

Universidad de las Ciencias Informáticas

Facultad 2



Sistema de Operadoras en Línea para Asterisk

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

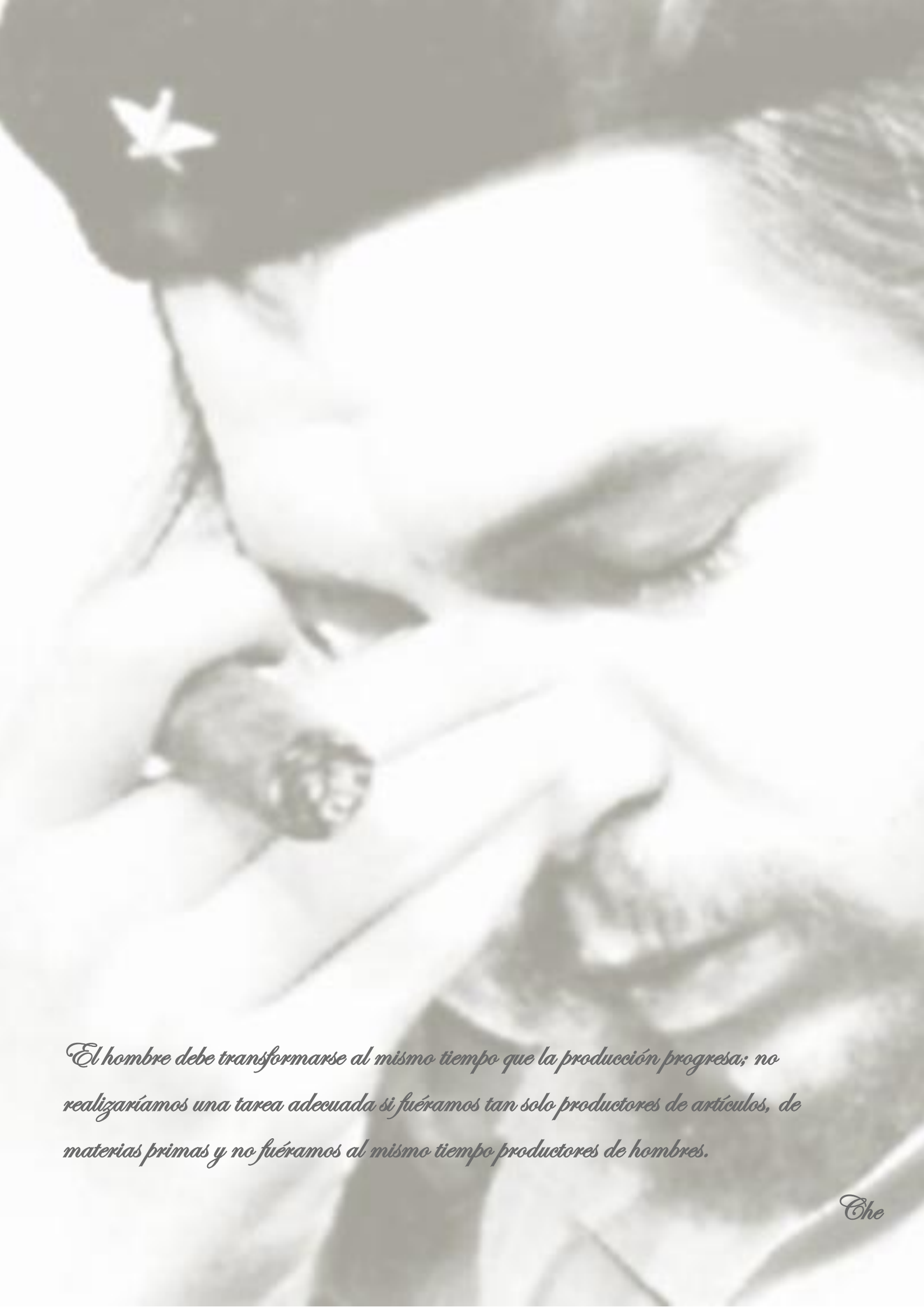
Autores: Evis Domínguez Pérez.

Yasiel Cabrera Iglesias.

Tutores: Ing. Félix Maikel García.

Co-Tutores: Ing. Rainer Segura Peña.

Ing. Danae Pérez Áreas.



El hombre debe transformarse al mismo tiempo que la producción progresa; no realizaríamos una tarea adecuada si fuéramos tan solo productores de artículos, de materias primas y no fuéramos al mismo tiempo productores de hombres.

Declaración de Autoría

Declaramos ser los autores de la presente tesis, reconociendo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de manera exclusiva.

Para que así conste, firmamos la presente a los ____ días del mes de _____ de _____

Firma Autor

Yasiel Cabrera Iglesias

Firma Autor

Evis Domínguez Pérez

Firma del Tutor

Ing. Félix Maikel García Pérez

Dedicatoria

Yasiel

Dedico esta tesis a mi mamá y a mi papá que han sido un apoyo constante durante toda mi vida, a mi esposa Anaymi, a mi hijo por darme fuerzas con su presencia para enfrentarme a todos los retos de la carrera, a mis suegros por confiar en mí y apoyarme siempre, a mi cuñado, mi hermano el Tuti que siempre estuvo a mi lado aconsejándome, a mi primo Willy que fue mi inspiración. También a mi abuelita, a mi tía y a mi hermano que aunque no están presentes físicamente son personas que contribuyeron en mi formación.

Evis

Dedico esta tesis primeramente a mi mamá y a mi papá que han sido siempre una inagotable fuente de apoyo en cada etapa de mi vida, a mi novia Ilismay que me ha visto tropezar y levantarme en gran parte del transcurso de mi carrera guiándome siempre por el buen camino, a mis abuelos, a mis tíos y tías, a mis primos y primas, a mi hermano Yoennis, a todas mis grandes amistades y a todos aquellos que de una forma u otra han influido y ayudado en mi trayectoria estudiantil.

Agradecimientos

Yasiel

A mi mamá y a mi papá por siempre estar apoyándome en los buenos y malos momentos y nunca dejarme flaquear ante los obstáculos que se han presentado en mi camino.

A mi esposa por siempre estar a mi lado y esperarme durante estos cinco años en los cuales siempre supo ser paciente y confiar en mí.

A mi hijo por ser mi inspiración y mi mayor tesoro.

Al profesor Félix Maikel García y Carlos Molina por haber encontrado en ellos un amigo y guía durante mi carrera.

A mi compañero de tesis Evis Domínguez por ser más que un compañero mi amigo.

A mis compañeros de aula y en especial de apartamento Miguel, Jose, Pedro, Evis, Rodolfo por todo el apoyo brindado en todo este tiempo.

A mis co-tutores Rainer y Danae por todo el apoyo brindado.

A mis compañeros y profesores de proyecto Yaneris, Arianna, Yordan, Yadier, a mi tocallo Yasiel, a Yury La estaca, Norbel.

Al profesor Erick por darnos las mieles.

A mis compañeros de la secundaria Yandy, Acosta, Arturo, Marlon y Leonel.

Evis

Un agradecimiento sumamente especial a mi mamá y a mi papá que siempre han sabido estar ahí cuando más los he necesitado apoyándome y sobre todo siempre confiando en mí.

A mi novia Ilismay por estar a mi lado brindándome su incondicional apoyo y haciendo que lo malo parezca bueno y que lo imposible parezca más que posible alcanzable.

A toda mi familia en general que también siempre ha sabido apoyarme y aconsejarme en los malos momentos.

A mi compañero de tesis Yasiel Cabrera por haber sido un gran amigo en estos cinco años de andanzas y batallas.

A mi Tutor y amigo Félix Maiquel García (Felipao) por haber sido para mí un guía en mi desempeño y también quisiera agradecer al profesor Carlos Molina.

A mis compañeros de aula y en especial de apartamento Miguel, Yasiel, Pedro, Jose y Rodolfo por todo el apoyo brindado en esta gran jornada.

A mis co-tutores Rainer y Danae por todo el apoyo brindado y todo el tiempo dedicado.

A mis compañeros y profesores de proyecto Yordan, Yadier, Yaneris, Arianna, a Erik por darnos las mieles y la pulpa, a Yury ("La estaca"), Reinol.

A todos los que en general me ayudaron y apoyaron en esta ardua tarea.

Resumen

La comunicación entre usuarios a través de operadoras es un servicio básico que brindan hoy en día todas las plantas telefónicas. La Empresa de Telecomunicaciones de Cuba (ETECSA) cuenta con un sistema autónomo de operadoras (SYSOPE) para brindar este servicio a la población. Debido a que este sistema no posee soporte alguno, presenta una interfaz poco amigable, el personal que atiende este servicio no puede ser reutilizado en otras tareas, tiene dificultades en su sistema de reportes además de ser totalmente dependiente de su hardware el cual ya no es producido por ningún fabricante de equipos de telecomunicaciones lo que no permite la integración de nuevos servicios o sistemas a él y viceversa, se hace necesario contar con un sistema que erradique los problemas expuestos anteriormente. Con el presente trabajo se pretende desarrollar un Sistema de Operadoras en Línea para Asterisk que permita la reutilización del personal para la prestación de otros servicios, que cuente con una interfaz amigable que facilite el trabajo de las ejecutivas y que cuente con un sistema de reportes además de ser totalmente independiente de su hardware y permita brindarle soporte.

Palabras Claves:

Llamadas Telefónicas, Ejecutiva Telefónica, Planta Telefónica.

ÍNDICE

Dedicatoria.....	4
Agradecimientos	5
Introducción.....	1
Capítulo 1. Fundamentación Teórica.....	5
1 Introducción.....	5
1.2 Conceptos a tratar.....	5
1.2.1 Llamada.....	5
1.2.2 Llamadas Telefónicas	5
1.2.3 Ejecutiva Telefónica	5
1.2.4 Planta Telefónica	5
1.2.5 Asterisk.....	6
1.3 Sistemas de Operadoras en línea en el mundo.....	6
1.3.1 EOLA (Puesto Avanzado de Operadora Telefónica)	6
1.3.2 SmartIVR.....	7
1.4 Sistemas de Operadoras en línea en Cuba.....	7
1.4.1 Sistema de Operadoras en línea (SYSOPE).....	7
1.5 Metodologías de Desarrollo:.....	8
1.5.1 Metodología de software seleccionada	9
1.6 Sistema Gestor de Base de Datos	10
1.7 Herramientas de Desarrollo.....	11
1.7.1 Lenguaje programación del lado del servidor	11
1.7.2 Lenguaje de programación del lado del cliente.....	11
1.7.3 Symfony 1.4.....	12
1.7.4 Visual Paradigm 8.0	13
1.7.6 Quanta Plus 3.5	13
1.7.7 Servidor web Apache 2.2.....	14
1.7.9 BPM	14
1.8 Conclusiones	15
Capítulo 2. Exploración y Propuesta del Sistema	16
2 Introducción.....	16

2.1 Modelado del Negocio.....	16
2.2 Propuesta del Sistema	18
2.3 Lista de reservas del sistema de operadoras en línea.....	21
2.3.1 Usabilidad	21
2.3.2 Disponibilidad	21
2.3.3 Eficiencia.....	21
2.3.4 Hardware.....	22
2.3.5 Software	22
2.3.6 Interfaz de usuario	22
2.4 Personas relacionadas con el sistema	22
2.5 Fase de Exploración	22
2.5.1 Historias de Usuario	22
2.6 Planificación.....	33
2.6.1 Estimación de esfuerzo por Historias de Usuario.....	33
2.6.2 Plan de Iteraciones.....	35
2.6.3 Iteración 1	35
2.6.4 Iteración 2	35
2.6.5 Plan de duración de las iteraciones	35
2.7 Conclusiones	37
Capítulo 3. Diseño del Sistema	38
3 Introducción.....	38
3.1 Arquitectura.....	38
3.2 Patrones Arquitectónicos	39
3.2.1 Patrón Arquitectónico Modelo Vista Controlador.....	39
3.2.2 Patrón Arquitectónico N-Capas.....	40
3.3 Diagrama de Clases	41
3.3.1 Diagrama de Clases del Servicio	41
3.3.2 Diagrama de Clases de la Aplicación Web.....	43
3.4 Patrones de Diseño	45
3.5 Tarjetas Clase – Responsabilidad – Colaborador	47

3.6 Modelo Físico de la Base de Datos.....	50
3.7 Conclusiones	52
Capítulo 4. Implementación y Prueba	52
4 Introducción.....	52
4.1 Tareas de la Ingeniería	52
4.2 Pruebas.....	57
4.3 Conclusiones	61
Conclusiones	62
Recomendaciones.....	63
Trabajos citados.....	64
Bibliografía.....	66
Glosario de Términos:.....	68

ÍNDICE DE TABLAS

Tabla 1 Características del equipo de desarrollo.....	8
Tabla 2 Características del cliente.....	9
Tabla 3 HU #1: Autenticar Usuario.....	24
Tabla 4 HU #2: Cargar configuración.....	25
Tabla 6 HU #4: Llamar usuario.....	26
Tabla 7 HU #5: Cancelar llamada.....	27
Tabla 8 HU #6: Comunicar usuarios.....	28
Tabla 12 HU #10: Enviar ticket a facturar.....	29
Tabla 13HU #11: Establecer cita.....	29
Tabla 14HU #12: Mostrar la cantidad de llamadas establecidas por cada ejecutiva.....	30
Tabla 16 HU #14: Mostrar la actividad del sistema por hora.....	31
Tabla 17 HU #15: Mostrar actividad de una ejecutiva específica.....	32
Tabla 18 Estimación de esfuerzo por HU.....	34
Tabla 19 Plan de duración de las iteraciones.....	35
Tabla 20 CRC: Clase Arranque.....	47
Tabla 21 CRC: Clase Configuración.....	47
Tabla 22 CRC: Clase Comunicador.....	48
Tabla 23 CRC: Clase ControladorEventos.class.....	48
Tabla 24 CRC: Clase FTP_Manager.....	48
Tabla 25 CRC: Clase control_Col_Agents.class.....	48
Tabla 26 CRC: Clase Manager.....	48
Tabla 27 CRC: Clase SeguridadActions.....	49
Tabla 28 CRC: Clase ReportesActions.....	49
Tabla 29 CRC: Clase EjecutivaActions.....	49
Tabla 30 Tarea #1 Autenticar Usuario.....	53
Tabla 31 Tarea #2 Cargar configuración.....	53
Tabla 33 Tarea #4 Llamar usuario.....	54
Tabla 34 Tarea #5 Cancelar llamada.....	54
Tabla 35 Tarea #6 Comunicar usuarios.....	54
Tabla 39 Tarea #10 Enviar ticket a facturar.....	55
Tabla 40 Tarea #11 Establecer cita.....	55
Tabla 41 Tarea #12 Mostrar la cantidad de llamadas establecidas por cada ejecutiva.....	56
Tabla 43 Tarea #14 Mostrar la actividad del sistema por hora.....	56
Tabla 44 Tarea #15 Mostrar actividad de una ejecutiva específica.....	57

ÍNDICE DE FIGURAS

Ilustración 1 Modelo del Negocio	17
Ilustración 2 Propuesta del Sistema	18
Ilustración 3 Servicio de Comunicación	20
Ilustración 4 Arquitectura cliente servidor	38
Ilustración 5 Modelo Vista Controlador	40
Ilustración 6 Symfony - Modelo Vista Controlador	40
Ilustración 7 Arquitectura N-Capas	41
Ilustración 8 Capa Negocio	42
Ilustración 9 Capa Acceso a Datos	43
Ilustración 10 Capa Vista.....	43
Ilustración 11 Controladora	44
Ilustración 12 Capa Modelo	44
Ilustración 13 Modelo físico Base Datos call_center.....	51

Introducción.

