	Ver <input type="button" value="Editar"/> <input type="button" value="Etapas"/>

Mostrando registros del 1 al 4 de un total de 4 registros

Figura 24 Acción Editar Proyecto

En la Figura 25 se muestra el formulario para la edición de un proyecto en el cual se pueden modificar los datos del proyecto que se encuentran a continuación:

**Nombre:** Nombre del proyecto *(El nombre debe ser único)*

**Descripción:** Descripción del proyecto

**Fecha Inicio:** Fecha inicial de la planificación del proyecto *(No debe interferir con las fechas de las etapas)*

**Fecha Fin:** Fecha final de la planificación del proyecto *(La fecha final no puede ser menor a la fecha inicial) (No debe interferir con las fechas de las etapas)*

**Empresa:** Empresa a la cual pertenece el proyecto



Luego de realizar las modificaciones requeridas se procede a presionar el botón *Actualizar* y el proyecto será actualizado.

### Editar Proyecto

**Nombre**

**Descripción**

**Fecha Inicio**

**Fecha Fin**

**Empresa**

Figura 25 Formulario Editar Proyecto

## Ver Proyecto



Esta sección es restringida para usuarios con rol Administrador, Jefe de Departamento o Usuario

Para acceder a esta sección se lo realiza mediante el botón *Ver* en las acciones del listado de proyectos que se muestra en la Figura 26.

Lista de Proyectos					
Mostrar <input type="text" value="10"/> registros			Buscar: <input type="text"/>		
Nombre	Fecha Inicio	Fecha Fin	Empresa	Acciones	
Ecuador	01-02-2015	01-03-2015	TESIS EPN	Ver	Editar
EPN Tesis	05-12-2014	30-04-2015	GYPSAN	Ver	Editar
Proyecto Testing	01-12-2014	31-12-2014	MAGAP	Ver	Editar
GPS Nivel Nacional			MAGAP	Ver	Editar

Mostrando registros del 1 al 4 de un total de 4 registros

Figura 26 Acción Ver Proyectos

En la Figura 27 se muestra la información de un proyecto.

Proyecto	
Nombre	Ecuador
Empresa	TESIS EPN
Descripción	Proyecto sobre ecuador
Fecha Inicio	01-02-2015
Fecha Fin	01-03-2015

Figura 27 Formulario Ver Proyecto

## Eliminar Proyecto



Esta sección es restringida para usuarios con rol Administrador o Jefe de Departamento

Para eliminar una empresa se lo realiza mediante el botón *Eliminar* que se muestra en el formulario *Editar Proyecto* de cada proyectocomo indica la Figura 28.

Editar Proyecto	
Nombre	<input type="text" value="Ecuador"/>
Descripción	<input type="text"/>
Fecha Inicio	<input type="text" value="01-02-2015"/>
Fecha Fin	<input type="text" value="01-03-2015"/>
Empresa	<input type="text" value="TESIS EPN"/>

Figura 28 Eliminar Proyecto

## Etapas de Proyecto



Esta sección es restringida para usuarios con rol Administrador, Jefe de Departamento, Usuario

Para acceder a esta sección se lo realiza mediante el *Listado de Proyectos* mediante el botón *Etapas* que se muestra en la Figura 29.

Lista de Proyectos					
Mostrar 10 registros			Buscar: <input type="text"/>		
Nombre	Fecha Inicio	Fecha Fin	Empresa	Acciones	
Ecuador	01-02-2015	01-03-2015	TESIS EPN	Ver	Etapas
EPN Tesis	05-12-2014	30-04-2015	GYPSAN	Ver	Etapas
Proyecto Testing	01-12-2014	31-12-2014	MAGAP	Ver	Etapas
GPS Nivel Nacional			MAGAP	Ver	Etapas

Mostrando registros del 1 al 4 de un total de 4 registros

Anterior 1 Siguiente

Crear Nuevo Proyecto

Figura 29 Botón Etapas de Proyecto

En la Figura30 se muestra el listado de Etapas creadas en el proyecto seleccionado previamente. Este listado muestra la información de cada etapa. Los botones de acciones dependen de las restricciones del rol del usuario y se revisaran en cada acción.

Proyecto EPN Tesis					
Etapas					
Mostrar 10 registros			Buscar: <input type="text"/>		
Nombre	Fecha Inicio	Fecha Fin	Acciones		
Etapa Final	02-03-2015	31-03-2015	Detalle	Editar	Ver
Etapa Adicional	01-04-2015	30-04-2015	Detalle	Editar	Ver
Etapa Única	05-12-2014	01-03-2015	Detalle	Editar	Ver

Mostrando registros del 1 al 3 de un total de 3 registros

Anterior 1 Siguiente

Nueva Etapa Listado Proyectos

Figura 30 Listado de Etapas de Proyecto

## Creación de Etapas



Esta sección es restringida para usuarios con rol Administrador o Jefe de Departamento

Para acceder a esta sección se lo realiza mediante el botón *Nueva Etapa* que se muestra en el listado de etapas del proyecto como se indica en la Figura 31.

