Universidad de las Ciencias Informáticas

Facultad 2

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Título:

Herramienta para el diseño y soporte online de un Sistema de Respuesta de Voz Interactiva (PLATEL-IVR)

Autores:

Yaidel Alfredo Carvajal Rondón.

Héctor Alexander Pérez Coello.

Tutor:

Ing. Yuliesky Bello Chávez.

Ciudad de La Habana, Mayo de 2009

"Año 50 aniversario del triunfo de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 2 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de ___

del año___.

Héctor Alexander Pérez Coello

_____

Autor

Yaidel Alfredo Carvajal Rondón

_____

Autor

Yuliesky Bello Chávez

_____

Tutor

Datos de Contacto del Tutor:

I. Datos generales

Nombre: Yuliesky Bello Chávez

Edad: 25 años

Sexo: Masculino

Especialidad: Ingeniero en Ciencias Informáticas.

Centro Universitario: Universidad de las Ciencias Informáticas

Municipio: Colón

Provincia: Matanzas

Carné de Identidad: 83073107709

Teléfonos: 8358140

E-mail: ybelloc@uci.cu

"No hay nada más poderoso que una idea cuando llega su momento"

Víctor Hugo

Agradecimientos

De Yaidel:

A mis padres Reinaldo, Alfredo y Estela por ser un ejemplo en todo momento.

A mi esposa por su incondicional apoyo.

A mi hermano por darme consejos bastante buenos a pesar de su edad.

A mis amigos por alentarme y acompañarme en todo momento.

A mi tutor Yuliesky por el apoyo y la confianza que depositó sobre nosotros.

A todos los que han puesto su grano de arena en general, muchas gracias.

De Héctor:

A mis padres Héctor y Concepción por ser un ejemplo en todo momento.

A mi hermana Lisandra por estar siempre a mi lado.

A todo el resto de mi familia y a los que ya no están con nosotros, que siempre están en mi corazón.

A Lisbet por ser sincera y por darme su amistad desinteresada.

A mis amigos por estar conmigo en las buenas y en las malas.

A mi tutor Yuliesky por el apoyo y la confianza que tuvo en nosotros.

A todos muchas gracias…….

Dedicatoria

De Yaidel:

A mis padres, a mi papá Reinaldo en particular por todo su apoyo siempre.

A mi esposa por pasarse horas de desvelo para que nuestro trabajo saliera sin problema alguno.

A mi familia en general que me ha apoyado de una forma u otra.

A mi abuelo, que aunque no se encuentra a mi lado en estos momentos, se que nunca me ha abandonado.

De Héctor:

A papi y a mami les dedico todo mi éxito.

A mi tía Maribel por que siempre confió en mí.

A mis tíos, primos y abuelos que forman parte de mí.

A todas las otras personas que de una manera u otra han estado vinculadas a mi vida.

A todos se los dedico……

Resumen

Resumen

En el presente trabajo de diploma se desarrolla una herramienta para la construcción de un sistema de Respuesta de Voz Interactiva (*IVR*[1]). Se exponen algunos conceptos generales relacionados con el tema del trabajo diploma en el primero de sus Capítulos. En función del logro de los objetivos se realiza un análisis de algunos de los sistemas IVR existentes a nivel internacional y nacional, luego se hace una selección detallada de las metodologías, herramientas y lenguaje de programación para el desarrollo de la herramienta.

Por último se muestran las características del sistema, dando paso a la realización de las fases que propone la metodología de desarrollo seleccionada, mostrando los artefactos generados en cada una de estas.

[1] *Del Inglés Interactive Voice Response*

Índice de contenidos

Índice de contenidos

Introducción 15

Fundamentación Teórica 17

1.1 Introducción 17

1.2 Estado del Arte 17

1.2.1 Sistemas Automatizados Existentes 17

Voice-Guide 18

Calixta-IVR 18

1.2.2 Situación en Cuba 19

1.3 Metodologías de Desarrollo de Software 19

Ingeniería de software 19

Razones de la Ingeniería de software 19

Metodología 19

1.3.1 Extreme Programming (XP) 20

Características de XP: 21

¿Qué es lo que propone XP? 21

1.3.2 Scrum (envoltorio para la gestión de prácticas de ingeniería en Extreme Programming) 22

1.4 El Servicio Web para el Desarrollo de Aplicaciones 23

1.5 Modelo Cliente – Servidor 24

1.6 Tecnología del lado del Servidor 25

1.6.1 Python 25

1.7 Servidor Web para la transferencia de datos cliente-servidor. Apache 26

1.8 Sistema de Gestión de Base de Datos. PostgreSQL 27

1.9 Herramientas utilizadas para el desarrollo de la Aplicación 28

1.9.1 Easy Eclipse 28

Índice de contenidos

1.9.2 Framework....28
1.9.2.1 Turbogears....29
1.9.2.2 Componentes de Turbogears utilizados para el desarrollo de la Herramienta....29
1.9.2.3 Dojo Toolkid....30
1.10 Tecnologías del lado del cliente....31
1.10.1 HTML....31
1.10.2 Java Script....31
1.10.3 AJAX....32
1.11 Conclusiones....32
Características del Sistema....33
2.1 Introducción....33
2.2 Objeto de Automatización....33
*2.*3 Propuesta del Sistema....33
2.4 Arquitectura del sistema propuesto....33
2.5 Funcionamiento de la herramienta sobre la arquitectura propuesta....33
2.6 Requisitos no funcionales del sistema....34
2.7 Conclusiones....35
Exploración y Planificación....36
3.1 Introducción....36
3.2 Fase de Exploración....36
3.2.1 Historias de Usuario....36
3.3 Planificación....41
3.3.1 Estimación de esfuerzo por Historias de Usuario....41
3.3.2 Plan de Iteraciones....42
Iteración 1....42