Desde principios de la humanidad se hizo indispensable la comunicación como necesidad para poder transmitir ideas, ya fuera a través de señas o pinturas, luego sería por medio del lenguaje hablado como cúspide de la comunicación humana a raíz del desarrollo alcanzado e impulsado por la propia necesidad.

Tras siglos de evolución poco a poco se fue perfeccionando la comunicación y comienzan a surgir los primeros equipos encaminados a este fin hasta llegar al dispositivo que hoy en día se conoce como teléfono, capaz de permitir una conversación no presencial entre personas desde cualquier parte del mundo sin que la distancia sea una limitante, a partir de este momento se da comienzo a una nueva Era de las Telecomunicaciones.

Actualmente existe un gran desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC), llegando a formar parte indispensable de las actividades cotidianas de las personas. Las crecientes necesidades de comunicación en la actual sociedad han traído aparejado un desarrollo acelerado de la telefonía y por ende la demanda de nuevos servicios telefónicos que cuentan con los mecanismos de conmutación de llamadas de larga o corta distancia, solo posible de realizar mediante el uso de plantas telefónicas. Las plantas telefónicas físicas son de gran importancia para empresas e instituciones, pero no permiten agregarles nuevos servicios para mejorar la atención al cliente. La solución a este problema es la utilización de plantas telefónicas basadas en software ya que permiten de manera fácil la integración de nuevos servicios y funciones.

“Asterisk figura entre uno de los sistemas de software libre más usados y reconocidos internacionalmente para realizar llamadas y brindar servicios dentro de la rama de la telefonía. Básicamente Asterisk es un software de central telefónica con capacidad para voz sobre IP que es distribuido bajo licencia libre. Esto ha hecho que muchas empresas consideren a Asterisk como una seria opción al momento de planificar su proyecto telefónico y por esta razón Asterisk ha tenido gran acogida a nivel mundial.” (1)

En Cuba se está llevando a cabo una revolución en la rama de las telecomunicaciones por lo que se ha volcado hacia la informatización de la sociedad en este campo, basándose en el uso de tecnologías de software libre.

La Empresa de Telecomunicaciones de Cuba S.A (ETECSA) actualmente ofrece el servicio de llamadas de cobro revertido apoyándose en el sistema de operadoras en línea conocido por sus siglas SYSOPE para realizar llamadas de largas distancias, tanto nacionales como internacionales, en caso de que se desee definir el teléfono al que se le recargará la factura de la llamada realizada.

El uso del sistema SYSOPE trae consigo demoras en la atención al cliente debido a que no proporciona una interfaz amigable para interactuar con el mismo, el intercambio de datos con el sistema se realiza a través de una consola y del uso de comandos que se introducen mediante un teclado diseñado solamente para este uso. En los reportes realizados por el sistema solo se tiene la información de las llamadas realizadas en un período de 15 minutos, como mínimo, entre un reporte y otro, la información de las llamadas anteriores a este tiempo definido se pierde, además los reportes no presentan un conjunto de criterios que permitan filtrar la información.

SYSOPE es un sistema totalmente dependiente de su hardware el cual ya no es producido por ningún fabricante de equipos de telecomunicaciones. Esto impide que otros sistemas o servicios puedan ser integrados a él o viceversa.

ETECSA en busca de solucionar el problema anterior e integrar todos sus servicios sobre una misma plataforma ha optado por utilizar Asterisk e integrarle al mismo un sistema de operadoras en líneas que mejore la atención al cliente y pueda ser adaptado a sus necesidades.

A partir de la situación problemática descrita, se deriva el siguiente **Problema a resolver**:

¿Cómo brindar el servicio de operadoras en líneas para Asterisk?

Por tanto el **Objeto de Estudio** será: Los sistemas informáticos para operadoras telefónicas.

Partiendo del problema antes planteado se tiene como **Objetivo General**: Desarrollar un sistema de operadoras en línea integrado a Asterisk que permita comunicar dos usuarios a través de una ejecutiva y generar un ticket de reporte para facturar dichas llamadas.

El **Campo de Acción** está comprendido por los sistemas de operadoras en línea integrado a Asterisk.

Objetivos Específicos:

- ❖ Identificar los procesos que intervienen en el servicio de comunicación de dos usuarios a través de una ejecutiva.

- ❖ Diseñar el sistema de operadoras en línea para Asterisk.
- ❖ Implementar el sistema de operadoras en línea para Asterisk.
- ❖ Realizar pruebas el sistema de operadoras en línea obtenido.

La **Idea a Defender** plantea que el desarrollo de un sistema de operadoras en línea integrado a Asterisk permitirá comunicar dos usuarios a través de una ejecutiva y generar un reporte para la facturación de dichas llamadas.

Para dar cumplimiento a los objetivos propuestos se definen las siguientes **Tareas de Investigación**:

- ❖ Realización de encuentros y entrevistas con el personal del centro de llamadas para conocer cómo se realizan los procesos relacionados con las llamadas a través de ejecutivas.
- ❖ Estudio de los sistemas existentes para una mejor comprensión del funcionamiento de los mismos.
- ❖ Estudio de las herramientas a utilizar para un correcto desempeño con las mismas.
- ❖ Estudio de Lenguaje e IDE a utilizar para desarrollar el sistema de forma correcta.
- ❖ Estudio de la planta telefónica Asterisk para comprender su funcionamiento interno.
- ❖ Estudio del Call Center del Elastix para poder integrar el sistema a Asterisk.

Como base de apoyo para el desarrollo de la investigación se hace uso de los siguientes **Métodos Científicos**:

➤ **Métodos Teóricos:**

Analítico – Sintético: Este método posibilita el análisis de las teorías y documentos vinculados al objetivo de la investigación, permitiendo resumir los elementos más importantes relacionados con el objeto de estudio permitiendo construir una línea de investigación.

Modelación: Este método permite la modelación a través de diagramas que constituyen una representación abreviada y estructural de lo que luego sería el producto real.

➤ **Métodos Empíricos:**

Entrevista: Método que se utiliza como herramienta para a través del intercambio con el cliente detectar las necesidades del mismo y lograr satisfacerlas. Para ello se realizaron una serie de encuentros con el personal de ETECSA vinculado al trabajo con el sistema SYSOPE y algunos especialistas en el tema para definir los problemas existentes a los que se les dará solución posteriormente.

Observación: Consiste en aplicar los sentidos a un objeto o a un fenómeno, para estudiarlo tal como se presenta en realidad. Este método se utiliza inicialmente en el diagnóstico del problema y luego en el diseño de la investigación. Con este método se logra reunir información visual sobre todo lo que ocurre respecto al problema que se intenta solucionar, en este caso cómo se comporta el SYSOPE y qué es lo que hace.

El presente trabajo de diploma consta de 4 capítulos donde se describe todo el proceso de investigación.

Capítulo 1. Fundamentación Teórica: Describe los principales conceptos a tratar: llamadas, llamadas telefónicas, ejecutiva y Asterisk como software que proporciona las funcionalidades de una planta telefónica. Define la metodología de desarrollo, herramientas y tecnologías a utilizar para la aplicación. También los procesos que definen los sistemas de operadoras en línea.

Capítulo 2. Exploración y Propuesta del Sistema: Caracteriza el sistema, se plantea a grandes rasgos las Historias de Usuarios (HU) que son de interés para la primera entrega del producto así como una propuesta del prototipo no funcional de la aplicación.

Capítulo 3. Diseño del Sistema: Define la arquitectura del sistema y con ello patrones arquitectónicos empleados. Se procederá a abordar las HU definidas y tareas de la ingeniería en cada iteración para luego hacer una revisión de las mismas.

Capítulo 4. Implementación y Prueba: Materializa la propuesta del sistema que da cumplimiento a los objetivos trazados con la investigación y la validación del mismo a través de las pruebas realizadas al sistema.

Capítulo 1. Fundamentación Teórica

1 Introducción

En este capítulo se profundizan los conceptos fundamentales para el desarrollo del Sistema de Operadoras en Línea para Asterisk, se realiza un estudio sobre los sistemas de operadoras utilizados por empresas líderes en el mundo y se exponen un conjunto de características de las tecnologías, herramientas y metodología utilizada.

1.2 Conceptos a tratar

1.2.1 Llamada

“Voz, sonido o gesto que sirven como señal para atraer la atención de una persona o animal”. (2)

1.2.2 Llamadas Telefónicas

“En términos de telefonía, una llamada telefónica es la acción de emitir una señal o aviso por medio de un teléfono”. (3)

1.2.3 Ejecutiva Telefónica

“Persona que trabaja en el servicio de aparatos telefónicos, especialmente la que se dedica a contestar llamadas en una centralita” (4).

1.2.4 Planta Telefónica

“Una planta telefónica es el sistema para la operación de llamadas telefónicas en el sentido de hacer conexiones y retransmisiones de información de voz, en forma analógica, digital o por IP.” (5)

Estas plantas también poseen un sin número de funcionalidades que hacen más fácil la prestación del servicio de telefonía entre las que se encuentran: permitir marcar el último número marcado, dejar mensajes, llamadas en espera, alarma, etc.

En el mundo existen muchos productores de plantas telefónicas entre ellos se tienen: ALCATEL, MITEL, PANASONIC, ERICSSON, INFINITY, TOSHIBA y LG.

1.2.5 Asterisk

“Asterisk es una aplicación para controlar y gestionar comunicaciones de cualquier tipo, ya sean analógicas, digitales o VoIP mediante todos los protocolos VoIP que implementa.

Es una aplicación OpenSource basada en licencia GPL y por lo tanto con las ventajas que ello representa, lo que lo hace libre para desarrollar sistemas de comunicaciones profesionales de gran calidad, seguridad y versatilidad.

Incluye funcionalidades encontradas en los sistemas de comunicación más recientes tales como correo de voz, colas de llamadas, conferencias, audio respuesta, música en espera y otras funcionalidades más avanzadas que permiten la interconexión con sistemas de telefonía externos a través de troncales análogas, digitales o las más avanzadas con interfaces para VoIP .

Esta poderosa combinación de funcionalidades permite construir aplicaciones tan complejas o avanzadas como se desee sin incurrir en altos costos y con más flexibilidad que en cualquier sistema de telefonía existente a la fecha.” (6)

1.3 Sistemas de Operadoras en línea en el mundo

1.3.1 EOLA (Puesto Avanzado de Operadora Telefónica)

“Sistema informático a través del cual las operadoras de la centralita pueden realizar funciones propias de una telefonista, sustituyendo la consola de operadora clásica por una avanzada aplicación informática que se ejecuta desde un ordenador.

EOLA es una solución dirigida en general a empresas que gestionan de forma corporativa un elevado volumen de tráfico telefónico. En el sector público está especialmente indicado para su instalación en Gabinetes Telegráficos Ministeriales.” (7)

Este sistema no permite su integración a Asterisk y además funciona con IVRs.

1.3.2 SmartIVR

“Es un sistema informático de generación y explotación de soluciones de respuesta vocal interactiva (IVR) destinadas a la automatización de procesos de atención de llamadas telefónicas. El sistema SmartIVR atiende con rapidez y eficiencia todas las llamadas entrantes con independencia del nivel de ocupación de los operadores las 24 horas, los 365 días del año.

Características del sistema:

- ❖ Incremento del número de llamadas telefónicas atendidas, ofreciendo un acceso inmediato.*
- ❖ Las llamadas atendidas pueden ser transferidas a las operadoras, extensiones, teléfonos móviles, o ser gestionadas por el propio sistema, ofreciendo con eficacia una atención de calidad en cuestiones que no requieren de la intervención de un operador.*
- ❖ Ofrece un método de tratamiento homogéneo de las llamadas telefónicas ofreciendo una respuesta eficaz al llamante y por tanto, una mejor atención telefónica.*
- ❖ Transferencia supervisada o no del comunicante a una extensión. Programación de llamadas salientes.” (7)*

Este sistema no permite su integración a Asterisk

1.4 Sistemas de Operadoras en línea en Cuba

1.4.1 Sistema de Operadoras en línea (YSOPE)

“YSOPE es una aplicación desktop que realiza el tratamiento del tráfico de operadora asociado a una gestión informática de los tickets. Le ofrece a la ejecutiva la posibilidad de distribuir las tareas según la hora, el volumen de tráfico, la cantidad de ejecutivas o supervisores presentes. Permite adecuar los recursos del personal a las necesidades de tráfico, realiza funciones de centro manual y semiautomático para el tráfico nacional e internacional así como para el tráfico de información y reclamaciones. Funciona de modo autónomo en las redes existentes independientemente de la central a la que se interconecte.” (8)

El sistema no posee un mecanismo de filtrado para los reportes que permita recoger solo los datos relevantes de cada llamada.

YSOPE es un sistema totalmente dependiente de su hardware el cual ya no es producido por ningún fabricante de equipos de telecomunicaciones. Esto impide que otros sistemas o servicios puedan ser integrados a él o viceversa.

1.5 Metodologías de Desarrollo:

Las metodologías de desarrollo son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevos software.

Existen varios tipos de metodologías, las tradicionales como RUP (Rational Unified Process) que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán, estas han resultado ser efectivas en un gran número de proyectos.

Por otra parte, se encuentran las metodologías ágiles como Extreme Programming (XP), Scrum, Crystal, Evolutionary Project Management (Evo), Feature Driven Development (FDD), Adaptive Software Development (ASD) las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas, actualmente se dice que están revolucionando la manera de producir software.