Proyecto EPN Tesis					
Etapas					
Mostrar 10 registros			Buscar: <input type="text"/>		
Nombre	Fecha Inicio	Fecha Fin	Acciones		
Etapa Final	02-03-2015	31-03-2015	Detalle	Editar	Ver
Etapa Adicional	01-04-2015	30-04-2015	Detalle	Editar	Ver
Etapa Única	05-12-2014	01-03-2015	Detalle	Editar	Ver

Mostrando registros del 1 al 3 de un total de 3 registros

Anterior 1 Siguiente

Nueva Etapa Listado Proyectos

Figura 31 Acción Crear Nueva Etapa

En la Figura32 sección 1 se muestra el formulario para la creación de una etapa en el cual se deben ingresar los siguientes datos:

**Nombre:** Nombre de la etapa (*Nombre único en cada Proyecto*)

**Descripción:** Descripción de la etapa

En la sección 2 se mostrará una lista de los rangos de fecha disponibles para la creación de la etapa. Estos rangos permiten ser modificados hasta lapsos de un solo día.

En la sección 3 se puede añadir los usuarios asociados en la etapa. La prioridad depende del orden en el cual se encuentra el listado.

Figura 32 Formulario Creación Etapa



Los Jefes de Departamento solo pueden añadir usuarios de su Departamento

Luego de ingresar los datos se procede a presionar el botón **Crear** y la etapa será creada en el proyecto.

## Editar Etapa



Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el botón **Editar** en las acciones del listado de etapas que se muestra en la Figura 33.

Nombre	Fecha Inicio	Fecha Fin	Acciones
Etapa Final	02-03-2015	31-03-2015	Detalle Editar Ver
Etapa Adicional	01-04-2015	30-04-2015	Detalle Editar Ver
Etapa Única	05-12-2014	01-03-2015	Detalle Editar Ver

Figura 33 Acción Editar Etapa

En la Figura 34 sección 1 se muestra el formulario para la edición de una etapa, en el cual se pueden modificar los siguientes datos:

**Nombre:** Nombre de la etapa (*Nombre único en cada Proyecto*)

**Descripción:** Descripción de la etapa

En la sección 2 se mostrará una lista de los rangos de fecha disponibles para la modificación de la etapa. Estos rangos permiten ser modificados hasta lapsos de un solo día.

En la sección 3 se puede añadir o quitar los usuarios asociados en la etapa. La prioridad depende del orden en el cual se encuentra el listado.

Luego de realizar las modificaciones requeridas se procede a presionar el botón *Actualizar* y la etapa será actualizada.

Figura 34 Formulario Editar Etapa

## Ver Etapas



Esta sección es restringida para usuarios con rol Administrador, Jefe de Departamento o Usuario

Para acceder a esta sección se lo realiza mediante el botón *Ver* en las acciones del listado de etapas de cada proyecto como se muestra en la Figura 35.

Proyecto EPN Tesis			
Etapas			
Mostrar	10	registros	Buscar: <input type="text"/>
Nombre	Fecha Inicio	Fecha Fin	Acciones
Etapa Final	02-03-2015	31-03-2015	Detalle Editar Ver
Etapa Adicional	01-04-2015	30-04-2015	Detalle Editar Ver
Etapa Única	05-12-2014	01-03-2015	Detalle Editar Ver

Mostrando registros del 1 al 3 de un total de 3 registros

Anterior 1 Siguiente

Nueva Etapa Listado Proyectos

Figura 35 Acción Ver Etapa

En la Figura 36 se muestra la información de la Etapa, tiene controles similares a *Editar Etapa* pero se encuentran desactivados.

The screenshot shows the 'Editar Etapa' form with the following elements:

- Nombre:** Input field containing 'Hola'.
- Descripción:** Input field.
- Rango Actual:** Range selector with dates '01-2-2015' and '04-2-2015'.
- Listado de Usuarios:** A table with columns 'Nombre', 'Apellido', 'Extensión', and 'Departamento'. It contains the message 'Ningún dato disponible en esta tabla' and 'Mostrando registros del 0 al 0 de un total de 0 registros'. Navigation buttons 'Anterior' and 'Siguiete' are present.
- Buttons:** 'Regresar a Lista' and 'Eliminar' at the bottom.

Figura 36 Formulario Ver Etapa

## Eliminar Etapa

**i** Esta sección es restringida para usuarios con rol Administrador

Para eliminar una empresa se lo realiza mediante el botón *Eliminar* que se muestra en el formulario *Editar Etapa* de cada etapacom como indica la Figura 37.

The screenshot shows the 'Editar Etapa' form with the following elements:

- Nombre:** Input field containing 'Hola'.
- Descripción:** Input field.
- Rango Actual:** Range selector with dates '01-2-2015' and '04-2-2015'.
- Opción 1:** Range selector with dates '05-2-2015' and '01-3-2015'.
- Listado de Usuarios:** A table with columns 'Nombre', 'Apellido', 'Extensión', and 'Departamento'. It contains the message 'Ningún dato disponible en esta tabla' and 'Mostrando registros del 0 al 0 de un total de 0 registros'. A note below says 'Para cambiar la prioridad de los usuarios arrastrar y soltar las filas del listado'. Buttons 'Añadir Usuarios', 'Quitar Seleccionados', 'Actualizar', 'Regresar a Lista', and 'Eliminar' are present.