Índice de contenidos

Iteración 2 42

Iteración 3 42

Iteración 4 42

3.3.3 Plan de duración de las iteraciones 42

3.3.4 Plan de entregas 43

3.4 Conclusiones 44

Implementación y Pruebas 45

4.1 Introducción 45

4.2 Iteración 1 45

4.2.1 Tareas de las historias de usuario implementadas en la primera iteración 46

4.3 Iteración 2 49

4.3.1 Tareas de las historias de usuario implementadas en la segunda iteración 49

4.4 Iteración 3 51

4.4.1 Tareas de las historias de usuario implementadas en la tercera iteración 51

4.5 Iteración 4 53

4.5.1 Tareas de las historias de usuario implementadas en la cuarta iteración 53

4.6 Pruebas 55

4.7 Pruebas de Aceptación 56

4.8 Conclusiones 56

Conclusiones Generales 57

Recomendaciones 58

Referencias Bibliográficas 59

Bibliografía consultada 60

Anexo I 62

Anexo II 70

Opinión del usuario del Trabajo de Diploma 70

Índice de contenidos

Opinión del tutor del Trabajo de Dipoma 71

Glosario de Términos 78

Índice de Tablas y Figuras

Índice de Tablas y Figuras

Fig. 1 Modelo cliente-servidor.....24

Fig. 2 Arquitectura del sistema propuesto.....33

Tabla 3.2 Personas relacionadas con el sistema.....36

Tabla 3.2.1 Historia de usuario Contexto.....37

Tabla 3.2.2 Historia de usuario Extensión.....37

Tabla 3.2.3 Historia de usuario Prioridad.....38

Tabla 3.2.4 Historia de usuario Servicio.....38

Tabla 3.2.5 Historia de usuario Datos de Servicio.....39

Tabla 3.2.6 Historia de usuario Actualizar IVR.....39

Tabla 3.2.7 Historia de usuario Eliminar IVR.....40

Tabla 3.2.8 Historia de usuario Árbol de Acciones.....40

Tabla 3.3.1 Estimación de esfuerzo por historia de usuario.....41

Tabla 3.3.2 Plan de duración de las iteraciones.....43

Tabla 3.3.4 Plan de entregas.....43

Tabla 4.2.1 Historias de usuario implementadas en la primera iteración.....45

Tabla 4.2.2 Tarea #1 de la historia de usuario Contexto.....46

Tabla 4.2.3 Tarea #2 de la historia de usuario Contexto.....46

Tabla 4.2.4 Tarea #1 de la historia de usuario Extensión.....47

Tabla 4.2.5 Tarea #2 de la historia de usuario Extensión.....47

Tabla 4.2.6 Tarea #1 de la historia de usuario Prioridad.....48

Tabla 4.2.7 Tarea #2 de la historia de usuario Prioridad.....48

Tabla 4.3.1 Historias de usuario implementadas en la segunda iteración.....49

Tabla 4.3.2 Tarea #1 de la historia de usuario Servicio.....49

Tabla 4.3.3 Tarea #2 de la historia de usuario Servicio.....50

Tabla 4.3.4 Tarea #1 de la historia de usuario Datos del Servicio.....50

Índice de Tablas y Figuras

Tabla 4.4.1 Historias de usuario implementadas en la tercera iteración 51

Tabla 4.4.2 Tarea #2 de la historia de usuario Datos del Servicio 51

Tabla 4.4.3 Tarea #1 de la historia de usuario Actualizar IVR 52

Tabla 4.4.4 Tarea #2 de la historia de usuario Actualizar IVR 52

Tabla 4.5.1 Historias de usuario implementadas en la cuarta iteración 53

Tabla 4.5.2 Tarea #1 de la historia de usuario Eliminar IVR 53

Tabla 4.5.3 Tarea #2 de la historia de usuario Eliminar IVR 54

Tabla 4.5.4 Tarea #1 de la historia de usuario Árbol de acciones.....54

Tabla 4.5.5 Tarea #2 de la historia de usuario Árbol de acciones.....55

Tabla 4.5.1 Prueba de aceptación # 1 para la historia de usuario Extensión 62

Tabla 4.5.2 Prueba de aceptación # 2 para la historia de usuario Extensión 62

Tabla 4.5.3 Prueba de aceptación # 1 para la historia de usuario Prioridad 63

Tabla 4.5.4 Prueba de aceptación # 2 para la historia de usuario Prioridad 63

Tabla 4.5.5 Prueba de aceptación # 1 para la historia de usuario Servicio 64

Tabla 4.5.6 Prueba de aceptación # 2 para la historia de usuario Servicio 64

Tabla 4.5.7 Prueba de aceptación # 1 para la historia de usuario Datos del Servicio 65

Tabla 4.5.8 Prueba de aceptación # 2 para la historia de usuario Datos del Servicio 65

Tabla 4.5.9 Prueba de aceptación # 1 para la historia de usuario Actualizar IVR 66

Tabla 4.5.10 Prueba de aceptación # 2 para la historia de usuario Actualizar IVR 67

Tabla 4.5.11 Prueba de aceptación # 1 para la historia de usuario Eliminar IVR 67

Tabla 4.5.12 Prueba de aceptación # 2 para la historia de usuario Eliminar IVR 68

Tabla 4.5.13 Prueba de aceptación # 1 para la historia de usuario Árbol de acciones ..68

Tabla 4.5.14 Prueba de aceptación # 2 para la historia de usuario Árbol de acciones ..69

Fig. 3 Iniciar IVR.....72

Fig. 4 Validación Iniciar IVR 73

Fig. 5 Configurar IVR.....74

Índice de Tablas y Figuras

Fig. 6 Validación Configurar IVR 75

Fig. 7 Actualizar IVR.....76

Fig. 8 Validación Actualizar IVR 77

Introducción

Hoy la imagen de una empresa no solo se construye a través de una estrategia de marketing, sino además, con el apoyo de los canales con los que actúan sus clientes, entre ellos un *Call-Center* (Centro de Llamadas).

La importancia que ha alcanzado en los últimos años este tipo de plataformas, es un reflejo de la evolución que se experimenta en la forma de relacionarse una empresa con sus clientes. Su capacidad de influir en su lealtad y su impacto sobre las estrategias comerciales demuestran la importancia que tienen en la cadena de valor de las mismas. El desarrollo continuo de la Informática y las telecomunicaciones permiten crear Centros de Llamadas cada vez más robustos, con servicios que encaminen a las empresas a nuevas oportunidades de negocio.

En Cuba se han venido dando pasos de avance en este sentido, hoy la Empresa de Telecomunicaciones de Cuba (ETECSA) cuenta con Centros de Llamadas ubicados en varias zonas del país. Dichos Centros de Llamadas soportan sus servicios sobre plataformas propietarias, las cuales poseen un elevado costo en el mercado de las telecomunicaciones.

Comienza así el desarrollo de PLATEL; una solución nacional orientada a centros de contacto y PMGE (pequeñas, medianas y grandes empresas). Esta plataforma telefónica realiza un conjunto de tareas para brindar servicios telemáticos, garantizar el rendimiento, supervisión y administración de un centro de contacto. El núcleo de la plataforma es Asterisk, un software de Central Telefónica Privada (*PBX*[2]), basada en arquitectura PC (computadora personal) y completamente de código abierto (*OS*[3]).

Esta PBX tiene varios servicios entre los que se encuentra un sistema de Respuesta de Voz Interactiva (IVR), servicio que se configura mediante una consola, líneas de comandos y ficheros de configuración. Teniendo en cuenta que ya existen en la actualidad aplicaciones para humanizar el proceso de construcción de un sistema IVR pero que ninguna satisfacen las necesidades de la plataforma telefónica PLATEL se define como **problema científico** de este trabajo **:** ¿Cómo facilitar la adaptación de un sistema IVR a las especificidades de los clientes y garantizar el soporte online del mismo?

[2] Acrónimo de Private Branch Exchange
[3] Acrónimo de Open Source

Este problema se enmarca en el **objeto de estudio:** Servicios IVR existentes en una plataforma telefónica y en el **campo de acción:** Servicio IVR sobre la plataforma PLATEL.

El **objetivo general** que se persigue con la realización del presente trabajo es:

Desarrollar una herramienta capaz de construir un sistema IVR sobre la plataforma PLATEL, así como el soporte online del mismo.

Para dar cumplimiento al objetivo expuesto se proponen las siguientes **tareas de la investigación:**

- ✓ Investigar y elaborar el estado del arte de las herramientas para la construcción de sistemas IVR existentes hasta el momento.
- ✓ Definir las tecnologías y metodología a utilizar para la creación con calidad de la Herramienta.
- ✓ Desarrollar una aplicación Web amigable, con el objetivo de permitirle al usuario crear un IVR.

Fundamentación Teórica

1.1 Introducción

En el presente capítulo se abordan conceptos teóricos importantes que fueron necesarios investigar para la concepción del trabajo. Como parte fundamental de la investigación se describe el estudio del estado del arte realizado a los sistemas IVR existentes. También se explican las metodologías, herramientas y tecnologías utilizadas para el desarrollo de la propuesta.

1.2 Estado del Arte

¿Qué es un sistema IVR?

Un IVR es la típica máquina que responde con una voz grabada cuando se llama a una central telefónica. Consiste en un sistema que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz. Está orientado a entregar y/o capturar información automatizada a través del teléfono permitiendo el acceso a los servicios de información y operaciones autorizadas.

¿Por qué es necesario tener un IVR?

- ✓ Porque brinda 24 horas de atención ininterrumpida.
- ✓ Porque facilita y potencia la relación con los clientes.
- ✓ Porque automatiza las necesidades de cada sector y elimina el error humano.
- ✓ Porque permite lograr una organización interna más eficiente.
- ✓ Porque brinda funcionalidad en todas las áreas de trato al cliente.

¿Para qué tener un IVR?

- ✓ Para reducir costos y aumentar la productividad.
- ✓ Para disminuir los tiempos de atención.
- ✓ Para controlar y maximizar la actividad de las líneas telefónicas.
- ✓ Para mejorar la imagen de la empresa.

1.2.1 Sistemas Automatizados Existentes

Voice-Guide

Voice-Guide es un software que permite la creación fácil de sistemas de Contestación de Voz Interactiva, usando sistema operativo Windows (siendo utilizados estos sistemas en una amplia gama de áreas).

Características:

- ✓ El apoyo de la línea telefónica múltiple (a 240 líneas).
- ✓ Modelo señalado totalmente personalizable que empareja las definiciones que permite la compatibilidad con cualquier PBX.
- ✓ El buzón de voz totalmente destacado e ilimitado, con mensaje que se remite por el teléfono, email y Protocolo de Transferencia de Archivos (*FTP*[4]).
- ✓ Simple para usar Interfaces de Plan de Sistema Gráfica intuitiva.
- ✓ Fácil de instalar y configurar.