En la actualidad es de suma importancia seleccionar una metodología de desarrollo de software que esté acorde con las necesidades que requiera un proyecto. Pues es necesario tener en cuenta las propiedades de la metodología a seleccionar, para tener una visión clara de su funcionamiento respecto a los objetivos del proyecto. Para llevar a cabo esta selección se debe tener en cuenta una serie de parámetros como son las características del equipo de desarrollo y características del cliente.

El equipo de desarrollo define las siguientes características para guiar el proceso de selección de la metodología de desarrollo.

Tabla 1 Características del equipo de desarrollo.

Características	Valoración
Composición del equipo de desarrollo	2 personas ✓ 2 desarrolladores
Experiencia productiva	(baja, media, alta) ✓ media
Conocimientos de Metodologías Ágiles.	(bajo, medio, avanzado) ✓ medio

Conocimientos de las técnicas de programación.	(bajo, medio, avanzado) ✓ medio
---	--

Tabla 2 Características del cliente.

Características	Valoración
Cliente	Se encuentra fuera de la universidad, pero interactúa de forma frecuente con el equipo.
Relación con el cliente	No forma parte del equipo de desarrollo.
Disponibilidad de tiempo	Ilimitada
Retroalimentación cliente-equipo desarrollo	(continua, discontinua) ✓ continua

Se decide utilizar la metodología ágil XP para el desarrollo del presente trabajo ya que se adapta al desarrollo del proyecto y a las condiciones en que se realiza el mismo además de ser la metodología escogida para el desarrollo de la línea de CallCenter a la cual pertenece el proyecto.

1.5.1 Metodología de software seleccionada

La Programación Extrema (XP) es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Surge como respuesta y posible solución a los problemas derivados del cambio en los requerimientos, se plantea como una metodología a emplear en proyectos de riesgo, y esta aumenta la productividad del equipo. Esta metodología posee las siguientes características:

- ❖ Plantea que se trabaje directamente con el cliente haciendo pequeñas iteraciones donde no existe más documentación que el propio código.
- ❖ Utiliza historias de usuarios (HU) en vez de requisitos y casos de uso como en RUP. Permitiendo un trabajo en parejas donde los programadores menos experimentados aprenden de los más experimentados.

- ❖ Define un estándar de codificación, posibilitando que en vez de que los programadores desarrollen cada uno en su propio estilo lo hagan sobre uno solo, logrando uniformidad y organización.
- ❖ Plantea un desarrollo iterativo e incremental: Lo cual permite hacer pequeñas mejoras, unas tras otras.
- ❖ Define la programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.

Razones de la elección:

- ❖ El equipo de desarrollo posee mediana experiencia en el trabajo con esta metodología y tiene a su alcance la documentación suficiente para desarrollar el proyecto.
- ❖ Es un proyecto donde todo el trabajo lo realizan dos programadores.
- ❖ Los requisitos tienden a cambiar frecuentemente, así, conforme avanza el trabajo, el cliente puede agregar nuevas HU, dividirlos o simplemente eliminarlos.
- ❖ El cliente está integrado al equipo de desarrollo por lo que se logra una mayor retroalimentación y un producto que satisfaga todas las necesidades del mismo.
- ❖ No es necesario generar tanta documentación pues es un proyecto pequeño centrado en el desarrollo del mismo.
- ❖ Se tiene la posibilidad de probar lo que se desarrolla permitiendo la integración una vez terminado.

1.6 Sistema Gestor de Base de Datos

En la actualidad es necesario mantener almacenada la información que a diario es procesada por las diferentes empresas e instituciones. Existen diferentes formas de almacenar esta información ya sea en formato duro o formato digital, pero no cabe duda que una de las más conocidas y seguras sean las bases de datos.

Una base de datos es un almacén que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

“Una base de datos es el conjunto de datos informativos organizados en un mismo contexto para su uso y vinculación.” (9)

“Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos.” (10)

.1.6.1 MySQL

“MySQL es un potente servidor de base de datos SQL, ligero y preparado para manejar grandes cantidades de datos a la mayor velocidad posible.” (11)

MySQL es un SGBD relacional, multihilo y multiusuario, que funciona en diferentes plataformas y sencillo de usar. Es gratis para aplicaciones no comerciales.

Además esta es la base de datos que usa Asterisk por defecto.

1.7 Herramientas de Desarrollo

1.7.1 Lenguaje programación del lado del servidor

1.7.1.1 Lenguaje de Programación PHP

“PHP es un lenguaje de programación usado normalmente para la creación de páginas web dinámicas y conectadas a base de datos. PHP es un acrónimo recursivo que significa (Hypertext Pre-processor) inicialmente PHP Tools, o, Personal Home Page Tools, y se trata de un lenguaje interpretado.” (12)

“Es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML.” (13) Este lenguaje es utilizado para la creación de páginas web dinámicas utilizado fundamentalmente para la programación del lado del servidor.

1.7.2 Lenguaje de programación del lado del cliente

HTML 4

“HTML es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web. El HTML es un lenguaje de marcación de elementos para la creación de documentos hipertexto, muy fácil de aprender, lo que permite que cualquier persona, aunque no haya programado en la vida, pueda enfrentarse a la tarea de crear una web.” (14) Este lenguaje es utilizado para el diseño de las interfaces correspondientes a la aplicación.

JavaScript

“JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java.” (15) Durante el desarrollo de la presente aplicación se utiliza este lenguaje para el desarrollo de las ventanas de confirmación de algunas de las operaciones realizadas.

CSS

“CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.” (16) Estas hojas de estilo permiten mantener un diseño estándar de la aplicación.

1.7.3 Symfony 1.4

“Symfony es un framework PHP que facilita el desarrollo de las aplicaciones web. Symfony se encarga de todos los aspectos comunes y aburridos de las aplicaciones web, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto. Symfony aumenta

exponencialmente la productividad y ayuda a mejorar la calidad de las aplicaciones web aplicando todas las buenas prácticas y patrones de diseño que se han definido para la web.” (17)

Este framework posee las siguientes características:

- Fácil de instalar y configurar
- Independiente del sistema gestor de bases de datos.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Código fácil de leer que incluye comentarios de php_Documentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

1.7.4 Visual Paradigm 8.0

“Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.” (18) Esta herramienta es utilizada para modelar el sistema.

1.7.5 NetBeans 7.0.1

“NetBeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.” (19)

1.7.6 Quanta Plus 3.5

“Quanta Plus es una aplicación gratuita y de código abierto para GNU/Linux que nos permitirá desarrollar nuestra web de forma fácil y sencilla, gracias a su versatilidad y a su facilidad de uso. Permite trabajar tanto en modo visual como en modo editor de código, poniendo a nuestra disposición las herramientas necesarias para trabajar en cada uno de ellos. También posee analizador de sintaxis, generador automático de etiquetas, marcadores, corrección ortográfica y editores de formularios, listas y tablas.” (20) Esta herramienta se utiliza confeccionar las vistas de la aplicación.

1.7.7 Servidor web Apache 2.2

“Apache 2.2 es un servidor web de software libre desarrollado por la Apache Software Foundation cuyo objetivo es servir o suministrar páginas web (en general, hipertextos) a los clientes web o navegadores que las solicitan.” (21)

Apache es un servidor web que continuamente se está actualizando y es multiplataforma

1.7.9 BPM

“Business Process Management (BPM), o BPMS (BPM Suite) es el conjunto de servicios y herramientas que facilitan la administración de procesos de negocio. Por administración de procesos entendemos: análisis, definición, ejecución, monitoreo, y control de los procesos.

BPM además contempla soporte para interacción humana, e integración de aplicaciones, y es aquí la diferencia fundamental con la tecnología de WorkFlow existente, que es que BPM integra en los flujos a los sistemas.

En la práctica un flujo BPM (o modelo de proceso BPM) visualmente es muy parecido a un WorkFlow, la diferencia está en que en que uno puede notar que ciertas actividades son realizadas por personas, y otras son actividades sistematizadas (realizadas por sistemas), y ambas aparecen en el flujo.

El otro “valor agregado” de BPM es que ofrece una solución completa, que abarca todo el ciclo de vida de un proceso de negocio: análisis, modelamiento, ejecución y monitoreo de los procesos.” (22)

1.7.10 JQuery

“jQuery es considerado un Framework de Javascript, o ambiente de desarrollo. Lo que no es más que un conjunto de utilidades las cuales no necesitan ser programadas, de hecho ya fueron programadas, probadas y podemos utilizarlas de una manera muy simplificada.

En otras palabras, podremos lograr los mismos resultados, en menos tiempo sin necesidad de programar una funcionalidad completamente.” (23)

1.8 Conclusiones

En el presente capítulo se analizaron los principales conceptos, sistemas de comunicación de usuarios a través de operadoras existentes en el mundo y en el país. Además, se seleccionó la metodología, sistema gestor de base de datos, lenguaje de programación y herramientas de desarrollo necesarias para desarrollar la aplicación dependiendo de las características de la misma.

Capítulo 2. Exploración y Propuesta del Sistema

2 Introducción

En este capítulo se caracteriza el sistema que se desea desarrollar, además se identifican y comprenden los procesos relacionados con el Sistema de Operadoras en línea. Como parte de la exploración y propuesta del sistema se plantean a grandes rasgos las Historias de Usuarios (HU) de mayor interés para la primera entrega del producto así como una propuesta del prototipo no funcional de la aplicación.

2.1 Modelado del Negocio

ETECSA actualmente ofrece el servicio de llamadas de larga y corta distancia para comunicar a dos usuarios apoyándose en el SYSOPE. Con este sistema la ejecutiva recibe las llamadas en su consola con un formato de ticket en donde se introducen los datos necesarios para la confección del mismo (clasificación, número de ticket, número que llama, número al que se llama, nombre de quien llama, nombre de la persona a quien llama, número que paga, tiempo que dura la llamada) el cual se utiliza para facturar las llamadas efectivas. Para poder establecer la comunicación, la ejecutiva debe ponerse de acuerdo con las partes para ver quién paga y poder comunicarlas, de no ponerse de acuerdo las partes se termina la llamada y se cancela el ticket. Si la llamada se realiza satisfactoriamente este ticket se almacena, contándose con el tiempo hasta el envío a facturar para corregir algún dato en caso de cometerse algún error en la confección del mismo.

A continuación se modela el proceso antes descrito:

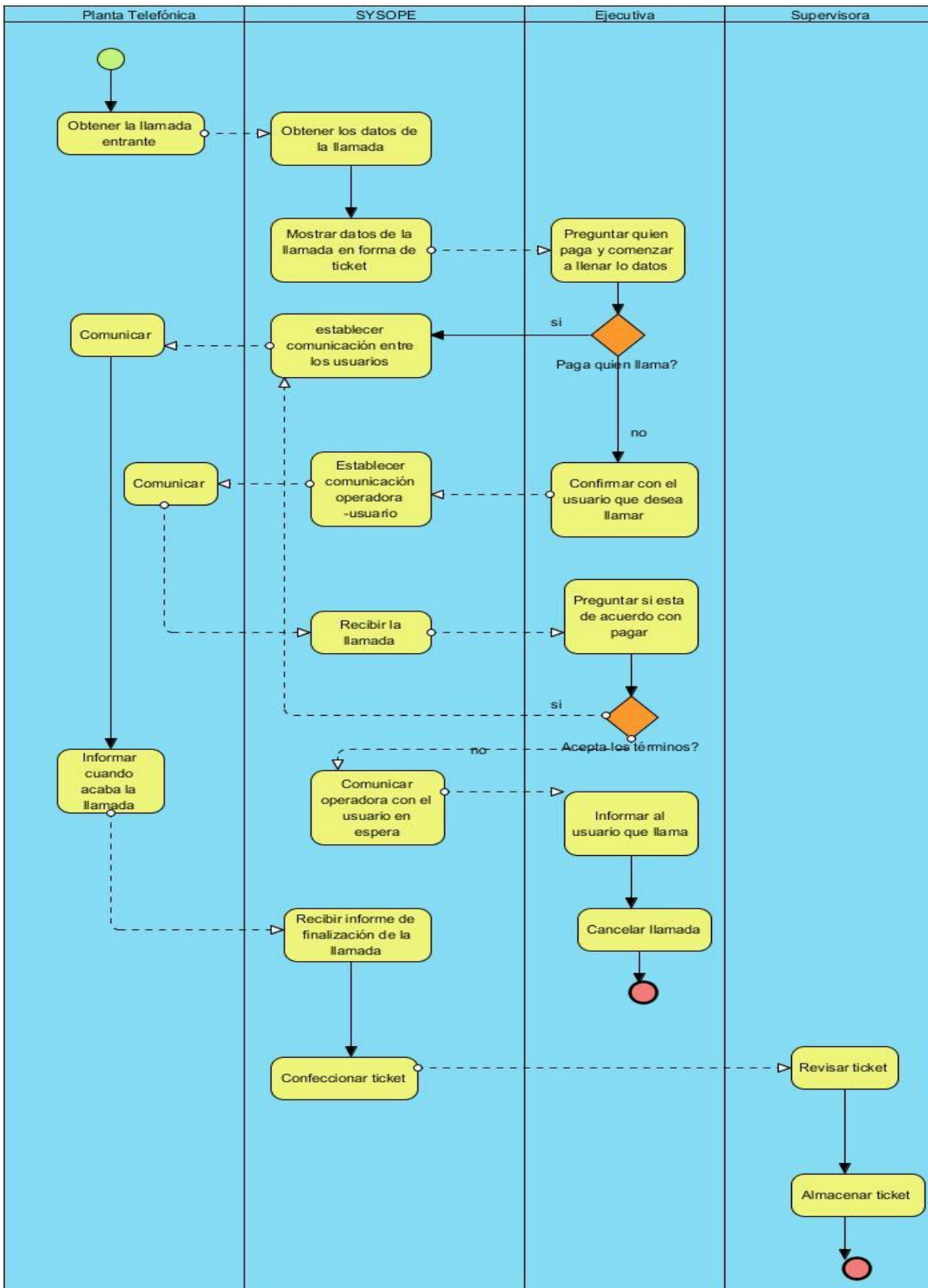


Ilustración 1 Modelo del Negocio

2.2 Propuesta del Sistema

La solución propuesta está diseñada para ETECSA en pos de prestar el servicio de operadoras en línea para comunicar dos usuarios mediante una llamada telefónica sea de corta o larga distancia, el mismo estará integrado a Asterisk como software que provee todas las funcionalidades de una central telefónica. El presente servicio de operadoras en línea permite al comunicarse dos usuarios, establecer quién paga la llamada y generar un ticket para la facturación de la misma. El ticket generado podrá ser modificado por las ejecutivas después de confeccionado. El sistema gestor de base de datos que utilizará será MYSQL. El sistema está compuesto por una aplicación Web con la cual las ejecutivas van a interactuar además de un servicio que actuará como intermediario entre el Asterisk y la aplicación.