Figura 37 Eliminar Etapa

## 7 Usuarios

Este módulo permite gestionar los usuarios y los roles asignados a cada uno.

### Listado de Usuarios

**i** Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el menú *Sistema* submenú *Listado de Usuarios* como se muestra en la Figura 38.

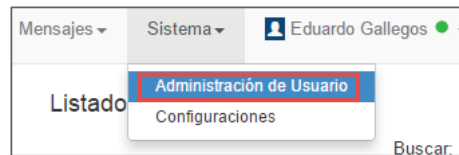


Figura 38 Enlace Administración de Usuario

En la Figura39 se muestra el listado de los usuarios creados. Este listado permite realizar una búsqueda por su nombre o apellido. Los botones de acciones dependen de las restricciones del rol del usuario y se revisaran en cada acción.

Listado Usuarios				
Mostrar	10	registros	Buscar:	<input type="text"/>
Nombre	Apellido	Estado	Acciones	
Alan	Endara	Activo	Ver	Editar Perfil Resetelear Contraseña
Andrés	Valdez	Activo	Ver	Editar Perfil Resetelear Contraseña
Carlos	Vaca	Activo	Ver	Editar Perfil Resetelear Contraseña
Damian	Danez	Activo	Ver	Editar Perfil Resetelear Contraseña
Eduardo	Gallegos	Activo	Ver	Editar Perfil Resetelear Contraseña
Hector	Lavoe	Activo	Ver	Editar Perfil Resetelear Contraseña
Martha	Valdez	Activo	Ver	Editar Perfil Resetelear Contraseña
Oscar	Cadena	Activo	Ver	Editar Perfil Resetelear Contraseña

Mostrando registros del 1 al 8 de un total de 8 registros

Anterior 1 Siguiente

Crear Nuevo

Figura 39 Listado de Usuarios

## Creación de Usuarios



Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el botón *Crear Nuevo* que se muestra en el listado de usuarios en la Figura 40.

Listado Usuarios				
Mostrar	10	registros	Buscar:	<input type="text"/>
Nombre	Apellido	Estado	Acciones	
Alan	Endara	Activo	Ver	Editar Perfil Reseteo Contraseña
Andrés	Valdez	Activo	Ver	Editar Perfil Reseteo Contraseña
Carlos	Vaca	Activo	Ver	Editar Perfil Reseteo Contraseña
Damian	Danez	Activo	Ver	Editar Perfil Reseteo Contraseña
Eduardo	Gallegos	Activo	Ver	Editar Perfil Reseteo Contraseña
Hector	Lavoe	Activo	Ver	Editar Perfil Reseteo Contraseña
Martha	Valdez	Activo	Ver	Editar Perfil Reseteo Contraseña
Oscar	Cadena	Activo	Ver	Editar Perfil Reseteo Contraseña

Mostrando registros del 1 al 8 de un total de 8 registros

Anterior 1 Siguiete

[Crear Nuevo](#)

Figura 40 Acción Crear Nuevo Usuario

En la Figura41 se muestra el formulario para la creación de una empresa en el cual se deben ingresar los siguientes datos:

**Nombre:** Nombre del usuario

**Apellido:** Apellido del usuario

**Nombre de Usuario:** Nombre de Usuario para acceso *(Debe ser único)*

**Contraseña:** Contraseña para acceso

**Repita Contraseña:** Verificación de Contraseña

**Teléfonos:** Teléfonos de Oficina, Móvil. Casa

**Email:** Email de contacto del usuario *(Debe ser único)*

**Activo:** Indica si el usuario está activo o no

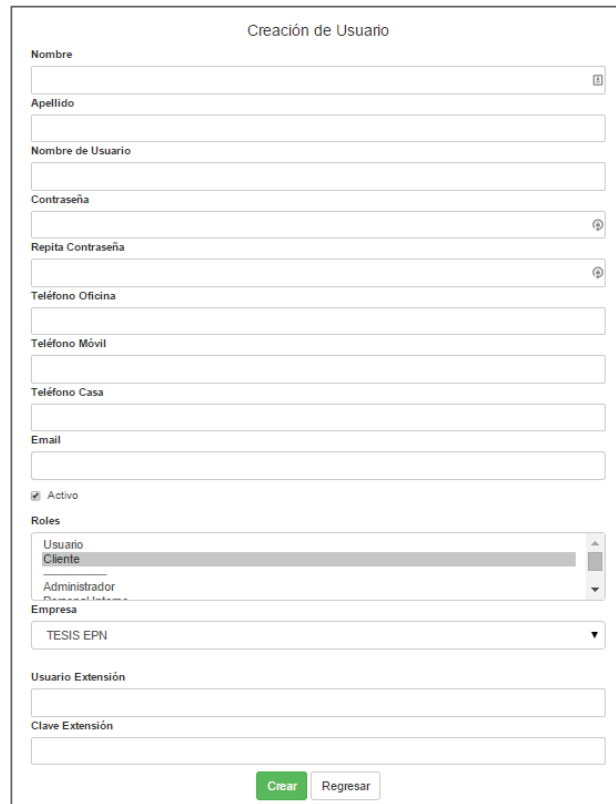
**Roles:** Se puede escoger uno o varios roles que serán asignados al usuario *(Si tiene el rol cliente escogido solo permitirá ese rol)*