Calixta-IVR

Calixta-IVR es un sistema de respuesta interactiva de voz, que permite contestar cientos o miles de llamadas al día, dando información e interactuando con bases de datos vía telefónica. Cuenta con una interfaz de programación gráfica, que permite desarrollar complejos sistemas de atención telefónica mediante el arrastra y suelta (*d-d*[5]). Es compatible solamente con sistemas operativos Windows 2000, XP, 2003 y se conecta a las bases de datos mediante conectividad abierta a Base de Datos (*ODBC*[6]) y SQL.

Calixta-IVR es ideal para:

- ✓ Consulta de saldos en resumen y al detalle.
- ✓ Transacciones bancarias.
- ✓ Levantamientos de pedidos.
- ✓ Autorizaciones de crédito.

[4] Acrónimo de File Transfer Protocol
[5] Acrónimo de Drag and Drop
[6] Acrónimo de Open DataBase Connectivity

1.2.2 Situación en Cuba

Actualmente ETECSA cuenta con varias plantas telefónicas privadas entre las que se encuentran MITEL, ALCATEL y ERICSON, las cuales brindan el servicio IVR. Estas son adquiridas a un alto costo a nivel internacional y traen todos sus servicios definidos, privando a las empresas el poder configurar cualquiera de estos a sus necesidades.

Después de analizado el estado del tema a nivel internacional y la situación en Cuba, se concluye que los sistemas automatizados presentan como principal característica su desarrollo sobre plataforma propietaria, trayendo consigo las limitaciones que ello significa para desarrolladores de sistemas IVR de código abierto. Debido a esto surge la necesidad de realizar un estudio de las tendencias y tecnologías actuales que servirán para el desarrollo del presente trabajo, eligiendo para ello las más convenientes.

1.3 Metodologías de Desarrollo de Software

Ingeniería de software

"La Ingeniería de Software es una tecnología que indica "COMO" construir técnicamente un software: económico, fiable y que funcione eficientemente. Dentro de esta se encuentra la metodología que se va a utilizar para el desarrollo del trabajo." [1]

Razones de la Ingeniería de software

"Estudiar ¿cuáles son las actividades que organizadas, haciendo uso racional de los recursos y apoyándose en técnicas y herramientas, logran la mayor eficiencia al construir un software?" [1]

Metodología

"La rama de la metodología, dentro de la ingeniería de software, se encarga de elaborar estrategias de desarrollo de software que promuevan prácticas adaptables en vez de predecibles; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente". [2]

Para desarrollar un software es necesario seleccionar la metodología adecuada, teniendo en cuenta parámetros como los requerimientos, calidad, límite de tiempo, entre otros.

Existen varias metodologías de desarrollo de software entre las que se encuentran las pesadas o tradicionales y las ágiles o ligeras.

Las tradicionales se enfocan más en la definición de roles, incluyendo el modelado, siendo utilizadas en proyectos a largo plazo, con un numeroso equipo de desarrollo.

"Las metodologías ágiles están orientadas especialmente a proyectos pequeños y permiten que los programadores se concentren solamente en aquellas funciones que se necesitan inmediatamente. Se realizan entregas al cliente frecuentemente, de las cuales se obtiene retroalimentación constante, posibilitando respuestas rápidas a los cambios en el negocio". [3]

Debido a que se dispone de un corto período tiempo para el desarrollo del software, poca disponibilidad del personal y existe una buena comunicación entre los clientes y los programadores del sistema, donde la prioridad es satisfacer al mismo mediante tempranas y continuas entregas de software en iteraciones que no excedan los 30 días, se decidió utilizar la metodología ágil Programación Extrema (*XP*[7]) y como envoltorio para la gestión Scrum.

1.3.1 Extreme Programming (XP)

"La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Las cuatro variables de esta metodología son:

- ✓ **Coste:** Máquinas, especialistas y oficinas.
- ✓ **Tiempo:** Total y de Entregas.
- ✓ **Calidad:** Externa e Interna.
- ✓ **Alcance:** Intervención del cliente.

XP surgió 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, con ella se aumenta la productividad y su ciclo de vida ideal consiste de seis fases: Exploración, Planificación de la Entrega (*Release*), Implementación por Iteraciones, Pruebas, Mantenimiento y Muerte del Proyecto." [3]

[7] Acrónimo de Extreme Programming

Características de XP:

- ✓ "Pruebas Unitarias: son pruebas que se les realizan a los principales procesos para detectar las fallas que pudieran ocurrir en el futuro.
- ✓ Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- ✓ Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa." [3]

¿Qué es lo que propone XP?

- ✓ Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- ✓ El manejo del cambio se convierte en parte sustantiva del proceso.
- ✓ El costo del cambio no depende de la fase o etapa.
- ✓ No introduce funcionalidades antes que sean necesarias.
- ✓ El cliente o el usuario se convierte en miembro del equipo.

Derechos del Cliente

"El cliente tiene como derechos el de decidir que se implementa, saber el estado real y el progreso del proyecto, añadir, cambiar o quitar requerimientos en cualquier momento, así como obtener lo máximo de cada semana de trabajo y obtener un sistema funcionando cada 3 o 4 meses." [4]

Derechos del Desarrollador

"El desarrollador tiene como derechos el de decidir como se implementan los procesos, crear el sistema con la mejor calidad posible, pedir al cliente en cualquier momento aclaraciones de los requerimientos, estimar el esfuerzo para implementar el sistema y cambiar los requerimientos sobre la base a nuevos descubrimientos." [4]

1.3.2 Scrum (envoltorio para la gestión de prácticas de ingeniería en Extreme Programming)

"Tanto Scrum como la Programación Extrema (XP) requieren que los equipos completen algún tipo de producto potencialmente liberable al final de cada iteración. Estas iteraciones están diseñadas para ser cortas y de duración fija. Estos equipos aceptan que puede que se equivoquen por el camino, pero también son conscientes de que la mejor manera de encontrar dichos errores es dejar de pensar en el software a un nivel teórico de análisis y diseño y sumergirse en él, ensuciarse las manos y comenzar a construir el producto." [5]

Valores de Scrum:

- ✓ "Equipos auto-dirigidos y auto-organizados. No hay manager que decida, la excepción, es el Scrum Máster que debe ser 50% programador y que resuelve problemas, pero no manda. Los observadores externos; pueden observar, pero no interferir ni opinar.
- ✓ Una vez elegida una tarea, no se agrega trabajo extra. En caso que se agregue algo, se recomienda quitar alguna otra cosa.
- ✓ Encuentros diarios que impiden caer en el dilema: ¿Cómo es que un proyecto puede atrasarse un año?: Un día a la vez.
- ✓ Iteraciones de treinta días; se admite que sean más frecuentes.
- ✓ Demostración a participantes externos al fin de cada iteración.
- ✓ Al principio de cada iteración, existe un planeamiento adaptable guiado por el cliente. " [5]

Beneficios de XP y Scrum:

- ✓ "La agilidad de gestión y mecanismos de control de Scrum son aplicables para cualquier tipo de proyecto, incluyendo las iniciativas empresariales que consisten en múltiples, simultáneos de desarrollo de software, desarrollo empresarial, re-ingeniería, comercialización, apoyo y ejecución de proyectos.
- ✓ La realización de los proyectos con los beneficios de auto-organización; equipos de iteración (o Sprint) dirigida a objetivo, en lugar de dirigirse historia.

- ✓ En la Programación Extrema, cuando los proyectos son envueltos por Scrum, pasan a ser escalables y se pueden ejecutar simultáneamente.
- ✓ Con XP y Scrum los proyectos pueden beneficiarse de los negocios de valor ADM (métrica para medir el retorno de la inversión y la gestión de la iniciativa)". [5]

1.4 El Servicio Web para el Desarrollo de Aplicaciones

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. La facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. En vez de crear clientes para Windows, Mac OS, GNU/Linux, y otros sistemas operativos, la aplicación es escrita una vez y es mostrada casi en todos lados.