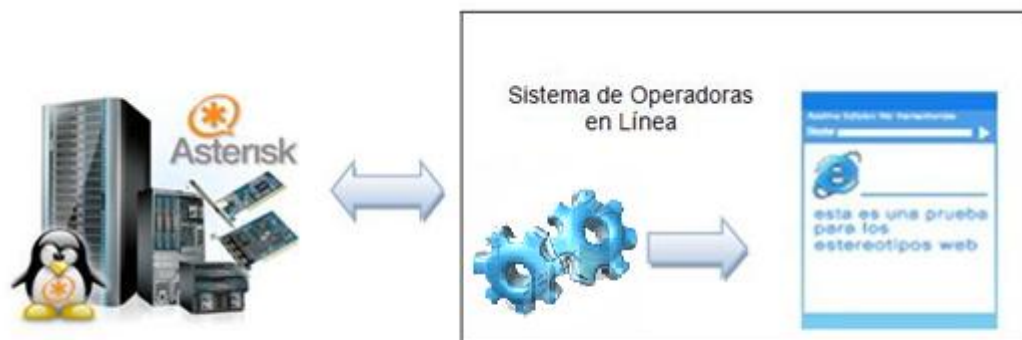


Ilustración 2 Propuesta del Sistema

Servicio de Operadoras en Línea:

Se inicia con la entrada de una llamada al Asterisk perteneciente a la cola del sistema, la cual es redireccionada automáticamente hacia la posición de ejecutiva que este libre, en ese momento el sistema debe haber sido capaz de capturar y almacenar los datos de esa llamada que luego serán enviados a la aplicación web como respuesta a las peticiones de la misma mostrándose el número del

usuario que llama en la interfaz de la ejecutiva que atenderá dicha llamada. Se le pregunta al usuario a que número quiere llamar y quien paga la llamada, si el usuario que llama va a pagar ella los comunica inmediatamente, en caso de que sea el cliente a quien se desea llamar quién pagará entonces se comunica con él para confirmar. Si el cliente acepta los términos se procede a comunicar ambos usuarios desde la propia interfaz de la ejecutiva. De lo contrario la ejecutiva cancela la llamada. Cada llamada lleva asociada un ticket de factura que se genera de forma automática y se visualiza en la interfaz de la ejecutiva que introduce los datos necesarios (número que llama, nombre de la persona que llama, número al que se llama, nombre de la persona a llamar, número que paga, nombre de la persona que paga, tiempo que dura la llamada, hora de inicio de la llamada, si es de cita, si fue cancelado, hora de la cita, agente que atendió la llamada) el cual es almacenado siempre que concluya la misma. Una vez generado el ticket la ejecutiva podrá acceder y modificar el mismo en caso de ser necesario. El sistema será capaz de depositar todos los ticket a una hora determinada en un servidor ftp para que puedan ser facturados posteriormente.

A continuación se muestra el modelo del siguiente proceso:

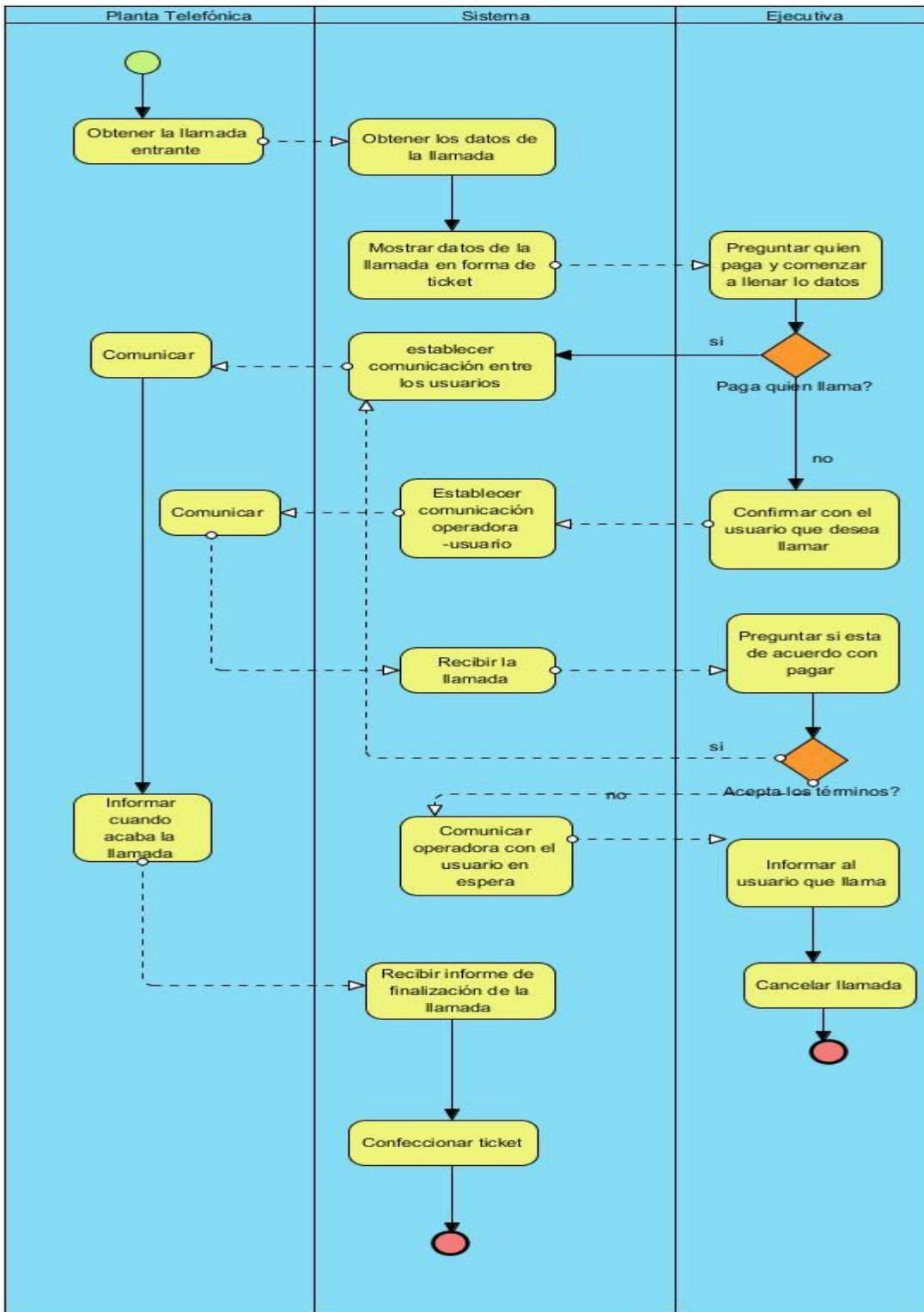


Ilustración 3 Servicio de Comunicación

2.3 Lista de reservas del sistema de operadoras en línea.

El Sistema de Operadoras en línea debe ser capaz de permitir:

- **Autenticar Usuarios**
- **Cargar configuración**
- **Gestionar ticket**
 - 1) Crear ticket
 - 2) Modificar ticket
 - 3) Buscar ticket
 - 4) Almacenar ticket
- **Llamar usuario**
- **Cancelar llamada**
- **Comunicar usuarios**
- **Enviar ticket a facturar**
- **Establecer cita.**
- **Realizar reportes**
 - 1) Mostrar la cantidad de llamadas establecidas por cada ejecutiva
 - 2) Mostrar la cantidad de llamadas canceladas por cada ejecutiva
 - 3) Mostrar actividad del sistema por horas
 - 4) Mostrar actividad de una ejecutiva específica.

Para un correcto funcionamiento del sistema se deben tener en cuenta la:

2.3.1 Usabilidad

Es necesario tener una preparación previa para operar con el sistema. Se requiere un nivel medio o alto de conocimientos de computación, aunque el manejo de la aplicación es sencillo, permitiendo la fácil comprensión por el usuario.

2.3.2 Disponibilidad

Es necesaria la ejecución de la aplicación las veinticuatro horas del día, para poder atender a todos los usuarios que soliciten este servicio en cualquier horario.

2.3.3 Eficiencia

La eficiencia del servicio estará determinada en su mayoría por la velocidad de obtención de los datos de las llamadas de la planta telefónica Asterisk y por el tiempo de respuesta del sistema, el cual depende de las prestaciones del servidor donde esté instalado.

2.3.4 Hardware

Para la instalación de la aplicación se debe disponer como mínimo de una computadora de 1 GB de RAM y un procesador Pentium 4 a una velocidad de 2.8 GHz.

2.3.5 Software

Se requiere la instalación del PHP 5.1. Elastix como distribución de software libre de comunicaciones unificadas que integra una interfaz web, soporta compatibilidad con sistema operativo Centos 5.5, navegadores Internet Explorer y Mozilla Firefox con su módulo Call Center instalado. Se requiere la comunicación entre el Elastix en su versión 2.0.2 o superior y Asterisk en su versión 1.6 o superior.

2.3.6 Interfaz de usuario

La aplicación propuesta poseerá una interfaz sencilla dirigida a las personas que se relacionen con el sistema. El servicio que se desarrolla es un servicio independiente que realiza sus propias funciones.

2.4 Personas relacionadas con el sistema

Se define como persona relacionada con el sistema a los administradores quienes tendrán el control del módulo de reportes, y las ejecutivas que podrán tener acceso al panel de operadoras para prestar el servicio al cliente además de realizar la gestión de los tickets correspondientes a las llamadas que entran al sistema. .

2.5 Fase de Exploración

La metodología XP propone como primera fase la de Exploración, es aquí donde se define el alcance real del sistema. Esta fase comienza por la creación de una serie de Historias de Usuario (HU), las cuales definen mediante su redacción qué es lo que verdaderamente necesita el cliente.

2.5.1 Historias de Usuario

Las HU representan una breve descripción del comportamiento del sistema, emplea terminología del cliente sin lenguaje técnico, solamente proporcionarán una breve descripción de las mismas, detalles sobre la estimación del riesgo y cuánto tiempo será empleado en su implementación, se realiza una por cada funcionalidad del sistema. Las HU permiten responder rápidamente a los requisitos cambiantes.

Dentro de la metodología XP las HU deben ser escritas por los clientes. Es el cliente el encargado de asignarle una prioridad a cada HU y es el equipo de desarrollo el encargado de asignarle un costo, este se traduce en las semanas que llevará el desarrollo de las mismas. Si las historias según lo planificado demoran en desarrollarse se sugiere dividirla en historias más pequeñas. También, es

importante destacar, que las HU nuevas pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología.

Clasificación de las Historias de Usuario

La prioridad en el negocio:

Alta: Se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.

Media: Se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

Baja: Se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

El riesgo en su desarrollo:

Alta: Cuando en la implementación de las HU se consideran la posible existencia de errores que lleven la inoperatividad del código.

Media: Cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.

Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU serán representadas mediante tablas divididas por las siguientes secciones:

- ✓ **Número:** número de la historia de usuario incremental en el tiempo.
- ✓ **Nombre de Historia de Usuario:** el nombre de la historia de usuario sería para identificarlas mejor entre los desarrolladores y el cliente.
- ✓ **Usuario:** involucrados en la ejecución de la HU.
- ✓ **Iteración Asignada:** número de la iteración.
- ✓ **Prioridad en negocio:** Alta, Media o Baja.
- ✓ **Riesgo en Desarrollo:** Alta, Media o Baja.
- ✓ **Programador responsable:** involucrados en el desarrollo de la HU.
- ✓ **Puntos estimados:** tiempo estimado que se demorará el desarrollo de la HU, debe ser menor que tres semanas.
- ✓ **Descripción:** breve descripción de la HU.
- ✓ **Observaciones:** señalamiento o advertencia del sistema.

- ✓ **Prototipo de interfaz:** Prototipo de interfaz si aplica.

El cliente y el equipo de desarrollo trabajan en conjunto para definir como agrupar las HU para su lanzamiento. A continuación se muestran las HU de alta prioridad definidas para la realización del sistema:

Tabla 3 HU #1: Autenticar Usuario

Historia de usuario	
Número: 1	Usuario: Administrador, Ejecutiva
Nombre de Historia de Usuario: Autenticar Usuario	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1/3	Iteración asignada: 2
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite al administrador autenticarse en el sistema con su usuario y contraseña y la ejecutiva con su extensión y ping.	
Observaciones: Las ejecutivas tienen que autenticarse también en el teléfono después de haberlo hecho en el sistema, para que el mismo pueda redireccionarle las llamadas hacia su teléfono.	
Prototipo de interface:	



Tabla 4 HU #2: Cargar configuración

Historia de usuario	
Número: 2	Usuario: Administrador

Nombre de Historia de Usuario: Cargar configuración	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1/3	Iteración asignada: 1
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite cargar la configuración del sistema automáticamente.	
Observaciones:	
Prototipo de interface:	

Tabla 5 HU #4: Llamar usuario

Historia de usuario	
Número: 4	Usuario: Ejecutiva
Nombre de Historia de Usuario: Llamar usuario	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1/3	Iteración asignada: 1
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite llamar a un usuario para establecer un acuerdo sobre el pago y la recepción de la llamada.	
Observaciones: Si no se logra establecer el acuerdo se procede a cancelar la llamada.	
Prototipo de interface:	