**Empresa:** Empresa a la cual pertenece el usuario

**Usuario Extensión:** Extensión SIP

**Clave Extensión:** Contraseña SIP





Creación de Usuario

Nombre

Apellido

Nombre de Usuario

Contraseña

Repita Contraseña

Teléfono Oficina

Teléfono Móvil

Teléfono Casa

Email

Activo

Roles

- Usuario
- Cliente
- Administrador

Empresa

TESIS EPN

Usuario Extensión

Clave Extensión

Figura 41 Formulario Creación Usuario

Luego de ingresar los datos se procede a presionar el botón Crear y el usuario será creado.

## Editar Usuario



Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el botón *Editar* en las acciones del listado de usuarios que se muestra en la Figura 42.

Listado Usuarios			
Mostrar <input type="text" value="10"/> registros		Buscar: <input type="text"/>	
Nombre	Apellido	Estado	Acciones
Alan	Endara	Activo	Ver <span>Editar</span> <span>Perfil</span> <span>Resetear Contraseña</span>
Andrés	Valdez	Activo	Ver <span>Editar</span> <span>Perfil</span> <span>Resetear Contraseña</span>
Carlos	Vaca	Activo	Ver <span>Editar</span> <span>Perfil</span> <span>Resetear Contraseña</span>
Damian	Danez	Activo	Ver <span>Editar</span> <span>Perfil</span> <span>Resetear Contraseña</span>
Eduardo	Gallegos	Activo	Ver <span>Editar</span> <span>Perfil</span> <span>Resetear Contraseña</span>
Hector	Lavoe	Activo	Ver <span>Editar</span> <span>Perfil</span> <span>Resetear Contraseña</span>
Martha	Valdez	Activo	Ver <span>Editar</span> <span>Perfil</span> <span>Resetear Contraseña</span>
Oscar	Cadena	Activo	Ver <span>Editar</span> <span>Perfil</span> <span>Resetear Contraseña</span>

Mostrando registros del 1 al 8 de un total de 8 registros

Anterior **1** Siguiente

[Crear Nuevo](#)

Figura 42 Acción Editar Usuario

En la Figura43 se muestra el formulario para la edición de un usuario en el cual se pueden modificar los mismos datos del formulario de creación a excepción de la contraseña.

Editar Usuario	
<b>Nombre</b>	<input type="text" value="Eduardo"/>
<b>Apellido</b>	<input type="text" value="Gallegos"/>
<b>Nombre de Usuario</b>	<input type="text" value="admin"/>
<b>Teléfono Oficina</b>	<input type="text" value="565956"/>
<b>Teléfono Móvil</b>	<input type="text" value="28232856"/>
<b>Teléfono Casa</b>	<input type="text" value="32323516"/>
<b>Email</b>	<input type="text" value="edugallegos@gmail.com"/>
<input checked="" type="checkbox"/> Activo	
<b>Roles</b>	<input type="text" value="Usuario"/> <input type="text" value="Cliente"/> <input type="text" value="Administrador"/>
<b>Empresa</b>	<input type="text" value="TESIS EPN"/>
<b>Usuario Extensión</b>	<input type="text" value="1003"/>
<b>Clave Extensión</b>	<input type="text" value="1003"/>
<input type="button" value="Editar"/> <input type="button" value="Regresar"/>	
<input type="button" value="Eliminar"/>	

Figura 43 Formulario Editar Usuario

Luego de realizar las modificaciones requeridas se procede a presionar el botón *Editar* y el usuario será actualizado.

## Ver Usuario

 Esta sección es restringida para usuarios con rol Administrador

Para acceder a esta sección se lo realiza mediante el botón *Ver* en las acciones del listado de usuarios que se muestra en la Figura 44.

Listado Usuarios			
Mostrar	10	registros	Buscar: <input type="text"/>
Nombre	Apellido	Estado	Acciones
Alan	Endara	Activo	Ver Editar Perfil Reseteo Contraseña
Andrés	Valdez	Activo	Ver Editar Perfil Reseteo Contraseña
Carlos	Vaca	Activo	Ver Editar Perfil Reseteo Contraseña
Damian	Danez	Activo	Ver Editar Perfil Reseteo Contraseña
Eduardo	Gallegos	Activo	Ver Editar Perfil Reseteo Contraseña
Hector	Lavoe	Activo	Ver Editar Perfil Reseteo Contraseña
Martha	Valdez	Activo	Ver Editar Perfil Reseteo Contraseña
Oscar	Cadena	Activo	Ver Editar Perfil Reseteo Contraseña

Mostrando registros del 1 al 8 de un total de 8 registros

Anterior 1 Siguiente

Crear Nuevo

Figura 44 Acción Ver Departamento

En la Figura 45 se muestra la información de un usuario, excepto la contraseña y la clave de la extensión.