Ventajas:

- ✓ Desarrollo barato, sencillo y rápido.
- ✓ Datos centralizados y fácil integración de datos de múltiples fuentes.
- ✓ Permiten el desarrollo de comunidades que dan valor a las aplicaciones (software social).
- ✓ Una empresa puede migrar de sistema operativo o cambiar el Hardware libremente sin afectar el funcionamiento de las aplicaciones de servidor.
- ✓ Actualizar o hacer cambios en el Software es sencillo y sin riesgos de incompatibilidades. Existe solo una versión en el servidor lo que implica que no hay que distribuirla entre los demás computadores. El proceso es rápido y limpio.
- ✓ Se facilita el trabajo a distancia. Se puede trabajar desde cualquier PC o computador portátil con conexión a Internet o a una red interna o privada.

"Todas estas ventajas dejan claro el potencial de las aplicaciones Web. La utilización de ésta tecnología conlleva a reducir costos y complicaciones, y proporciona mayor libertad a la hora de realizar cualquier tipo de cambios." [6]

1.5 Modelo Cliente – Servidor

La arquitectura Cliente/Servidor es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones y hacer más fácil el desarrollo y mejorar su mantenimiento, cuenta con tres elementos fundamentales; cliente, servidor y red de comunicación.

La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas. Esto se aplica tanto a clientes como servidores, la forma más estándar de aplicación y uso de estos sistemas es mediante la explotación de las PC's a través de interfaces gráficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales. De esta forma un servidor da servicio a múltiples clientes de forma concurrente y los cambios realizados en las plataformas de los clientes o de los servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.

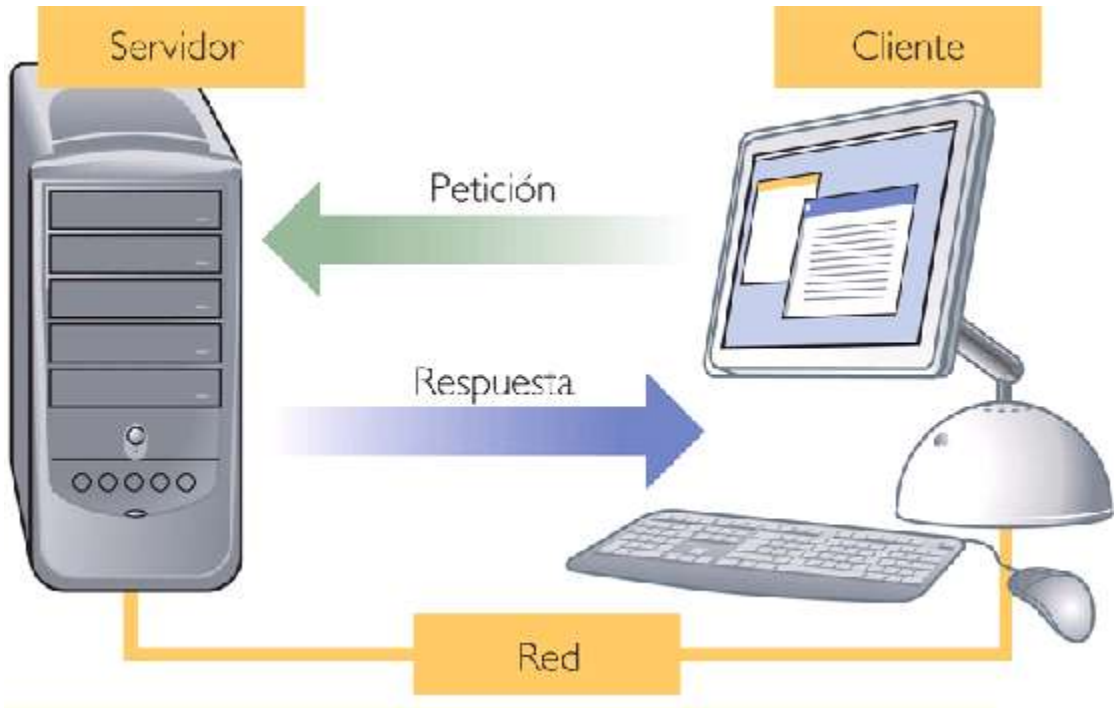

Fig. 1 Modelo cliente-servidor.

1.6 Tecnología del lado del Servidor

Estas tecnologías pueden estar incrustadas o no en el código HTML pues ellas no dependen del navegador, ya que son ejecutadas en el servidor, o sea que se debe tener en cuenta que el servidor con que se trabaje cuente con soporte para la tecnología seleccionada.

La *AGI*[8], interfaz estándar externa por la que los programas pueden controlar el plan de discado (Dialplan) de la PBX Asterisk. La PBX Asterisk brinda una interfaz estándar para los scripts de la AGI, los cuales pueden ser escritos en casi cualquier lenguaje de programación moderno. Dentro de los lenguajes más comúnmente utilizados y que se destacan en la programación AGI está el Python como lenguaje seleccionado para desarrollar la propuesta.

1.6.1 Python

Tiene una sintaxis muy visual, gracias a una notación identada (con márgenes) de obligado cumplimiento. Es un lenguaje fácil de aprender y potente. " La elegante sintaxis de Python, su gestión de tipos dinámica y su naturaleza interpretada hacen de él el lenguaje ideal para guiones y desarrollo rápido de aplicaciones en muchas áreas y en la mayoría de las plataformas; desde aplicaciones Windows a servidores de red o incluso, páginas Web y también cuenta con eficaces estructuras de datos de alto nivel." [7]

Es Interactivo, dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudar a entender mejor el lenguaje y probar los resultados de la ejecución.

Su código no necesita ser compilado, por lo que se dice que el código es interpretado. Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores a que adopten por un estilo de programación particular.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- ✓ La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.

[8] Acrónimo de Asterisk Gateway Interface

- ✓ La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- ✓ La cantidad de plataformas en las que se puede desarrollar, como Unix, Windows, OS/2, Mac y otros.
- ✓ Además, Python es gratuito, incluso para propósitos empresariales.

Ventaja:

- ✓ **Desarrollo más rápido:** Puedes escribir un programa, salvarlo y ejecutarlo. En un lenguaje compilado tienes que pasar por los pasos de compilar y ligar el software, lo cual puede ser un proceso lento.

Desventaja:

- ✓ **Lentitud:** Los programas interpretados son más lentos que los compilados. Sin embargo los programas interpretados suelen ser cortos (como los scripts), en los que la diferencia es inapreciable.

1.7 Servidor Web para la transferencia de datos cliente-servidor. Apache

Apache es un servidor de red, de código abierto, se desarrolla dentro del proyecto HTTP Server (The Apache Software Foundation) y sale en 1995. “Consistía inicialmente en un conjunto de parches que se le aplicaban al servidor NCSA (National Center for Supercomputing Aplications). Posteriormente se convirtió en un servidor muy eficiente para la configuración de mensajes de error ejecutando un script en dicho caso e incluye una base de datos para la autenticación de forma sencilla y negociado de contenido. Apache tiene amplia aceptación en la red: en el 2005, fue el servidor HTTP más usado, empleado en el 48% de los sitios Web en el mundo.” [8]

Esta disponible para diferentes plataformas como:

- ✓ GNU/Linux
- ✓ Mac OS y Mac OSX Server
- ✓ Netware
- ✓ UNIX comerciales como AIX, Digital UNIX, HP-US, IRIX, SCO, Solaris, SunOs, UnixWare

- ✓ Windows

El servidor Apache es capaz de funcionar sobre casi todas las plataformas existentes confiándole así gran independencia. Se decidió utilizar Apache por tener una serie de características funcionales que a diferencia de otros servidores Web permite crear aplicaciones con un mayor rendimiento y estabilidad, por ser un producto distribuido como software libre y por tener una amplia documentación de apoyo en la Universidad.

1.8 Sistema de Gestión de Base de Datos. PostgreSQL

"PostgreSQL es un servidor de base de datos objeto relacional libre. Como muchos otros proyectos Open Source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). Incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional." [9]

Ventajas:

- ✓ Puede ser utilizado en los principales sistemas operativos: Linux, Unix, Mac OS, Windows.
- ✓ Soporte nativo para los lenguajes más populares: PHP, C, C++, Perl, Python, etc.
- ✓ Soporte de protocolo de comunicación encriptado por Protocolo de Capa de Conexión Segura (*SSL*[9]).
- ✓ Soporta distintos tipos de datos, además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
- ✓ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- ✓ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.