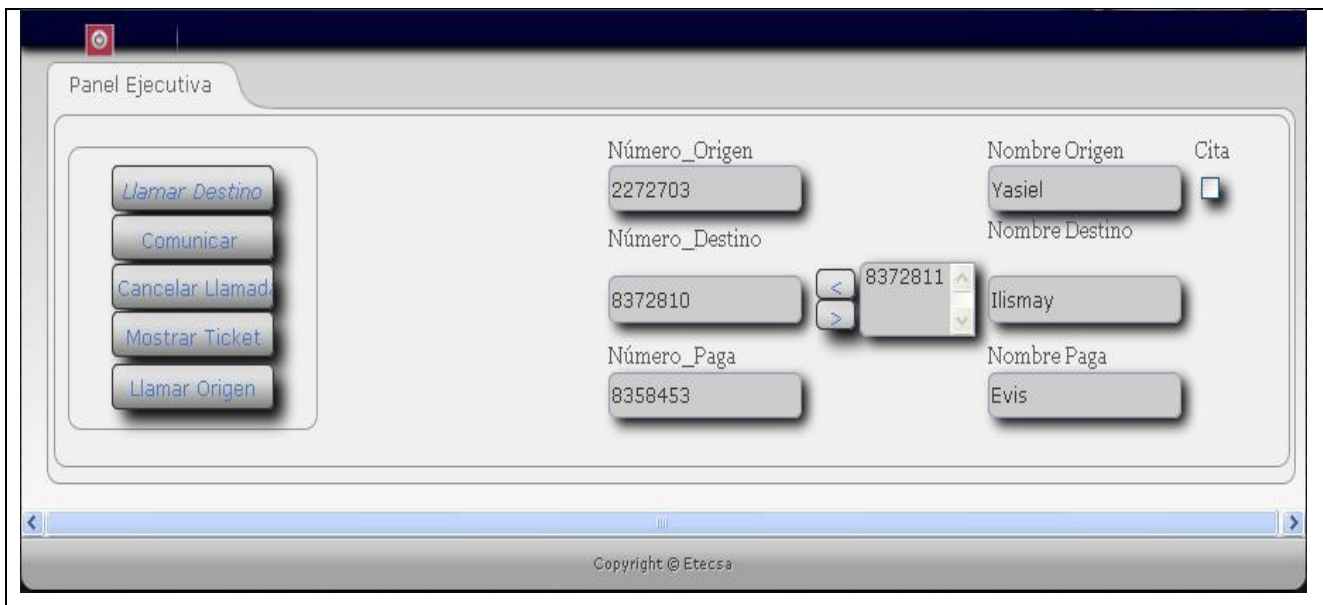


Tabla 6 HU #5: Cancelar llamada

Historia de usuario	
Número:5	Usuario: Ejecutiva
Nombre de Historia de Usuario: Cancelar llamada	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1/3	Iteración asignada: 1
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite cancelar una llamada.	
Observaciones: Se almacena el ticket como cancelado.	
Prototipo de interface:	



Tabla 7 HU #6: Comunicar usuarios

Historia de usuario	
Número:6	Usuario: Ejecutiva
Nombre de Historia de Usuario: Comunicar usuarios	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1/3	Iteración asignada: 1
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite establecer la comunicación entre el usuario que llama y el destinatario.	
Observaciones: No permite establecer la comunicación sin haber determinado quien paga la llamada.	
Prototipo de interface:	

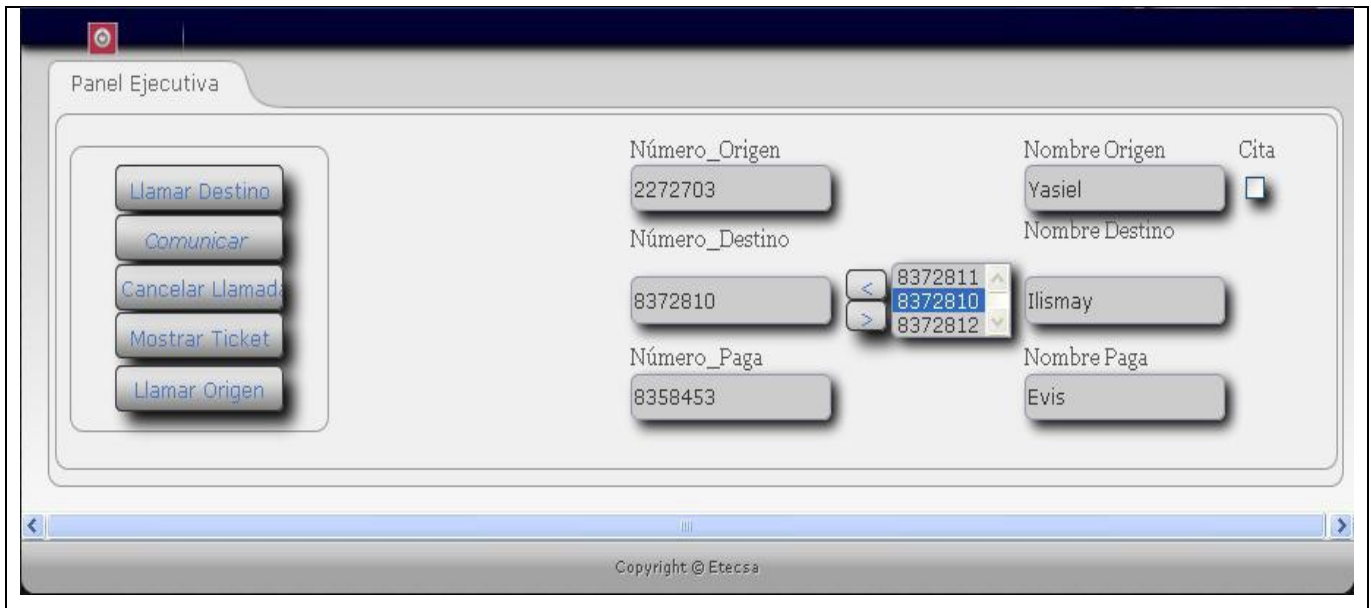


Tabla 8 HU #10: Enviar ticket a facturar.

Historia de usuario	
Número: 10	Usuario: Sistema
Nombre de Historia de Usuario: Enviar ticket a facturar.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1/3	Iteración asignada: 2
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite enviar los ticket a un ftp para ser facturados a una hora determinada, teniendo en cuenta los parámetros de acceso al FTP.	
Observaciones:	
Prototipo de interface:	

Tabla 9HU #11: Establecer cita.

Historia de usuario	
Número: 11	Usuario: Ejecutiva
Nombre de Historia de Usuario: Establecer cita	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1/3	Iteración asignada: 1
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite a la ejecutiva establecer una llamada de cita reservada con anterioridad. Este tipo de llamadas son solicitudes que hacen los usuarios para que los comuniquen con otra persona a una hora determinada.	
Observaciones:	
Prototipo de interface:	

Tabla 10HU #12: Mostrar la cantidad de llamadas establecidas por cada ejecutiva.

Historia de usuario	
Número: 12	Usuario: Sistema
Nombre de Historia de Usuario: Mostrar la cantidad de llamadas establecidas por cada ejecutiva.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta

Puntos estimados: 1/3	Iteración asignada: 2
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite mostrar un gráfico al administrador del sistema con la cantidad de llamadas establecidas por cada ejecutiva.	
Observaciones:	
Prototipo de interface:	

Tabla 11 HU #14: Mostrar la actividad del sistema por hora.

Historia de usuario	
Número: 14	Usuario: Sistema
Nombre de Historia de Usuario: Mostrar la actividad del sistema por hora.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta

Puntos estimados: 1/3	Iteración asignada: 2
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite mostrar un gráfico al administrador del sistema con la actividad del sistema por hora.	
Observaciones:	
Prototipo de interface:	

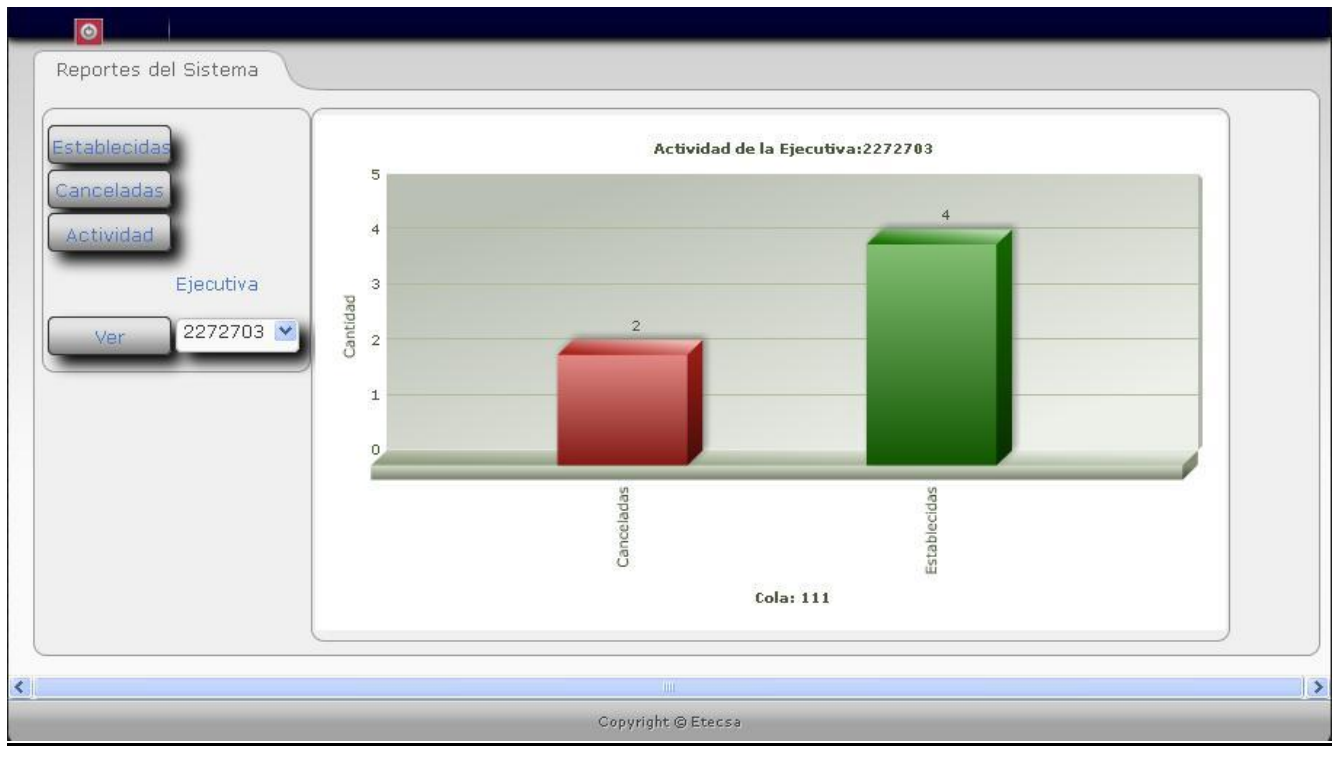
Tabla 12 HU #15: Mostrar actividad de una ejecutiva específica.

Historia de usuario	
Número: 15	Usuario: Sistema
Nombre de Historia de Usuario: Mostrar actividad de una ejecutiva específica.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1/3	Iteración asignada: 2
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	

Descripción: Permite mostrar un gráfico al administrador del sistema con la cantidad de llamadas establecidas y canceladas de una ejecutiva.

Observaciones:

Prototipo de interface:



2.6 Planificación

La actividad de planeación en la metodología XP comienza con la creación de una serie de HU que describen las características y funcionalidades requeridas para el software a construir. Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar cada HU. Este se expresa utilizando como medida el punto.

Un punto se considera como una semana ideal de trabajo donde los miembros del equipo de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de la HU.

2.6.1 Estimación de esfuerzo por Historias de Usuario

A continuación se muestra la estimación del esfuerzo por cada HU propuesta para el desarrollo de la aplicación:

Tabla 13 Estimación de esfuerzo por HU

Historia de usuario	Puntos de Estimación
1- Autenticar usuarios	1
2- Cargar configuración	1
3- Crear ticket	1
4- Modificar ticket	1
5- Buscar ticket	1
6- Almacenar ticket	1
7- Llamar usuarios	1
8- Cancelar llamada	1
9- Comunicar usuarios	1
10- Enviar ticket a facturar	1
11- Establecer cita.	1
12- Mostrar la cantidad de llamadas establecidas por cada ejecutiva	1
13- Mostrar la cantidad de llamadas canceladas por cada ejecutiva.	1
14- Mostrar la actividad del sistema por hora.	1
15- Mostrar la cantidad de llamadas establecidas y canceladas por una	1

ejecutiva	
-----------	--

2.6.2 Plan de Iteraciones

Después de identificadas y describir las HU y estimar el esfuerzo dedicado a la realización de cada una de ellas, se procede a la planificación de la fase de implementación estableciendo una división de dos iteraciones.

2.6.3 Iteración 1

En esta iteración se lleva a cabo el desarrollo de las HU del número 2 al número 9. Esto representa que al terminar la iteración se ha implementado un 57% de la aplicación.

2.6.4 Iteración 2

En esta iteración se lleva a cabo el desarrollo de la HU del número 1 y de la 11 a la 14. Esto representa que al terminar la iteración se ha implementado un 43% de la aplicación.

2.6.5 Plan de duración de las iteraciones

El plan de duración de las iteraciones se encarga de mostrar las HU en el orden en que se implementarán en cada iteración así como la duración estimada de las mismas.

Tabla 14 Plan de duración de las iteraciones

Iteración	Orden de la HU a implementar	Duración total
1	Cargar configuración Crear ticket Modificar ticket Buscar ticket Almacenar ticket Llamar usuario Cancelar llamada	9 semanas

	<p>Comunicar usuarios</p> <p>Establecer cita</p>	
2	<p>Autenticar usuarios</p> <p>Enviar tique a facturar</p> <p>Mostrar la cantidad de llamadas establecidas por cada ejecutiva</p> <p>Mostrar la cantidad de llamadas canceladas por cada ejecutiva.</p> <p>Mostrar la actividad del sistema por hora.</p> <p>Mostrar la cantidad de llamadas establecidas y canceladas por una ejecutiva</p>	6 semanas
		<u>Total:</u> 15 semanas

2.7 Conclusiones

En este capítulo se describe la propuesta del sistema a desarrollar. Se identificaron las funcionalidades y requisitos que el sistema debe cumplir, así como la descripción de las HU divididas por iteraciones y la planificación del esfuerzo dedicado a la realización de cada una de ellas en el orden en que se les dará cumplimiento según las necesidades del cliente.

Capítulo 3. Diseño del Sistema

3 Introducción

En este capítulo se describe la fase de diseño, como parte de la metodología de desarrollo XP. Se identifican y organizan las clases relevantes para las funcionalidades del sistema así la arquitectura y los patrones empleados en el desarrollo del sistema. Se procede a realizar el diseño de la base de datos.

3.1 Arquitectura

Según el estándar ANSI/IEEE Std 1471-2000 una arquitectura es *“la organización fundamental de un sistema, compuesta por sus componentes, las relaciones entre ellos y su ambiente y los principios que gobiernan su diseño y evolución”*. (24)

“La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.” (25) Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos. Los clientes son estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red.

Una representación gráfica de este tipo de arquitectura sería la siguiente

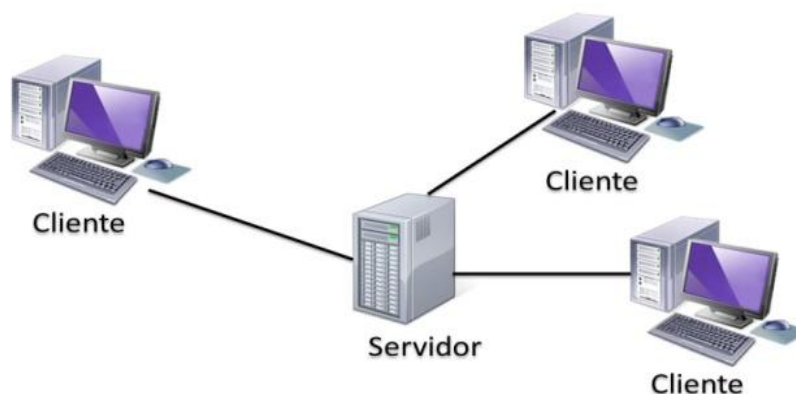


Ilustración 4 Arquitectura cliente servidor.