Información de Usuario	
Nombre	Alan
Apellido	Endara
Nombre de Usuario	aendara
Teléfono Oficina	
Teléfono Móvil	
Teléfono Casa	
Email	aendara@gmail.com
Estado	1
Extensión	1005

Regresar Editar Eliminar

Figura 45 Formulario Ver Usuario

## Eliminar Departamento

 Esta sección es restringida para usuarios con rol Administrador

Para eliminar un departamento se lo realiza mediante el botón *Eliminar* que se muestra en el formulario *Editar Usuario* de cada usuario como indica la Figura 46.

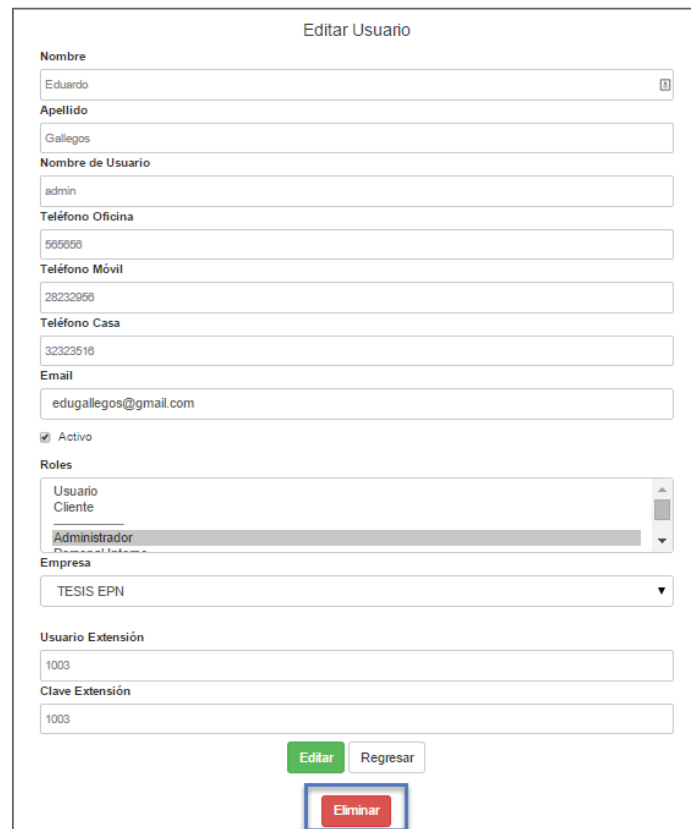


Figura 46 Eliminar Departamento

## 8 Llamadas

Este módulo permite realizar llamadas y ver los registros de las mismas.

### Marcación por Teclado



Esta sección es restringida para usuarios con rol Administrador, Jefe Departamento o Usuario

Para acceder a esta sección se lo realiza mediante el menú *Llamadas* submenú *Marcación por Teclado* como se muestra en la Figura 47.

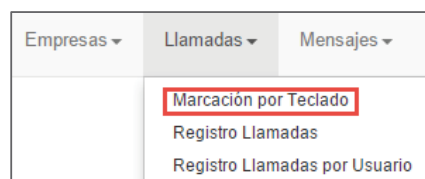


Figura 47 Enlace Marcación por Teclado

En la Figura48 sección 1 se muestra la pantalla de marcación por teclado previo a generar una llamada. Tiene una interfaz similar a un teléfono que permite realizar llamadas y/o contestarlas.

Mediante esta interfaz el usuario puede realizar llamadas a cualquier número, limitado según el plan de marcado que la extensión asociada al usuario tenga asignado en la central telefónica.

Para marcar el número el usuario puede hacerlo mediante el teclado en pantalla o con el teclado numérico físico. Para iniciar la llamada el usuario debe presionar el botón *Llamar*.

En la sección 2 se muestra el ingreso de una llamada que muestra los botones *Colgar* y *Contestar* que se los puede utilizar para cancelar la llamada o tomarla respectivamente.

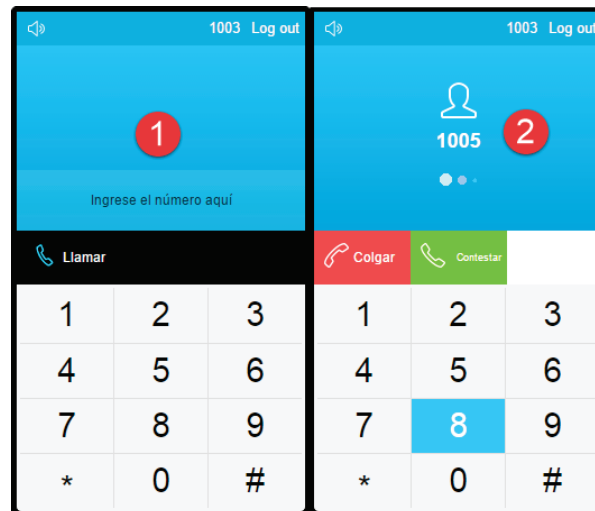


Figura 48 Marcación por Teclado

## Registro de Llamadas



Esta sección es restringida para usuarios con rol Administrador, Jefe Departamento o Usuario

Para acceder a esta sección se lo realiza mediante el menú *Llamadas* submenú *Registro Llamadas* como se muestra en la Figura 49.