[9] Acrónimo de Secure Sockets Layer

- ✓ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- ✓ Puede ser utilizado, modificado y distribuido por cualquiera gratuitamente, para cualquier propósito ya sea con fines privados, comerciales o académicos.

Por todas las razones antes expuestas se decidió utilizar PostgreSQL como gestor de bases de datos en el desarrollo de la aplicación, aunque la escalabilidad de la misma permite el soporte de otros gestores de base datos.

1.9 Herramientas utilizadas para el desarrollo de la Aplicación

1.9.1 Easy Eclipse

El entorno de desarrollo integrado o IDE de Eclipse emplea módulos o plug-in para proporcionar toda su funcionalidad, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Esta característica le permite a Eclipse extenderse usando lenguajes de programación como Ruby, PHP y Python entre otros.

Easy Eclipse dispone de varias distribuciones para desarrollo entre ellas una optimizada para el desarrollo en Python (Pydev).

Las principales fortalezas de esta herramienta son:

- ✓ Es Open Source y multiplataforma.
- ✓ Posee integración con Sistemas Controladores de Versiones o SVN.
- ✓ Posee un completamiento de código muy bueno el cual contribuye al desarrollo ágil de la aplicación
- ✓ Posee una extensa documentación y soporte en Internet.
- ✓ El consumo es aceptable lo cual no interfiere con el desarrollo.

1.9.2 Framework

El concepto framework viene aparejado con el desarrollo de software. Es una estructura de soporte robusta y bien definida en la cual otro proyecto de software puede ser

organizado y desarrollado. Los framework se han convertido en la piedra angular de la moderna ingeniería del software ya que están compuestos por componentes personalizables e intercambiables para el desarrollo de una aplicación, en otras palabras, es una aplicación genérica incompleta y/o configurable a la que se puede añadir las últimas piezas para construir sistema concreto.

Un framework Web, por tanto, se puede definir como un conjunto de clases y componentes que forman un diseño reutilizable al cual se le pueden incluir módulos para facilitar y agilizar el desarrollo de aplicaciones Web. La mayoría de los framework Web incorporan en su arquitectura MVC (Modelo Vista-Controlador), separando los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres capas distintas, donde la vista es la página (*HTML*[10]) Lenguaje de Marcas de Hipertexto y el código que provee de datos dinámicos a la página; el modelo es el SGBD (Sistema de Gestión de Base de Datos) y la lógica del negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista.

1.9.2.1 Turbogears

TurboGears constituye un framework diseñado para desarrollar aplicaciones Web 2.0 con Python, está compuesto por una serie de componentes que le permiten al desarrollador hacer aplicaciones Web de forma rápida. Todos estos componentes están empaquetados por separado, y pueden ser descargados y utilizado sin Turbogears. Posee amplia documentación, es multiplataforma, cuenta con una numerosa comunidad de desarrolladores y usuarios.

1.9.2.2 Componentes de Turbogears utilizados para el desarrollo de la Herramienta

MochiKit: Una biblioteca Java Script que incluye componentes tales como logueado (logging: del ingles) y otros efectos visuales que son comunes en muchas bibliotecas. Permite añadir comportamiento asíncrono (*AJAX*[11]) a una aplicación Web, carga y manipulación de conjuntos de datos (*JSON*[12]*)* y contiene un conjunto de funciones para facilitar la creación dinámica de componentes, llamada Mochikid.DOM.

Sus principales características son:

[10] Acrónimo de Hypertext Markup Language
[11] Acrónimo de Asynchronous JavaScript And XML
[12] Acrónimo de JavaScript Object Notation

- ✓ Gestión de tareas asíncronas.
- ✓ Funciones de manipulación de objetos y comparaciones.
- ✓ Capa de acceso al modelo de objeto (*DOM*[13]).
- ✓ Iteraciones y enumeradores en Java Script.
- ✓ Efectos visuales y funciones de color.
- ✓ Funciones Python

Kid: Una plantilla de lenguaje que le permite incluir Python incrustado en una página (*XHTML*[14]) lenguaje extensible de marcado de hipertexto, explícitamente con una etiqueta, o utilizar un conjunto de atributos que incluyen estructuras condicionales y bucles para definir el contenido dinámico. Soporta herencia de plantillas y emparejado (*XSLT*[15]) forma de transformar documentos XML en otros e incluso a formatos.

CherryPy: Un framework orientado a objetos que le permite desarrollar aplicaciones Web en Python. Incluye mecanismos para hacer simples operaciones centradas en la Web tales como mapeo de Localizador de Recurso Uniforme (*URL*[16]), solicitadas a métodos Python usando decoradores, control del servidor y solicitudes de filtrado.

SQLAlchemy: Un moderno Mapeador Objeto Relacional o ORM, que proporciona un potente y flexible sistema de gestión de la conexión entre la memoria y los objetos de Python, almacenamiento de datos relacional que proporciona la persistencia de los objetos. Simplifica el acceso a una base de datos por parte del programador convirtiendo toda sentencia SQL a operaciones con objetos. Así por ejemplo añadir una fila a una tabla es tan fácil como crear un objeto.

1.9.2.3 Dojo Toolkid

Un framework que contiene Interfaz de Programación de Aplicaciones (*APIs*[17]) y widgets o controles para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a

[13] Acrónimo de Document Object Model
[14] Acrónimo de Extensible Hypertext Markup Language
[15] Acrónimo de Extensible Stylesheet Language/Transform
[16] Acrónimo de Uniform Resource Locator
[17] Acrónimo de Application Programming Interface

ellas, y la habilidad de degradar cuando AJAX/Java Script no es completamente soportado en el cliente.

1.10 Tecnologías del lado del cliente

Son las que están insertadas en la página HTML del cliente, siendo interpretadas y ejecutadas por el navegador, por una parte, se encargan de gestionar la comunicación con el servidor, de solicitar un servicio concreto y de recibir los datos enviados por éste y por otra, es la tecnología que presenta al usuario los datos en pantalla y que le ofrece los comandos necesarios para utilizar las presentaciones que ofrece el servidor. Su correcta funcionalidad depende del soporte de la versión del navegador a ser utilizado por el usuario.

1.10.1 HTML

Lenguaje de Marcas de Hipertextos (*HTML*[18]), que utiliza una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto. Está diseñado para estructurar textos y presentarlos en forma de hipertextos (sistema en el que los archivos de texto, voz, imágenes y video interactúan con los lectores), formato estándar en el que se presentan las páginas web. Los documentos HTML pueden ser creados por cualquier editor que permita texto sin formato.

1.10.2 Java Script

Es una tecnología que está diseñada para manejar la apariencia de la ventana en el navegador y manipular los eventos, para lo cual usa un conjunto de objetos.

Ventajas:

- ✓ **Fácil de aprender, rápido y potente:** permite realizar ciertas funciones rápidas en una página Web solo con crear el código y cargarlo sin necesidad de crear una máquina virtual para compilar el mismo. Es un lenguaje de alto nivel, siendo capaz de aprovechar las propiedades de los exploradores Web e incluso puede realizar algunas acciones sobre el sistema en que se está ejecutando.
- ✓ **Usabilidad:** Es uno de los lenguajes que más se utiliza en la Web, donde están publicadas millones de páginas que lo usan.

[18] Acrónimo de Hyper Text Markup Language

- ✓ **Reducción de la carga del servidor:** Al contar con la habilidad de ejecutarse en el cliente y tener tantas funcionalidades, se ha podido ganar la atención de la mayoría de los desarrolladores Web pues ayuda a reducir la carga de trabajo del servidor, ejemplo de esto son las validaciones que posibilita.
- ✓ Con Java Script se evita el proceso de mandar información de un lado a otro haciendo uso excesivo de la red ya que él valida antes de enviarse la información al servidor.

Desventajas:

Uno de los inconvenientes que tiene este lenguaje es que tiene que estar activado en los navegadores para que su uso sea posible por estos. Por otro lado no le proporciona al programador control total de la página Web. En ocasiones los desarrolladores deben realizar diferentes implementaciones de código Java Script para sus aplicaciones, debido a que no todos los navegadores interpretan de la misma manera el código.