3.2 Patrones Arquitectónicos

El desarrollo de software ha adquirido un notable auge trayendo consigo el aumento de la complejidad de los mismos. El uso de los patrones arquitectónicos es una solución a los problemas comunes que se presentan durante el desarrollo de sistemas de software, debido a que dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. “Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos.” (26) Para el desarrollo de este sistema el patrón de arquitectura que se seguirá es el Modelo -Vista -Controlador (MVC) para la aplicación web y para el servicio N-Capas.

3.2.1 Patrón Arquitectónico Modelo Vista Controlador

MVC es un patrón arquitectónico usado para Web, donde sus siglas significan Modelo-Vista-Controlador. Este patrón arquitectónico se encuentra estructurado por 3 capas:

- ❖ **Capa Modelo:** Está constituida por clases cuya función es acceder a los datos y es donde se implementa la lógica de negocio.
- ❖ **Capa Vista:** Esta capa está compuesta por el código HTML que constituye la interfaz de usuario, los estilos CSS y los script.
- ❖ **Capa Controlador:** Esta responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista. Recibe los datos de las vistas a través de los métodos GET y POST que poseen sus formularios y envía respuestas a las vistas. Esta capa está constituida por archivos de clases.



Ilustración 5 Modelo Vista Controlador

El framework **Symfony** utilizado para el desarrollo de la aplicación establece una distribución interna de carpetas para representar la arquitectura MVC. La carpeta **templates** define la capa vista, la carpeta **models** define la capa modelo y la carpeta **actions** define la capa control.

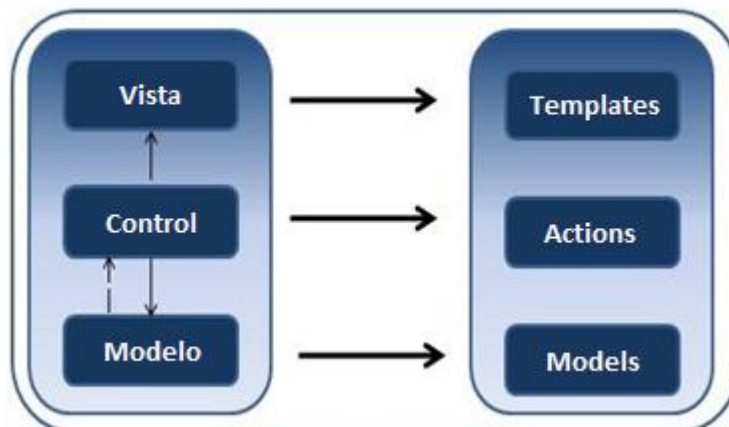


Ilustración 6 Symfony - Modelo Vista Controlador

3.2.2 Patrón Arquitectónico N-Capas.

“En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). “ (27). Esta patrón arquitectónico se aplica específicamente con dos capas, negocio y acceso a datos, para el servicio intermediario entre la planta telefónica y la aplicación web debido a que el mismo no presenta interfaz alguna.

Capa de negocio: es donde se implementa la lógica de negocio.

Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por el gestor de base de datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.



Ilustración 7 Arquitectura N-Capas

3.3 Diagrama de Clases

A continuación se presentan los diagramas de clases divididos por capas, tanto del servicio que actúa como intermediario entre Asterisk y la aplicación web que utiliza el patrón arquitectónico N-Capas como de la propia aplicación Web que utiliza MVC sobre el framework **Symfony**.

3.3.1 Diagrama de Clases del Servicio Capa Negocio

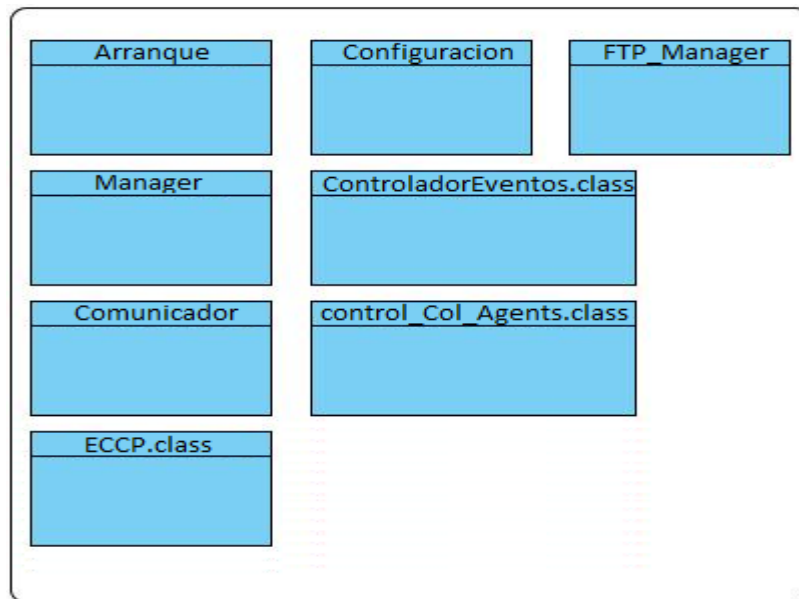


Ilustración 8 Capa Negocio

Arranque Se encarga de iniciar el sistema.

Configuración: Se encarga de cargar la configuración del sistema.

ControladorEventos.class: Cuenta con una instancia de la clase Configuración.class y control_Col_Agents.class y se encarga del manejo de los eventos y las peticiones del cliente además de llevar el control del tiempo de duración de las llamadas y de la activación de las citas.

Comunicador: Posee una instancia de la clase Manager y se encarga de iniciar la comunicación con Asterisk y la relación con el mismo.

Manager: Se encarga de establecer comunicación con Asterisk.

control_Col_Agents.class: Se encarga de manejar la información necesaria para el servicio relacionada con los agentes del sistema.

FTP_Manager: Se encarga de realizar la conexión con el servidor FTP y de guardar en el mismo el ticket.

ECCP.class: Se encarga de acceder y gestionar la información de todos los agentes.

Capa de Acceso a Datos

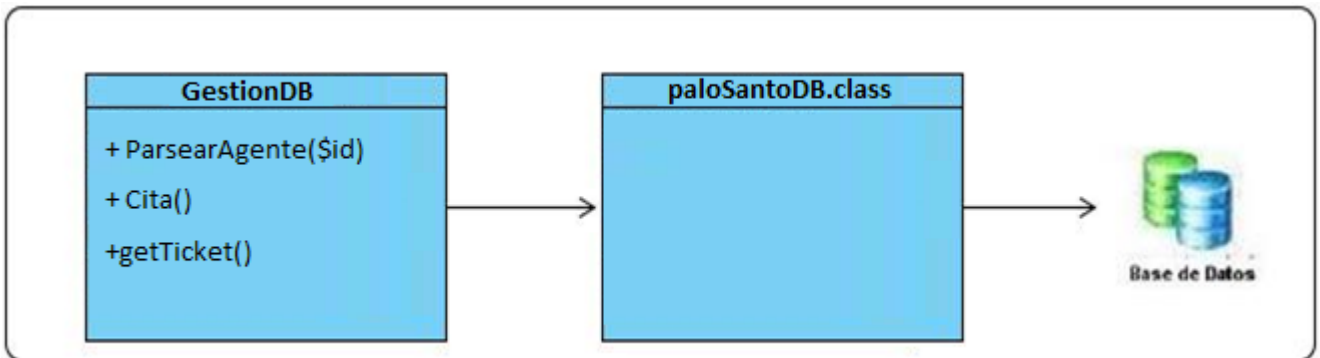


Ilustración 9 Capa Acceso a Datos

GestionDB: Realiza las consultas necesarias para cargar los tickets y buscar los agentes.

paloSantoDB.class: Realiza la conexión con la base de datos.

3.3.2 Diagrama de Clases de la Aplicación Web.

Capa Vista

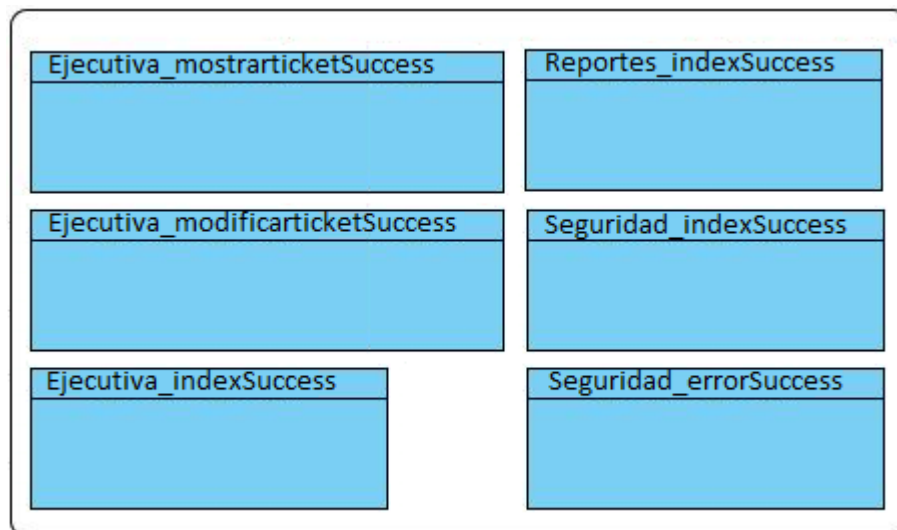


Ilustración 10 Capa Vista

Ejecutiva_mostrarticketSuccess, **Ejecutiva_modificarticketSuccess**, **Ejecutiva_indexSuccess**, **Reportes_indexSuccess**, **Seguridad_indexSuccess** y **Seguridad_errorSuccess** estos archivos se encuentran dentro de la carpeta templates de cada módulo al que pertenecen la cual define la capa

vista del patrón arquitectónico MVC en el framework **Symfony**. Los mismos definen las plantillas HTML, CSS, JavaScript y su extensión es **.php**.

Capa Controladora

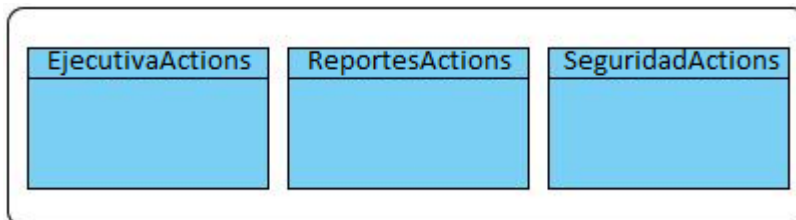


Ilustración 11 Controladora

EjecutivaActions: Maneja y controla todo el flujo de acciones del módulo Ejecutiva (llamar usuario, comunicar usuarios, cancelar ticket, mostrar ticket, modificar ticket).

ReportesActions: Maneja y controla todo el flujo de acciones del módulo Reportes (Mostrar cada uno de los reportes).

SeguridadActions: Maneja y controla todo el flujo de acciones del módulo Seguridad (autenticar usuarios, cargar datos automáticos).

Capa Modelo

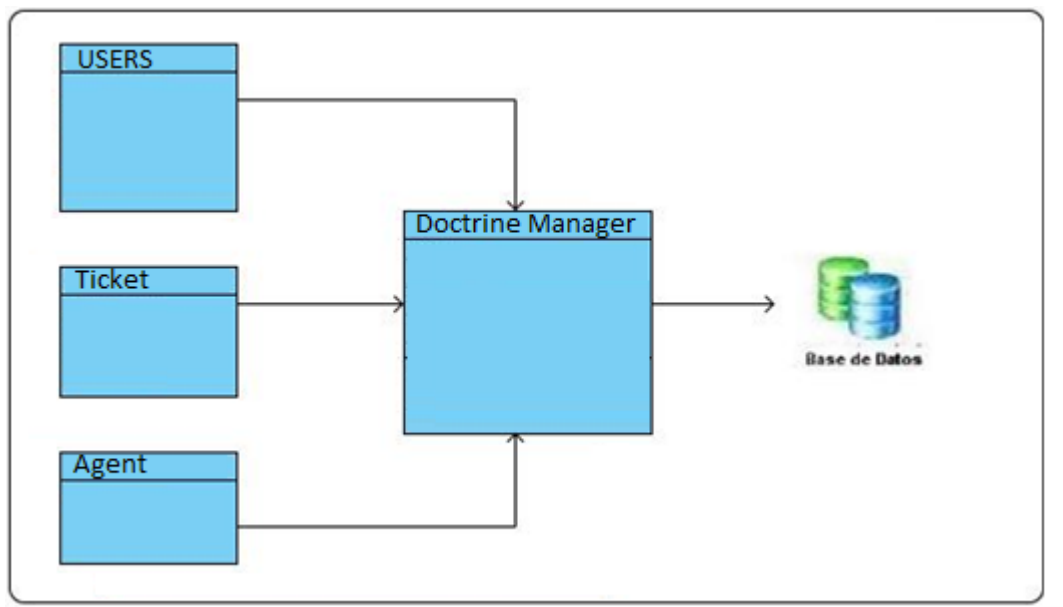


Ilustración 12 Capa Modelo

USERS: Gestiona las extensiones de los usuarios.

Ticket: Se encarga de gestionar los tickets.

Agent: Gestiona a los agentes.

Doctrine Manager: Este es el ORM que se utiliza para trabajar con la base de datos.

3.4 Patrones de Diseño

“Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.” (28)

Proponen una forma de reutilizar la experiencia de los programadores.

- Contribuyen a dar flexibilidad y extensibilidad a los diseños.
- Contribuyen a reutilizar diseño, identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones.
- Mejoran (aumentan, elevan) la flexibilidad, modularidad y extensibilidad. Incrementan el vocabulario de diseño, ayudando a diseñar desde un mayor nivel de abstracción.

3.4.1 Patrones para Asignar Responsabilidades (GRASP)

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades.

Para el desarrollo de este sistema se utilizaron los siguientes patrones GRASP:

- **Experto**

Problema: ¿Cómo lograr que cada clase cumpla con la responsabilidad que le corresponde?

Solución: Asignar la responsabilidad a la clase que tiene la información necesaria para cumplir con la responsabilidad.

Ejemplo:

Las clases **FTP_Manager** y **control_Colas_Agents.class** cuentan con la información necesaria para cumplir cada una las responsabilidades que le corresponden.