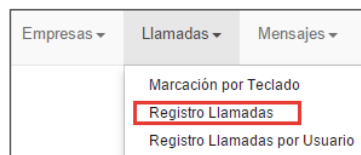


Figura 49 Enlace Registro de Llamadas

En la Figura50 se muestra el registro de llamadas realizadas y recibidas del usuario actual.

**Usuario: Eduardo Gallegos**  
Llamadas Realizadas

Mostrar 10 registros Buscar:

Usuario Origen	Usuario Destino	Duración	Fecha	Proyecto
Eduardo Gallegos	Carlos Vaca	00:00:02	04/01/2015 02:38:16	
Eduardo Gallegos	Alan Endara	00:00:02	04/01/2015 04:11:23	

Mostrando registros del 1 al 2 de un total de 2 registros Anterior 1 Siguiente

---

Llamadas Recibidas

Mostrar 10 registros Buscar:

Usuario Origen	Usuario Destino	Duración	Fecha	Proyecto
Carlos Vaca	Eduardo Gallegos	00:00:00	04/01/2015 01:42:44	
Carlos Vaca	Eduardo Gallegos	00:00:06	04/01/2015 01:43:39	
Carlos Vaca	Eduardo Gallegos	00:00:00	04/01/2015 01:44:40	

Mostrando registros del 1 al 3 de un total de 3 registros Anterior 1 Siguiente

Figura 50 Registro de Llamadas

## Registro de Llamadas por Usuario



Esta sección es restringida para usuarios con rol Administrador, Jefe Departamento

Para acceder a esta sección se lo realiza mediante el menú *Llamadas* submenú *Registro Llamadas por Usuario* como se muestra en la Figura 51.

Empresas	Llamadas	Mensajes
	Marcación por Teclado Registro Llamadas <b>Registro Llamadas por Usuario</b>	

Figura 51 Enlace Registro de Llamadas por Usuario

En la Figura 52 se muestra un listado de Usuarios cada uno con un botón *Ver Registro*, que permite ver los registros de llamada de cada uno.

**Registros de Llamadas por Usuario**

Mostrar 10 registros Buscar:

Nombre	Apellido	Departamento	Acciones
Alan	Endara	Desarrollo	Ver Registro
Andrés	Valdez		Ver Registro
Carlos	Vaca	Ventas	Ver Registro
Damian	Danez		Ver Registro
Eduardo	Gallegos	Desarrollo	Ver Registro
Hector	Lavoe		Ver Registro
Martha	Valdez		Ver Registro
Oscar	Cadena		Ver Registro

Mostrando registros del 1 al 8 de un total de 8 registros Anterior 1 Siguiente

Figura 52 Listado de Usuarios

En la Figura 53 se muestra el registro de llamadas realizadas y recibidas del usuario escogido previamente.

**Usuario: Eduardo Gallegos**

Llamadas Realizadas

Mostrar: 10 registros      Buscar:

Usuario Origen	Usuario Destino	Duración	Fecha	Proyecto
Eduardo Gallegos	Carlos Vaca	00:00:02	04/01/2015 02:38:16	
Eduardo Gallegos	Alan Endara	00:00:02	04/01/2015 04:11:23	

Mostrando registros del 1 al 2 de un total de 2 registros      Anterior 1 Siguiente

---

Llamadas Recibidas

Mostrar: 10 registros      Buscar:

Usuario Origen	Usuario Destino	Duración	Fecha	Proyecto
Carlos Vaca	Eduardo Gallegos	00:00:00	04/01/2015 01:42:44	
Carlos Vaca	Eduardo Gallegos	00:00:06	04/01/2015 01:43:39	
Carlos Vaca	Eduardo Gallegos	00:00:00	04/01/2015 01:44:40	

Mostrando registros del 1 al 3 de un total de 3 registros      Anterior 1 Siguiente

Figura 53 Registro de Llamadas por Usuario

## 9 Cliente

### Acceso

Para el acceso a la aplicación se debe dirigir mediante un navegador a la siguiente url:

*http://[dirección\_del\_servidor]/Support/web/login*

En esta url se mostrará el recuadro de la Figura 54 que nos permite ingresar el usuario y contraseña asignados previamente por el Administrador del sistema.

Figura 54 - Acceso a la Plataforma



La casilla **Recordar** permite a los usuarios que su información de acceso quede almacenada por 1 hora aunque la ventana sea cerrada. De esta manera no tendrá que ingresar sus datos de autenticación cada vez que ingrese a la aplicación.