1.10.3 AJAX

"No es una tecnología en sí misma; es una técnica de desarrollo Web para crear aplicaciones interactivas. En realidad, se trata en la unión de varias tecnologías independientes. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad, usabilidad en las aplicaciones y mejor respuesta a las acciones del usuario." [10] Es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores.

1.11 Conclusiones

En este capítulo se hizo un análisis de las metodologías, las herramientas y tecnologías elegidas para darle solución al sistema, se ha llegado a la conclusión que es importante realizar una aplicación Web teniendo en cuenta la facilidad de acceso a la misma desde distintas partes. Es de vital importancia el dominio de estas herramientas seleccionadas y analizar la mejor forma de aplicarlas, para desarrollar un sistema de máxima calidad que cumpla con los requisitos propuestos y de al cliente una versión del producto que satisfaga sus intereses.

Características del Sistema

2.1 Introducción

En este capítulo se realiza un análisis de las características del sistema a desarrollar, teniendo como aspecto principal la arquitectura propuesta para la solución, brindando además el objeto de automatización, el prototipo de interfaz de usuario y los requerimientos no funcionales del sistema.

2.2 Objeto de Automatización

Proceso de configuración y actualización de los archivos de Asterisk para la creación de un IVR.

2.3 Propuesta del Sistema

Se ha decidido darle solución al problema mediante una aplicación Web que facilite la realización de las tareas antes mencionadas en el menor tiempo posible.

2.4 Arquitectura del sistema propuesto

Fig. 2 Arquitectura del sistema propuesto.

2.5 Funcionamiento de la herramienta sobre la arquitectura propuesta

Esta aplicación es la encargada de automatizar el trabajo de las entidades relacionadas con el proceso de edición de archivos de configuración de Asterisk relacionados con la creación de un IVR, donde el usuario al iniciar el sistema entra a la interfaz **Iniciar_IVR**, mediante la cual introduce la extensión y selecciona un archivo de sonido para dar bienvenida al sistema. Luego accede a **Configurar_IVR**, guiado por una leyenda de servicios básicos, seleccionando el camino a seguir de la llamada entrante al sistema. Toda esta información se almacena en el servidor de Base de Datos, utilizándola el

servidor Asterisk para la puesta en funcionamiento del sistema IVR. Por último el usuario interactúa con **Actualizar_IVR**, permitiendo seleccionar uno ya configurado y realizar actualizaciones en el campo extensión, así como eliminarlo. (Ver Anexo II)

2.6 Requisitos no funcionales del sistema

Requerimientos de apariencia o interfaz externa

La aplicación propuesta será usada por personas con conocimientos básicos de informática y de Asterisk, la interfaz debe ser amigable y fácil de usar, de manera que no sea una dificultad para los usuarios el trabajo con la misma. La comunicación entre las máquinas clientes y el servidor Web será por HTTP y entre el servidor de aplicaciones, la Base de Datos y otra máquina con un Asterisk mediante TCP/IP.

Requerimientos de usabilidad

A los administradores del sistema se les dará un adiestramiento básico en el uso de la aplicación. Estas personas tendrán un nivel de acceso amplio en la aplicación para poder darle respuesta a la especificidad que desee el cliente para el sistema IVR.

Requerimientos de rendimiento

La aplicación propuesta utilizará AJAX, permitiendo quela misma sea rápida y el tiempo de respuesta no exceda los 2 segundos, aparejado a la rapidez con que el usuario requiere la respuesta a su petición.

Requerimiento de portabilidad

La herramienta podrá ser usada bajo cualquier distribución de Linux. El servidor Web, el servidor de Base de Datos y el servidor Asterisk pueden estar en la misma PC conformando una arquitectura centralizada o por otra parte una arquitectura distribuida sería cada servidor en una máquina aparte sin ocasionar problema alguno para el funcionamiento del sistema en ninguno de los dos casos.

Requerimientos de seguridad

Confiabilidad: La información manejada por el sistema debe estar protegida de acceso no autorizado.

Integridad: La información manejada por el sistema debe ser objeto de cuidadosa protección contra la corrupción y estados de inconsistencia.

Disponibilidad: La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información y los mecanismos utilizados para lograr la seguridad no deben ser un obstáculo a los usuarios para obtener los datos deseados en un momento dado.

Requerimientos de software

En las computadoras de los usuarios solo se requiere el navegador Web Mozilla Firefox versión 3.0 o superior, con la opción de JavaScript activada.

Requerimientos de hardware

Requerimientos mínimos para los servidores:

- Computador Pentium 4 a 3.0 GHz.
- 512 MB de RAM o superior.
- 40 de espacio libre en disco duro.

Requerimientos mínimos para la conexión del cliente:

- Computador Pentium 3 a 1.6 GHz.
- 128 MB de RAM.

2.7 Conclusiones

En el presente capítulo se realizó un estudio detallado de las características del sistema a implementar. Obteniéndose como resultado una propuesta de arquitectura respondiendo al objeto de automatización existente, así como una descripción de las distintas interfaces que componen el mismo y los requerimientos no funcionales que debe tener la aplicación.

Exploración y Planificación

3.1 Introducción

En el presente capítulo se abordan las fases de exploración y planificación pertenecientes a la metodología de desarrollo XP, fases en la que se potencian las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Además se exponen los artefactos generados durante el transcurso de dichas fases.

3.2 Fase de Exploración

"La metodología de desarrollo XP comienza con la fase de exploración. Durante esta se realiza el proceso de identificación de las historias de usuario, así como la familiarización de los equipos de trabajo con las tecnologías y herramientas seleccionadas para la construcción del proyecto." [11]

Personas relacionadas con el sistema

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios de los procesos que se ejecutan en el mismo. Además de aquellas que se encuentran involucradas en los mismos, pues participan en ellos pero no obtienen ningún resultado de valor.

Tabla 3.2 Personas relacionadas con el sistema

Personas Relacionadas con el Sistema	Justificación
Administrador	Es la persona con privilegios para realizar el diseño del IVR. Es el encargado de crear el IVR así como de realizar su actualización.

3.2.1 Historias de Usuario

"Las historias de usuario son la forma en que se especifican en XP los requisitos funcionales del sistema. Estas se escriben desde la perspectiva del cliente aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. El contenido de estas debe ser concreto y sencillo." [11] Durante la fase de exploración se identificaron ocho historias de usuario, las cuales se detallan a continuación:

Tabla 3.2.1 Historia de usuario Contexto

Historia de Usuario	
Número: 1	Usuario: Administrador
Nombre historia: Contexto	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Héctor A. Pérez Coello-Yaidel A. Carvajal Rondón	
Descripción: Al modelar la tabla correspondiente a la configuración del sistema IVR en la Base de Datos, en el campo contexto se le pone el mismo nombre de contexto configurado para dicha tabla en el fichero de Asterisk (extensions.conf).	

Tabla 3.2.2 Historia de usuario Extensión

Historia de Usuario	
Número: 2	Usuario: Administrador
Nombre historia: Extensión	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Héctor A. Pérez Coello-Yaidel A. Carvajal Rondón	
Descripción: El Administrador inserta el número de extensión en el componente visual de la interfaz *Configurar_IVR*, que no es más que la extensión a la que se puede llamar para acceder al sistema.	

Tabla 3.2.3 Historia de usuario Prioridad

Historia de Usuario	
Número: 3	Usuario: Administrador
Nombre historia: Prioridad	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Héctor A. Pérez Coello-Yaidel A. Carvajal Rondón	
Descripción: El Administrador inserta el número de prioridad en el componente visual de la interfaz *Configurar_IVR*, no es más que el orden de ejecución del servicio seleccionado.	

Tabla 3.2.4 Historia de usuario Servicio

Historia de Usuario	
Número: 4	Usuario: Administrador
Nombre historia: Servicio	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Héctor A. Pérez Coello-Yaidel A. Carvajal Rondón	
Descripción: El Administrador selecciona en el componente visual de la interfaz *Configurar_IVR* el servicio que desea ejecute el sistema en la prioridad especificada.	