- **Creador**

Problema: ¿Qué clase debería ser la responsable de crear una nueva instancia de alguna clase?

Solución: Asignarle a una clase la responsabilidad de crear una instancia necesaria de otra clase.

Ejemplo:

La clase **GestionDB** que realizan la conexión a la base de datos serán las responsables de crear una nueva instancia de la clase **paloSantoDB**.

- **Controlador**

Problema: ¿Quién debería encargarse de atender un evento del sistema?

Solución: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una o varias clases.

Ejemplo:

Las clases **ReporteActions**, **EjecutivaActions**, **ErrorActions** son algunas de las clases encargadas de atender a los eventos que lanza el sistema en sus respectivas interfaces y realizar la operación correspondiente.

- **Alta Cohesión**

Problema: ¿Cómo mantener la complejidad dentro de los límites manejables?

Solución: Asignar responsabilidades manteniendo alta la cohesión.

Ejemplo:

Las clases **FTP_Managery** **GestionDB** se les asignarán responsabilidades propias con el objetivo que trabajen en la misma área de aplicación y que no tengan mucha complejidad.

- **Bajo Acoplamiento**

Problema: ¿Cómo lograr una baja dependencia y una alta reutilización de clases?

Solución: Asignar responsabilidades manteniendo bajo el acoplamiento.

Ejemplo:

A la clase **GestionDB** se les asignan responsabilidades de forma tal que solo se comunique con la clase **paloSantoDB** que se encarga de conectarse a la base de datos.

3.4.2 Patrones de diseño (GOF)

Creacionales: Solucionan problemas de creación de instancias. Ayudan a encapsular y abstraer dicha creación.

- **Singleton**

Problema: ¿Cómo lograr tener una clase sólo tenga una instancia y proporcione un punto de acceso global a ésta instancia?

Solución: La solución es crear un método dentro de la clase que instancie a la propia clase si esta ya no está instanciada.

Ejemplo:

La clase **Configuración**, implementan este patrón ya que se instancian una sola vez y permiten el acceso global a ellas.

3.5 Tarjetas Clase – Responsabilidad – Colaborador

Como complemento de la fase de diseño XP propone el modelado de Clase-Responsabilidad-Colaborador (CRC), lo que constituye un modelo simple de organizar las clases más relevantes del sistema. Este modelado CRC utiliza tarjetas para representar de forma organizada las clases.

Estas tarjetas se dividen en tres secciones:

- 1) A lo largo del borde superior de la tarjeta se escribe el nombre de la clase.
- 2) En el cuerpo de la tarjeta a la izquierda se listan las responsabilidades de la clase.
- 3) En el cuerpo de la tarjeta a la derecha los colaboradores.

Una **responsabilidad** es una acción que la clase sabe o hace. Los **colaboradores** son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad.

A continuación las tarjetas CRC correspondientes a las clases más relevantes para las funcionalidades del Servicio de Operadoras en línea:

Tabla 15 CRC: Clase Arranque

Clase Arranque	
Descripción: Clase que llama las demás clases necesarias para el funcionamiento del sistema	
Responsabilidad	Colaborador
Llamar la clase encargada de cargar la configuración del sistema	Configuración
Llamar la clase que inicia la conexión con la planta Asterisk.	Comunicador
Llamar la clase que inicia la captura de los eventos de la llamada, las peticiones del cliente.	ControladorEventos.class

Tabla 16 CRC: Clase Configuración

Clase Configuración	
Descripción: Clase que carga toda la configuración del sistema	
Responsabilidad	Colaborador
Cargar los datos necesarios para el funcionamiento del sistema	---

Tabla 17 CRC: Clase Comunicador

Clase Comunicador	
Descripción: Clase que inicia la conexión a la consola de Asterisk	
Responsabilidad	Colaborador
Establecer la conexión con Asterisk	Manager

Tabla 18 CRC: Clase ControladorEventos.class

Clase ControladorEventos.class	
Descripción: Clase que se encarga del manejo de los eventos y las peticiones del cliente.	
Responsabilidad	Colaborador
Capturar las llamadas entrantes al sistema.	Manager
Atender las peticiones del cliente.	Configuracion
Guardar los ticket en el servidor FTP	FTP_Manager
Controlar el tiempo de duración de las llamadas y la activación de las citas	control_Col_Agents paloSantoDB.class

Tabla 19 CRC: Clase FTP_Manager

Clase FTP_Manager	
Descripción: Clase encargada de conectarse al servidor ftp y guardar los ticket.	
Responsabilidad	Colaborador
Realizar conexión al servidor FTP.	Configuracion
Guardar los tickets en el servidor FTP.	Configuracion

Tabla 20 CRC: Clase control_Col_Agents.class

Clase control_Col_Agents.class	
Descripción: Clase encargada de manejar información de los agentes.	
Responsabilidad	Colaborador
Conocer el estado de los agentes.	ECCP.class, Manager

Tabla 21 CRC: Clase Manager

Clase Manager	
Descripción: Clase encargada de conectarse directamente a la consola de Asterisk y el trabajo con el mismo.	
Responsabilidad	Colaborador
Conexión con Asterisk	---

Tabla 22 CRC: Clase SeguridadActions

Clase SeguridadActions	
Descripción: Clase que controla el acceso al sistema	
Responsabilidad	Colaborador
Comprobar que el usuario y la contraseña de los usuarios sean válidos en el caso de los administradores.	sfConfig
Cargar los números de las ejecutivas y sus extensiones.	Agent,Users
Mostrar el modulo correspondiente a cada usuario	

Tabla 23 CRC: Clase ReportesActions

Clase ReportesActions	
Descripción: Clase que permite mostrar los reportes del sistema	
Responsabilidad	Colaborador
Mostrar el reporte de las Llamadas establecidas por ejecutivas	Agent,Ticket
Mostrar el reporte de las Llamadas canceladas por ejecutivas	Agent,Ticket
Mostrar el reporte de la Actividad en el sistema por horas	Agent,Ticket
Mostrar el reporte de la Actividad de una ejecutiva	Agent,Ticket

Tabla 24 CRC: Clase EjecutivaActions

Clase EjecutivaActions	
Descripción: Clase que controla todas las acciones de la ejecutiva en sistema	
Responsabilidad	Colaborador
Permitir llamar a un usuario	Agent,Manager
Permitir comunicar dos usuarios	CurrentCallEntry,Manager
Permitir cancelar un ticket	Agent,Ticket,
Permitir almacenar un ticket	Agent,Ticket
Permitir leer datos de un socket	
Permitir mostrar un ticket	Agent,Ticket
Permitir modificar un ticket	Agent,Ticket

3.6 Modelo Físico de la Base de Datos

El siguiente modelo físico de la base de datos del sistema muestra la relación que existe entre todas las tablas de la base de datos general `call_center` de Asterisk de las cuales para el desarrollo de la aplicación se utilizan `users`, `agent`, `current_callentry`, y se creó una nueva tabla llamada `ticket` para llevar el control de los tickets generados por cada llamada.

3.7 Conclusiones

En este capítulo se propone la arquitectura a utilizar en el desarrollo del sistema así como los patrones arquitectónicos y de diseño y se realizan las tarjetas CRC como elemento importante en la descripción de cada una de las clases para la posterior implementación del sistema.

Capítulo 4. Implementación y Prueba

4 Introducción

En este capítulo se explica cómo se lleva a cabo la implementación y pruebas de cada una de las historias de usuario definidas anteriormente como parte de la metodología XP. Se procede a realizar las tareas de la ingeniería y a confeccionar las pruebas del sistema.

Para la implementación de un software, XP plantea hacer varias iteraciones, obteniendo al final de cada una un producto final, que debe ser probado y mostrado al cliente. En el transcurso de cada iteración se descomponen la HU definidas por el cliente en fase de Exploración en tareas de la ingeniería que entregan a los programadores para ser implementadas durante la iteración correspondiente.

4.1 Tareas de la Ingeniería

Las tareas de la ingeniería consisten en una descripción más detallada y orientada al programador de cada una de las funcionalidades con que cuenta el sistema para una correcta implementación de las mismas.

Las tareas de la ingeniería serán representadas mediante tablas divididas por las siguientes secciones:

- **Número Tarea:** los números deben ser consecutivos.
- **Número Historia de Usuario:** número de la historia de usuario a la que pertenece la tarea.
- **Nombre Tarea:** nombre que identifica a la tarea.
- **Tipo de Tarea:** las tareas pueden ser de: Desarrollo, Corrección, Mejora, Otra (Especificar)
- **Puntos Estimados:** tiempo estimado en semanas que se le asignará a su desarrollo, no puede ser mayor que tres semanas.
- **Fecha Inicio:** fecha en que inicia el desarrollo de la tarea.
- **Fecha Fin:** fecha en que finaliza el desarrollo de la tarea.
- **Programador Responsable:** nombre y apellidos del programador.
- **Descripción:** breve descripción de la tarea.

A continuación las tareas de la ingeniería correspondientes a las HU de prioridad más alta:

Tabla 25 Tarea #1 Autenticar Usuario

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Autenticar Usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 16/1/2012	Fecha Fin: 23/1/2012
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite al administrador autenticarse en el sistema con su usuario y contraseña y la ejecutiva con su extensión y ping. Las ejecutivas tienen que autenticarse también en el teléfono después de haberlo hecho en la aplicación para que el sistema permita redireccionar las llamadas hacia su teléfono.	

Tabla 26 Tarea #2 Cargar configuración

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 2
Nombre Tarea: Configurar el sistema	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 24/1/2012	Fecha Fin: 31/1/2012
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite configurar algunos datos del sistema dependiendo de la necesidad (el acceso al Asterisk, al FTP, al gestor de base de datos). Estos datos tanto en la aplicación web como en el servicio son establecidos manualmente en sus respectivos archivos de comunicación desde donde serán cargados automáticamente al iniciar el sistema.	

Tabla 27 Tarea #4 Llamar usuario

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario:4
Nombre Tarea: Llamar usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 8/2/2012	Fecha Fin: 15/2/2012
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite llamar a un usuario para establecer un acuerdo sobre el pago de la llamada y si está de acuerdo en atender la misma. La ejecutiva deberá pasar el teléfono al que desea llamar.	

Tabla 28 Tarea #5 Cancelar llamada

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario:5
Nombre Tarea: Cancelar llamada	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 15/2/2012	Fecha Fin: 22/2/2012
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite cancelar una llamada. El ticket correspondiente a esta llamada se almacena como cancelado.	

Tabla 29 Tarea #6 Comunicar usuarios

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario:6
Nombre Tarea: Comunicar usuarios	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 23/2/2012	Fecha Fin: 1/3/2012
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite establecer la comunicación entre el usuario que llama y el destinatario, se procede a guardar automáticamente el ticket correspondiente a la misma luego de iniciar. No permite establecer la comunicación sin haber determinado quien paga la llamada.	

Tabla 30 Tarea #10 Enviar ticket a facturar

Tarea de Ingeniería	
Número Tarea: 10	Número Historia de Usuario:10
Nombre Tarea: Enviar ticket a facturar	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 3/4/2012	Fecha Fin: 10/4/2012
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite enviar a la hora establecida los ticket a un servidor FTP para luego ser facturados, teniendo en cuenta los parámetros de acceso al FTP (dirección, usuario, contraseña).	

Tabla 31 Tarea #11 Establecer cita.

Tarea de Ingeniería	
Número Tarea: 11	Número Historia de Usuario: 11

Nombre Tarea: Establecer cita	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 11/4/2012	Fecha Fin: 18/4/2012
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Pérez	
Descripción: Permite a la ejecutiva establecer una llamada de cita reservada con anterioridad, esta cita se estable mostrándole a la ejecutiva los datos de la cita en forma de ticket 15 minutos antes de la hora pactada para que esta pueda establecer la comunicación. En caso que la ejecutiva encargada de atender la cita esté ocupada se espera un minuto más y se vuelve a intentar establecer la cita.	

Tabla 32 Tarea #12 Mostrar la cantidad de llamadas establecidas por cada ejecutiva.

Tarea de Ingeniería	
Número Tarea: 12	Número Historia de Usuario: 12
Nombre Tarea: Mostrar la cantidad de llamadas establecidas por cada ejecutiva.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 19/4/2012	Fecha Fin: 26/4/2012
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Perez	
Descripción: Permite mostrar gráficamente la cantidad de llamadas establecidas por cada ejecutiva.	

Tabla 33 Tarea #14 Mostrar la actividad del sistema por hora.

Tarea de Ingeniería

Número Tarea: 14	Número Historia de Usuario:14
Nombre Tarea: Mostrar la actividad del sistema por hora.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 5/5/2012	Fecha Fin: 12/5/2012
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Perez	
Descripción: Permite mostrar gráficamente la actividad del sistema por hora.	

Tabla 34 Tarea #15 Mostrar actividad de una ejecutiva específica

Tarea de Ingeniería	
Número Tarea: 15	Número Historia de Usuario:15
Nombre Tarea: Mostrar actividad de una ejecutiva específica	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 13/5/2012	Fecha Fin: 20/5/2012
Programador responsable: Yasiel Cabrera Iglesias Evis Domínguez Perez	
Descripción: Permite mostrar gráficamente la cantidad de llamadas establecidas y canceladas de una ejecutiva en específico.	

4.2 Pruebas

Dentro de los instrumentos capaces de medir el estado de calidad de un producto se encuentran las pruebas. El proceso de pruebas se dirige fundamentalmente a componentes del software o al sistema de software en general, con el objetivo de medir hasta donde el software cumple las funcionalidades establecidas por el cliente. Las pruebas del software verifican y revelan la calidad de un producto. Son utilizadas para identificar posibles errores en la implementación, calidad, o usabilidad de un programa de software.

Propio de la metodología XP, se lleva a cabo la Fase de Prueba. Durante el desarrollo de software, XP establece probar constantemente tanto como sea posible, esto permite un aumento de la calidad del sistema desarrollado reduciendo el número de errores no detectados.

XP divide las pruebas del sistema en dos grupos:

Pruebas Unitarias: las pruebas unitarias son las encargadas de verificar el código y son diseñadas por los programadores. Cada uno de los desarrolladores tiene que ir probando constantemente lo que va obteniendo en el transcurso de la implementación de un sistema, para garantizar que las funcionalidades exigidas por el cliente estén implementadas correctamente.

Las pruebas unitarias fueron desarrolladas constantemente cada vez que se concluía la implementación de alguna funcionalidad probándola directamente en el entorno real. Todos los errores encontrados en las pruebas realizadas fueron corregidos.

Pruebas de Aceptación: las pruebas de aceptación también llamadas pruebas del cliente las especifica el cliente y se enfocan en las características generales y las funcionalidades del sistema. En estas serán probadas las funcionalidades exigidas por el cliente, descritas en las HU que se han implementado.