## Perfil de Cliente

Luego de que el cliente acceda a la plataforma se mostrará una pantalla con el perfil del cliente como se muestra en la Figura 55 que cuenta las siguientes secciones:

6. Menú
7. Información del Usuario
8. Información de la Empresa
9. Información de Proyectos
10. Inicio de Chat

Soporte Andrés Valdez **1**

**Perfil Actual**

Usuario: avaldez

Nombre: Andrés

Apellido: **2**

Tif. Principal

Tif. Secundario

Tif. Auxiliar

Email: avaldez@gmail.com

**Proyectos Asociados**

Mostrar 10 registros

Buscar:

Proyecto	Descripción	Fecha Inicio	Fecha	Soporte
EPN Tesis	Proyecto para graduación	2014-12-05	<b>4</b> 14-30	Llamar Por Departamento

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior **1** Siguiente

**Datos de la Empresa**

Nombre: **3** AN

RUC: 1717130164

Email Institucional: admin@gypsan.com

Tif. Institucional: 56591423

Página Web: www.gypsan.com

Dirección Matriz

Ayuda en línea **5** EN LINEA CONTACTE A UN OPERADOR

Figura 55 Perfil del Cliente

### Menú

El menú permite acceder al perfil del usuario y salir de la plataforma

### Información del Usuario

En esta sección se muestra toda la información del usuario actual, entre estos datos se encuentra el nombre, apellido, extensión, departamento, teléfonos y email.

### Información de la Empresa

En esta sección se muestra la información de la Empresa a la cual pertenece el usuario.

### Información de Proyectos



En esta sección se muestran los proyectos que tiene el usuario.

### Inicio de Chat

Esta sección permite acceder al chat de soporte.

### Ingreso Chat Cliente

Para acceder al chat debemos presionar el botón que aparece en la pantalla de perfil que se muestra en la Figura 56.



Figura 56 Botón inicio de chat

Al presionar el botón mencionado anteriormente se mostrará la ventana de chat indica en la Figura 57 solicitando la información del departamento con el cual se desea comunicar. El cuadro nombre tiene el nombre del cliente que ha iniciado el chat. Este no se puede modificar. Al escoger el departamento se debe dar clic en el botón *Iniciar Chat* para que nos muestre la ventana de chat.

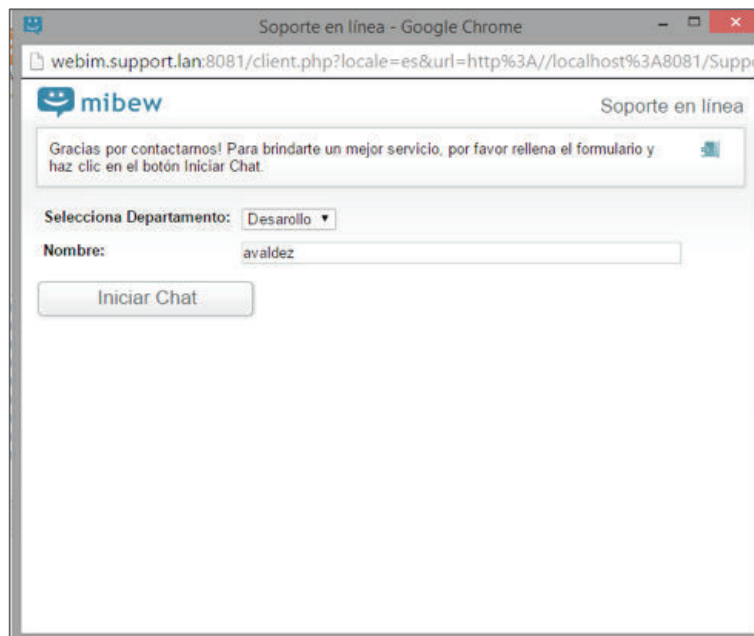
A screenshot of a web browser window titled 'Soporte en línea - Google Chrome'. The address bar shows 'webim.support.lan:8081/client.php?locale=es&url=http%3A//localhost%3A8081/Supp'. The page header features the 'mibew' logo and the text 'Soporte en línea'. A message box says: 'Gracias por contactarnos! Para brindarte un mejor servicio, por favor rellena el formulario y haz clic en el botón Iniciar Chat.' Below this is a form with a dropdown menu labeled 'Selecciona Departamento:' with 'Desarollo' selected. A text input field labeled 'Nombre:' contains the text 'avaldez'. At the bottom of the form is a button labeled 'Iniciar Chat'.

Figura 57 Escoger Departamento

En la ventana de chat que se muestra en la Figura 58 se indica la sección 1 correspondiente al cuadro de los mensajes y la sección 2 donde se pueden escribir los nuevos mensajes.