Tabla 3.2.5 Historia de usuario Datos de Servicio

Historia de Usuario	
Número: 5	Usuario: Administrador
Nombre historia: Datos de Servicio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 2 y 3
Programador responsable: Héctor A. Pérez Coello-Yaidel A. Carvajal Rondón	
Descripción: El Administrador especifica los datos del servicio seleccionado en la interfaz *Configurar_IVR*.	

Tabla 3.2.6 Historia de usuario Actualizar IVR

Historia de Usuario	
Número: 6	Usuario: Administrador
Nombre historia: Actualizar IVR	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Héctor A. Pérez Coello-Yaidel A. Carvajal Rondón	
Descripción: El administrador selecciona en la interfaz *Actualizar_IVR* el IVR que desea actualizar en la tabla de la base de datos.	

Tabla 3.2.7 Historia de usuario Eliminar IVR

Historia de Usuario	
Número: 7	Usuario: Administrador
Nombre historia: Eliminar IVR	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Héctor A. Pérez Coello-Yaidel A. Carvajal Rondón	
Descripción: El administrador selecciona en la interfaz *Actualizar_IVR* el IVR a eliminar.	

Tabla 3.2.8 Historia de usuario Árbol de Acciones

Historia de Usuario	
Número: 8	Usuario: Administrador
Nombre historia: Árbol de Acciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 4
Programador responsable: Héctor A. Pérez Coello-Yaidel A. Carvajal Rondón	
Descripción: Se muestra en la interfaz visual la secuencia grafica de funcionalidades seleccionadas.	

3.3 Planificación

"Durante esta fase se realiza una estimación del esfuerzo que costará implementar cada historia de usuario. Este se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin interrupción alguna. Esta estimación incluye todo el esfuerzo asociado a la implementación de la historia de usuario, por ejemplo: las pruebas unitarias, la integración y refactorización del código y la preparación y ejecución de las pruebas de aceptación." [11]

3.3.1 Estimación de esfuerzo por Historias de Usuario

Para el desarrollo de la aplicación propuesta en este trabajo se realiza una estimación del esfuerzo para cada una de las historias de usuario identificadas, llegándose a los siguientes resultados.

Tabla 3.3.1 Estimación de esfuerzo por historia de usuario

Historias de Usuario	Puntos estimados
Contexto	1
Extensión	1
Prioridad	1
Servicio	2
Datos de Servicio	2
Actualizar IVR	2
Eliminar IVR	1
Árbol de Acciones	2

3.3.2 Plan de Iteraciones

Una vez identificadas las historias de usuario del sistema y estimado el esfuerzo dedicado a la realización de cada una de estas se procede a la planificación de la etapa de implementación de la aplicación. Se decidió realizar la implementación de la aplicación en cuatro iteraciones, detalladas a continuación:

Iteración 1

Esta iteración tiene como objetivo la implementación de las historias de usuario 1,2 y 3 por su menor complejidad y prioridad en el negocio. Durante la misma se comienza el trabajo conforme a la arquitectura propuesta para el sistema, finalizando con una primera versión, la cual será discutida con el cliente para un mejor entendimiento y retroalimentación para el grupo de trabajo.

Iteración 2

El objetivo de esta iteración es la implementación de las historias de usuario 4 y la primera parte de la 5, historias que tienen un mayor grado de complejidad.

Iteración 3

En esta iteración se dará fin a la implementación de la historia 5 y la 6, obteniendo la versión 0.3 del sistema.

Iteración 4

Esta última iteración plantea la implementación de la historias de usuario 7 y 8. Al finalizar la misma se constará de la versión 1.0 del producto final listo para su puesta en funcionamiento durante un periodo determinado para evaluar su desempeño.

3.3.3 Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto utilizando la metodología XP se crea el plan de duración de cada una de las iteraciones, según los equipos de desarrollo con que se cuente, en este caso se hace para el único equipo de desarrollo que se tiene. Este plan se encarga de mostrar las historias de usuario que serán cumplimentadas en cada una de las iteraciones, así como la duración estimada de de cada una y el orden en que se implementarán.

Tabla 3.3.2 Plan de duración de las iteraciones

Iteración	Orden de las historias de usuario a implementar	Duración total de la iteración
Iteración 1	1.1- Contexto 1.2- Extensión 1.3- Prioridad	3 semanas.
Iteración 2	2.1- Servicio 2.2- Datos de Servicio(1ª parte)	3 semanas
Iteración 3	3.1- Datos de Servicio(2da parte) 3.2- Actualizar IVR	3 semanas.
Iteración 4	4.1- Eliminar IVR 4.2- Árbol de Acciones	3 semanas.

3.3.4 Plan de entregas

Plan de entregas aproximadas ideado para la fase de implementación. Al finalizar la cuarta iteración se obtendrá la primera versión del producto final.

Tabla 3.3.4 Plan de entregas

Herramienta	Final 1ra Iteración 1ra semana de Febrero.	Final 2da Iteración 3ra semana de Febrero.	Final 3ra Iteración última semana de Marzo.	Final 4ta Iteración 3ra semana de Abril.
Herramienta para el diseño y soporte online de un IVR	0.1	0.2	0.3	1.0

3.4 Conclusiones

En este capítulo se abordó todo lo referente a las fases de exploración y planificación del proyecto, haciendo una descripción de cada uno de los artefactos generados en las mismas durante su desarrollo, abordándose también la distribución de las iteraciones.

Implementación y Pruebas

4.1 Introducción

"XP plantea que la implementación de un software se hace de forma iterativa, obteniendo al final de cada iteración un producto funcional que debe ser probado y mostrado al cliente para que los desarrolladores trabajen conociendo las opiniones dadas sobre el mismo. En el presente capítulo se detallan las dos iteraciones llevadas a cabo durante la etapa de construcción del sistema, exponiendo las tareas generadas por cada historia de usuario, así como las pruebas de aceptación efectuadas sobre el proyecto." [11]

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para cada una de estas. Al inicio de las mismas, se lleva a cabo una revisión del plan de iteraciones y se modifica de ser necesario.

4.2 Iteración 1

Durante esta iteración se abordan las historias de usuario elegidas y se construye la base de la arquitectura del sistema con el fin de obtener un producto con las funcionalidades primarias para ser mostrado al cliente y obtener una rápida y amplia retroalimentación de este.

Tabla 4.2.1 Historias de usuario implementadas en la primera iteración

Historias de Usuario	Estimación	Real
Contexto	1	1
Extensión	1	1
Prioridad	1	1

4.2.1 Tareas de las historias de usuario implementadas en la primera iteración

Tabla 4.2.2 Tarea #1 de la historia de usuario Contexto

Tarea	
Número tarea: 1	Número historia: 1
Nombre tarea: Investigar característica de tabla para configuración de IVR.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 12 Enero 2009	Fecha fin: 15 Enero 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello	
Descripción: Se realiza un análisis de la posible configuración de la tabla correspondiente al sistema IVR en la Base de Datos que corresponda a las características de un DialPlan en Asterisk.	

Tabla 4.2.3 Tarea #2 de la historia de usuario Contexto

Tarea	
Número tarea: 2	Número historia: 1
Nombre tarea: Modelación de la tabla en la Base de Datos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 16 Enero 2009	Fecha fin: 19 Enero 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello	
Descripción: Se modela la tabla con las características correspondientes.	

Tabla 4.2.4 Tarea #1 de la historia de usuario Extensión

Tarea	
Número tarea: 1	Número historia: 2
Nombre tarea: Implementación de la funcionalidad Extensión.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 20 Enero 2009	Fecha fin: 22 Enero 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello	
Descripción: Se implementa la funcionalidad para guardar la extensión en la tabla modelada.	

Tabla 4.2.5 Tarea #2 de la historia de usuario Extensión

Tarea	
Número tarea: 2	Número historia: 2
Nombre tarea: Configuración en la interfaz de la funcionalidad Extensión	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 23 Enero 2009	Fecha fin: 26 Enero 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello	
Descripción: Se configura en la interfaz la funcionalidad que permite introducir un número de extensión para el sistema IVR.	

Tabla 4.2.6 Tarea #1 de la historia de usuario Prioridad

Tarea	
Número tarea: 1	Número historia: 3
Nombre tarea: Implementación de la funcionalidad Prioridad.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 27 Enero 2009	Fecha fin: 29 Enero 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se implementa la funcionalidad para guardar la prioridad en la tabla modelada.	