Las pruebas de aceptación correspondiente a cada una de las funcionalidades del Servicio de Operadoras en línea representadas mediante tablas divididas por las siguientes secciones:

Clases Válidas: se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas válidas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.

- **Clases Inválidas:** se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las posibles entradas inválidas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado y cómo responde el sistema.
- **Resultado Esperado:** se hará una breve descripción del resultado que se espera ya sea para entradas válidas o entradas inválidas.
- **Resultado de la Prueba:** se hará una breve descripción del resultado que se obtiene.
- **Observaciones:** algún señalamiento o advertencia que sea necesario hacerle a la sección que se está probando.

Las pruebas de aceptación se llevarán a cabo redactando los casos de prueba, teniendo en cuenta el orden de las HU y la prioridad que ha sido asignada a las funcionalidades. Luego se hará la planificación con el cliente de cuándo y cuáles pruebas serán llevadas a cabo, para así reunir los miembros del proyecto seleccionados para realizarlas. Finalmente, se completarán cada uno de los campos de las tablas de las pruebas de aceptación con el resultado de la prueba. Las pruebas de aceptación correspondientes a las funcionalidades del Sistema de Operadoras en línea pueden encontrarse en el anexo 1.

A continuación se presenta un caso de prueba de ejemplo.

Diseño del caso de prueba de la HU: Autenticar usuario

Clases Válidas	Clases Inválidas	Resultados esperados	Resultado de la Prueba	Observaciones
<p><u>Paso 1:</u> El usuario escoge la opción administrador e introduce su usuario y contraseña.</p> <p>Usuario: yasiel Contraseña: admin</p> <p><u>Paso 2:</u> Presiona el botón Entrar.</p>		El sistema muestra el módulo de reportes al administrador.	Satisfactorio	
<p><u>Paso 1:</u> El usuario escoge la opción ejecutiva, el sistema carga los números de ejecutivas y extensiones que existen en el sistema.</p> <p><u>Ejemplo:</u> Número: 2273-yasiel Extensión: SIP/2273</p> <p><u>Paso 2:</u> Presiona el botón</p>		El sistema muestra el módulo de ejecutiva.	Satisfactorio	

Entrar.				
<u>Paso 1:</u> El usuario escoge la opción administrador e introduce su usuario y contraseña. Usuario: root Contraseña: evis <u>Paso 2:</u> Presiona el botón Entrar.		El sistema muestra el módulo de reportes al administrador.	Satisfactorio	
	<u>Paso 1:</u> El usuario escoge la opción administrador e introduce su usuario y contraseña. Usuario: yasi* Contraseña: Admi <u>Paso 2:</u> Presiona el botón Entrar.	El sistema muestra un mensaje al usuario indicando que el campo usuario solo admite letras y la contraseña minúsculas.	Satisfactorio	
	<u>Paso 1:</u> El usuario escoge la opción ejecutiva, el sistema carga los números de ejecutivas y extensiones que existen en el sistema. <u>Ejemplo:</u> Número: 2271-pepe Extensión: SIP/2271	El sistema muestra un mensaje a la ejecutiva indicando que él ya está autenticado o su extensión está siendo usada.	Satisfactorio	

	<u>Paso 2:</u> Presiona el botón Entrar.			
--	--	--	--	--

4.3 Conclusiones

En el presente capítulo se describe el proceso de implementación del sistema detallando los patrones y arquitectura utilizados para el desarrollo del mismo, se planteó a grandes rasgos los tipos de pruebas que se le realizaron al sistema y se elaboraron los casos de prueba de aceptación que se le harán al sistema.

Conclusiones

Con el presente trabajo se lograron describir los procesos relacionados con el Sistema de Operadoras en línea una vez identificados. Se seleccionaron las herramientas y la metodología para el desarrollo del sistema, se realizó el estudio y comprensión del estilo arquitectónico así como de los patrones de diseño los cuales permitieron un mejor diseño y facilitaron la implementación del sistema. Se llevaron a cabo las pruebas del sistema en el transcurso de la implementación del mismo para corregir todos los posibles errores y fallas.

Con el desarrollo de este Sistema de Operadoras en línea para Asterisk se logró dar solución a cada uno de los principales problemas presentes en el sistema autónomo de operadoras utilizado por ETECSA tales como la dependencia del hardware, la falta de soporte, la interfaz poco amigable y la no reutilización del personal.

Por tanto se puede concluir que se le ha podido dar solución a cada uno de los objetivos específicos y por tanto al objetivo general planteado logrando erradicar el problema a resolver propuesto al inicio del trabajo.

Recomendaciones

Una vez desarrollado el sistema y expuesto las conclusiones del trabajo se recomienda que:

- ❖ El sistema además de comunicar usuarios a través de operadoras cuente con la funcionalidad de realizar la factura de los tickets sin necesidad de que esto se realice como un proceso independiente al sistema.
- ❖ El sistema permita comunicar dos usuarios por medio de IVRs también.

Trabajos citados

1. *ElastixBook-Comunicaciones_Unificadas_con_Elastix_Beta.pdf*[documento].
2. THE FREE DICTIONARY. [En línea] 1 de 11 de 2011. <http://es.thefreedictionary.com/llamada..>
3. **Pérez, Danae y Segura, Rainer.** *Servicio de Facturación para Elastix. Herramientas y Tecnologías.* Habana : s.n., 2011.
4. THE FREE DICTIONARY. [En línea] [Citado el: 1 de 11 de 2011.] <http://es.thefreedictionary.com/telefonista>.
5. Altatecnologia. [En línea] [Citado el: 2 de 11 de 2011.] http://www.altatecnologia.com.gt/plantas_telefonicas.html.
6. Comunidad de usuarios. [En línea] 3 de 11 de 2011. http://comunidad.asterisk-es.org/index.php?title=Documentos_sobre_Asterisk.
7. at4wireless. [En línea] [Citado el: 6 de 11 de 2011.] <http://www.at4wireless.com/es/ti-servicios-soluciones/automatizacion-telefonica-smartivr.html..>
8. *Descripción General del sistema SYSOPE 30001.pdf.* Habana : s.n.
9. definicionabc. [En línea] [Citado el: 17 de 11 de 2011.] <http://www.definicionabc.com/tecnologia/base-de-datos.php>.
10. Desarrolloweb. [En línea] [Citado el: 20 de 11 de 2011.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
11. Dirphp. [En línea] 27 de 11 de 2011. http://www.dirphp.com/dir/Software_y_servidores_PHP/Sitio_oficial_del_Servidor_de_Base_de_datos_MySQL_50.html.
12. Superhostin. [En línea] [Citado el: 27 de 11 de 2011.] <http://www.superhosting.cl/manuales/manual-tutorial-de-php.html>.
13. Php.ne. [En línea] [Citado el: 27 de 11 de 2011.] <http://php.net/manual/es/intro-what-is.php>.
14. Desarrolloweb. [En línea] [Citado el: 28 de 11 de 2012.] <http://www.desarrolloweb.com/articulos/que-es-html.html>.
15. Librosweb. [En línea] [Citado el: 28 de 11 de 2011.] <http://www.librosweb.es/javascript/index.html>.

16. Librosweb. [En línea] [Citado el: 28 de 11 de 2012.] <http://www.librosweb.es/css/index.html>.
17. Symfony. [En línea] [Citado el: 29 de 11 de 2011.] <http://www.symfony.es/que-es-symfony/>.
18. Freedownloadmanager. [En línea] [Citado el: 29 de 11 de 2011.]
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
19. Netbeans. [En línea] [Citado el: 29 de 11 de 2011.] http://netbeans.org/index_es.html.
20. Quanta-plus. [En línea] [Citado el: 29 de 11 de 2011.] <http://quanta-plus.uptodown.com/ubuntu>.
21. Recursostic. [En línea] [Citado el: 29 de 11 de 2011.]
<http://recursostic.educacion.es/observatorio/web/ca/software/servidores/580-elvira-mifsud>.
22. Soaagenda. [En línea] [Citado el: 29 de 11 de 2011.] [http://www.soaagenda.com/journal/articulos/que-es-bpm-que-es-bpms/..](http://www.soaagenda.com/journal/articulos/que-es-bpm-que-es-bpms/)
23. Programacionya.com. [En línea] [Citado el: 29 de 11 de 2011.] <http://programacionya.com/%C2%BFque-es-y-para-que-sirve-jquery/>.
24. Gbm. [En línea] [Citado el: 20 de 1 de 2012.]
http://www.gbm.net/bt/bt37/opinion/arquitecturas_empresariales.php.
25. solinfo. [En línea] [Citado el: 20 de 1 de 2012.] http://www.solinfo.cl/wp-content/themes/zine-style/archivos/cliente_servidor.html.
26. msdn.microsoft.com. [En línea] [Citado el: 23 de 1 de 2012.] <http://msdn.microsoft.com/es-es/library/bb972251.aspx>.
27. Diclib. [En línea] [Citado el: 23 de 1 de 2012.] http://www.diclib.com/cgi-bin/d1.cgi?l=es&base=es_wiki_10&page=showid&id=53629.
28. Ingenierosoftware. [En línea] [Citado el: 25 de 1 de 2012.]
<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.

Bibliografía

1. *ElastixBook-Comunicaciones_Unificadas_con_Elastix_Beta.pdf*[documento].
2. THE FREE DICTIONARY. [En línea] 1 de 11 de 2011. <http://es.thefreedictionary.com/llamada..>
3. **Pérez, Danae y Segura, Rainer.** *Servicio de Facturación para Elastix. Herramientas y Tecnologías.* Habana : s.n., 2011.
4. THE FREE DICTIONARY. [En línea] [Citado el: 1 de 11 de 2011.] <http://es.thefreedictionary.com/telefonista>.
5. Altatecnologia. [En línea] [Citado el: 2 de 11 de 2011.] http://www.altatecnologia.com.gt/plantas_telefonicas.html.
6. Comunidad de usuarios. [En línea] 3 de 11 de 2011. http://comunidad.asterisk-es.org/index.php?title=Documentos_sobre_Asterisk.
7. at4wireless. [En línea] [Citado el: 6 de 11 de 2011.] <http://www.at4wireless.com/es/ti-servicios-soluciones/automatizacion-telefonica-smartivr.html..>
8. *Descripción General del sistema SYSOPE 30001.pdf.* Habana : s.n.
9. definicionabc. [En línea] [Citado el: 17 de 11 de 2011.] <http://www.definicionabc.com/tecnologia/base-de-datos.php>.
10. Desarrolloweb. [En línea] [Citado el: 20 de 11 de 2011.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
11. Dirphp. [En línea] 27 de 11 de 2011. http://www.dirphp.com/dir/Software_y_servidores_PHP/Sitio_oficial_del_Servidor_de_Base_de_datos_MySQL_50.html.
12. Superhostin. [En línea] [Citado el: 27 de 11 de 2011.] <http://www.superhosting.cl/manuales/manual-tutorial-de-php.html>.
13. Php.ne. [En línea] [Citado el: 27 de 11 de 2011.] <http://php.net/manual/es/intro-what-is.php>.
14. Desarrolloweb. [En línea] [Citado el: 28 de 11 de 2012.] <http://www.desarrolloweb.com/articulos/que-es-html.html>.
15. Librosweb. [En línea] [Citado el: 28 de 11 de 2011.] <http://www.librosweb.es/javascript/index.html>.
16. Librosweb. [En línea] [Citado el: 28 de 11 de 2012.] <http://www.librosweb.es/css/index.html>.
17. Symfony. [En línea] [Citado el: 29 de 11 de 2011.] <http://www.symfony.es/que-es-symfony/>.

18. Freedownloadmanager. [En línea] [Citado el: 29 de 11 de 2011.]
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
19. Netbeans. [En línea] [Citado el: 29 de 11 de 2011.] http://netbeans.org/index_es.html.
20. Quanta-plus. [En línea] [Citado el: 29 de 11 de 2011.] <http://quanta-plus.uptodown.com/ubuntu>.
21. Recursostic. [En línea] [Citado el: 29 de 11 de 2011.]
<http://recursostic.educacion.es/observatorio/web/ca/software/servidores/580-elvira-mifsud>.
22. Soaagenda. [En línea] [Citado el: 29 de 11 de 2011.] [http://www.soaagenda.com/journal/articulos/que-es-bpm-que-es-bpms/..](http://www.soaagenda.com/journal/articulos/que-es-bpm-que-es-bpms/)
23. Programacionya.com. [En línea] [Citado el: 29 de 11 de 2011.] <http://programacionya.com/%C2%BFque-es-y-para-que-sirve-jquery/>.
24. Gbm. [En línea] [Citado el: 20 de 1 de 2012.]
http://www.gbm.net/bt/bt37/opinion/arquitecturas_empresariales.php.
25. solinfo. [En línea] [Citado el: 20 de 1 de 2012.] http://www.solinfo.cl/wp-content/themes/zine-style/archivos/cliente_servidor.html.
26. msdn.microsoft.com. [En línea] [Citado el: 23 de 1 de 2012.] <http://msdn.microsoft.com/es-es/library/bb972251.aspx>.
27. Diclib. [En línea] [Citado el: 23 de 1 de 2012.] http://www.diclib.com/cgi-bin/d1.cgi?l=es&base=es_wiki_10&page=showid&id=53629.
28. Ingenierosoftware. [En línea] [Citado el: 25 de 1 de 2012.]
<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
29. **Fernández Escribano, Gerardo.** *Introducción a Extreme Programming.* [Documento] .
30. Introducción a Extreme Programming. [En línea] [Citado el: 29 de 11 de 2012.]
<http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf...>
31. **Prieto, Félix.** *Patrones de Diseño.* [Documento] s.l. : Universidad de Valladolid, 2008. .
32. Gestión de Calidad y Pruebas de Software. [En línea] [Citado el: 30 de 11 de 2012.]
<http://pruebasdesoftware.com/pruebadeaceptacion.htm...>

Glosario de Términos:

Call Center: Conjunto de herramientas de Informática y de Telecomunicaciones que, puestas a disposición de un grupo de operadores encargados de atender llamadas telefónicas masivas, eleva la productividad de los recursos tecnológicos y de los recursos humanos.

VoIP: Voz sobre Protocolo de Internet, también llamado Voz sobre IP, VozIP, VoIP (por sus siglas en inglés), es un grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP (Internet Protocol).

GPL: General Public License. Es una licencia creada por la Free Software Fundación y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es Software Libre.

IVR: Son las siglas de Interactive Voice Response, que se traduce del inglés como Respuesta de Voz Interactiva.

WorkFlow: Flujo de trabajo.

ORM: Modelo de objetos relacional.