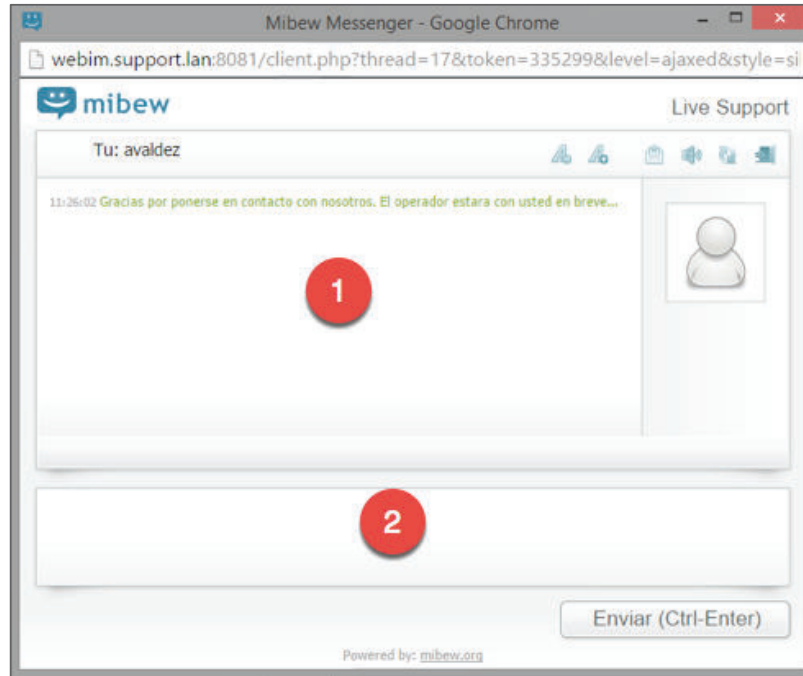


Figura 58 Ventana de Chat

## Llamadas a Usuarios

El cliente puede llamar a los usuarios que se encuentran asignados a un proyecto escogido, para esto en el perfil del usuario damos clic sobre el botón *Llamas* del proyecto que se muestra en la Figura 59.

Proyectos Asociados				
Proyecto	Descripción	Fecha Inicio	Fecha Fin	Soporte
EPN Tesis	Proyecto para graduación	2014-12-05	2015-04-30	Llamar Por Departamento

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

Figura 59 Botón Llamar

Al presionar el botón mencionado anteriormente se mostrará un listado con los usuarios asignados al proyecto escogido, cada uno con su respectivo botón llamar que permite realizar una llamada con cualquiera de ellos como se muestra en la Figura 60. El orden en el que se muestran corresponde es según la prioridad de asignación en la etapa actual. El botón *Colgar* permite interrumpir una llamada.

Soporte Proyecto EPN Tesis			
ETAPA			
Nombre: Etapa Adicional			
Descripción: Etapa para completar la implementación			
Nombre	Apellido	Departamento	Acciones
Alan	Endara	Desarrollo	Llamar Colgar
Carlos	Vaca	Ventas	Llamar Colgar

Regresar

Figura 60 Listado de Usuarios Asignados

## Llamadas a Departamentos

El cliente puede llamar a los departamentos asignados a un proyecto escogido, para esto en el perfil del usuario damos clic sobre el botón *Por Departamento* del proyecto que se muestra en Figura 61.

Proyectos Asociados				
Mostrar 10 registros		Buscar:		
Proyecto	Descripción	Fecha Inicio	Fecha Fin	Soporte
EPN Tesis	Proyecto para graduación	2014-12-05	2015-04-30	Llamar <b>Por Departamento</b>

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

Figura 61 Botón Por Departamento

Al presionar el botón mencionado anteriormente se mostrará un listado con los departamentos asignados al proyecto escogido, cada uno con su respectivo botón llamar que permite realizar una llamada con cualquiera de ellos como se muestra en la Figura 62. El botón *Colgar* permite interrumpir una llamada.

Soporte Proyecto EPN Tesis			
ETAPA			
Nombre: Etapa Adicional			
Descripción: Etapa para completar la implementación			
Nombre	Apellido	Departamento	Acciones
Alan	Endara	Desarrollo	Llamar Colgar
Carlos	Vaca	Ventas	Llamar Colgar

Regresar

Figura 62 Listado de Departamentos

## 10 Aplicación Móvil

---

### Acceso

Para el acceso a la aplicación se debe ejecutar la aplicación en el iPhone como muestra la Figura 63.

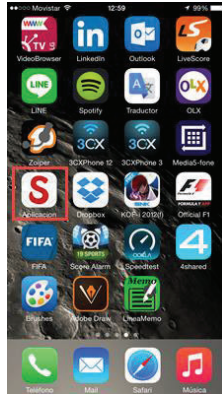


Figura 63 – Icono Aplicación

Al ingresar a la aplicación nos mostrará una pantalla de acceso para ingresar el nombre de usuario y la contraseña similar a la Figura 64.



Figura 64 - Acceso Aplicación



En caso de que los datos son erróneos mostrará un mensaje de datos incorrectos

### Conversaciones

Luego de acceder los datos correctamente se mostrará un listado de las conversaciones nuevas y abiertas del usuario registrado como indica la Figura 65.



Figura 65 Listado Conversaciones

En cada una se mostrará la hora a la cual fue realizado. Cuando una nueva conversación sea iniciada o un cliente envíe un mensaje al usuario le llegará una notificación.

## Mensajes

Al presionar sobre alguna conversación del listado se procederá a mostrar el historial de mensajes intercambiados entre el cliente y el usuario.

En caso de que sea una nueva conversación la misma será asociada directamente al usuario como operador de esa conversación.

En la Figura66 se muestra en la sección 1 los mensajes intercambiados y en la sección 2 el teclado con la caja de texto correspondiente para agregar nuevos mensajes.



Figura 66 Mensajes

El botón Send se activará cuando se encuentre escrito por lo menos un carácter en la caja de texto y nos permitirá enviar los mensajes.