Tabla 4.2.7 Tarea #2 de la historia de usuario Prioridad

Tarea	
Número tarea: 2	Número historia: 3
Nombre tarea: Configuración en la interfaz de la funcionalidad Prioridad	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 30 Enero 2009	Fecha fin: 2 Febrero 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se configura en la interfaz la funcionalidad que permite darle prioridad al servicio a seleccionar para el sistema IVR.	

4.3 Iteración 2

En el transcurso de esta iteración se implementan las historias de usuario 4 y la primera parte de la 5 como historias de mayor complejidad.

Tabla 4.3.1 Historias de usuario implementadas en la segunda iteración

Historias de Usuario	Estimación	Real
Servicio	2	2
Datos del Servicio(1ra parte)	1	1.5

4.3.1 Tareas de las historias de usuario implementadas en la segunda iteración

Tabla 4.3.2 Tarea #1 de la historia de usuario Servicio

Tarea	
Número tarea: 1	Número historia: 4
Nombre tarea: Implementación de la funcionalidad Servicio.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 3 Febrero 2009	Fecha fin: 9 Febrero 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se implementa la funcionalidad para guardar el servicio seleccionado en la tabla modelada.	

Tabla 4.3.3 Tarea #2 de la historia de usuario Servicio

Tarea	
Número tarea: 2	Número historia: 4
Nombre tarea: Configuración en la interfaz de la funcionalidad Servicio	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 10 Febrero 2009	Fecha fin: 16 Febrero 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se configura en la interfaz la funcionalidad que permite seleccionar el servicio para el sistema IVR.	

Tabla 4.3.4 Tarea #1 de la historia de usuario Datos del Servicio

Tarea	
Número tarea: 1	Número historia: 5
Nombre tarea: Implementación de la funcionalidad Datos del Servicio.	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha inicio: 17 Febrero 2009	Fecha fin: 26 Febrero 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se implementa la funcionalidad para guardar los datos del servicio seleccionado en la tabla modelada.	

4.4 Iteración 3

En esta iteración se culmina con la implementación de la historia 5 y 6, formalizando así la versión 0.3 del producto.

Tabla 4.4.1 Historias de usuario implementadas en la tercera iteración

Historias de Usuario	Estimación	Real
Servicio(2da parte)	1	1.5
Actualizar IVR	2	2

4.4.1 Tareas de las historias de usuario implementadas en la tercera iteración

Tabla 4.4.2 Tarea #2 de la historia de usuario Datos del Servicio

Tarea	
Número tarea: 2	Número historia: 5
Nombre tarea: Configuración en la interfaz de la funcionalidad Datos del Servicio.	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha inicio: 27 Febrero 2009	Fecha fin: 8 Marzo 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se configura en la interfaz la funcionalidad que permite facilitar los datos del servicio para el sistema IVR.	

Tabla 4.4.3 Tarea #1 de la historia de usuario Actualizar IVR

Tarea	
Número tarea: 1	Número historia: 6
Nombre tarea: Implementación de la funcionalidad Actualizar IVR.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 9 Marzo 2009	Fecha fin: 16 Marzo 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se implementa la funcionalidad para guardar los datos actualizados en la tabla de la base de datos.	

Tabla 4.4.4 Tarea #2 de la historia de usuario Actualizar IVR

Tarea	
Número tarea: 2	Número historia: 6
Nombre tarea: Configuración de la interfaz de la funcionalidad Actualizar IVR.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17 Marzo 2009	Fecha fin: 24 Marzo 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se configura en la interfaz la funcionalidad que permite actualizar la extensión del sistema IVR.	

4.5 Iteración 4

En esta iteración se culmina con la fase de implementación, contando con la versión final 1.0 del producto listo para su puesta en funcionamiento.

Tabla 4.5.1 Historias de usuario implementadas en la cuarta iteración

Historias de Usuario	Estimación	Real
Eliminar IVR	1	1
Árbol de Acciones	2	3

4.5.1 Tareas de las historias de usuario implementadas en la cuarta iteración

Tabla 4.5.2 Tarea #1 de la historia de usuario Eliminar IVR

Tarea	
Número tarea: 1	Número historia: 7
Nombre tarea: Implementación de la funcionalidad Eliminar IVR.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 25 Marzo 2009	Fecha fin: 27 Marzo 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se implementa la funcionalidad para eliminar el IVR seleccionado.	

Tabla 4.5.3 Tarea #2 de la historia de usuario Eliminar IVR

Tarea	
Número tarea: 2	Número historia: 7
Nombre tarea: Configuración de la funcionalidad Eliminar IVR en la interfaz Actualizar_IVR.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 28 Marzo 2009	Fecha fin: 1 Abril 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se configura en la interfaz visual la funcionalidad para eliminar un IVR.	

Tabla 4.5.4 Tarea #1 de la historia de usuario Árbol de acciones

Tarea	
Número tarea: 1	Número historia: 8
Nombre tarea: Implementación de la funcionalidad Árbol de acciones.	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha inicio: 2 Abril 2009	Fecha fin: 12 Abril 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se implementa la funcionalidad para modelar gráficamente el servicio seleccionado.	

Tabla 4.5.5 Tarea #2 de la historia de usuario Árbol de acciones

Tarea	
Número tarea: 2	Número historia: 8
Nombre tarea: Configuración en la interfaz del Árbol de acciones.	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha inicio: 13 Abril 2009	Fecha fin: 23 Abril 2009
Programador responsable: Yaidel A. Carvajal Rondón – Héctor A. Pérez Coello.	
Descripción: Se configura en la interfaz visual el Árbol de acciones.	

4.6 Pruebas

"Las pruebas dan la oportunidad de saber si lo implementado es lo que en realidad se tenía en mente. Ellas indican si el trabajo funciona, cuando no se pueda pensar en ninguna prueba que pudiese originar un fallo en el sistema, entonces se habrá acabado por completo.

Uno de los pilares fundamentales de XP es el proceso de pruebas. Mediante este método se reduce el número de errores no detectados así como el tiempo entre la introducción de este en el sistema y su detección. Todo esto tiende a elevar la calidad del producto desarrollado y la seguridad de los programadores a la hora de introducir cambios o modificaciones.

XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores y encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida además de comprobar que dicha funcionalidad sea la esperada por el cliente". [12]

La metodología XP propone las pruebas unitarias y las de aceptación. Durante el desarrollo de la herramienta en primer lugar se le realizaron pruebas unitarias al código

a medida que se iba desarrollando el mismo y finalmente se le aplicó al producto las pruebas de aceptación.

4.7 Pruebas de Aceptación

*"*Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario. Durante las iteraciones, las historias de usuario seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable." [12] Las pruebas de aceptación realizadas al sistema y las interfaces mostrando el resultado de estas pueden verse en el Anexo I y Anexo II respectivamente.

4.8 Conclusiones

En el presente capítulo se trataron las etapas de implementación y pruebas del software en desarrollo. Para ello se mostraron todos los artefactos generados, realizando una descripción de cada uno de ellos, dando paso a las pruebas de aceptación, con las cuales se probaron los requisitos trazados.

Conclusiones Generales

A partir del desarrollo de la Herramienta para el diseño y soporte online de un Sistema de Respuesta de Voz Interactiva (PLATEL-IVR), se puede arribar a las siguientes conclusiones:

- ✓ Se cumplió el objetivo general trazado.
- ✓ Se realizó un análisis y selección de las metodologías, herramientas y tecnologías necesarias para la creación con calidad del producto.
- ✓ Se obtuvo una aplicación que automatiza los procesos de configuración de un IVR en PLATEL.

Los resultados de este trabajo y la utilización de la herramienta desarrollada facilitarán el diseño de un IVR atendiendo a las especificidades del cliente en un Call Center.

Recomendaciones

Al concluir el desarrollo de este documento se recomienda:

- ✓ Continuar desarrollando la investigación a fin de perfeccionar e incrementar las funcionalidades de la herramienta para en próximas versiones diseñar IVR avanzados.
- ✓ Estudiar e implementar una AGI en Python capaz de interactuar con cualquier diseño de IVR con el objetivo de realizar su configuración.
- ✓ Continuar con la investigación de nuevas tecnologías informáticas para garantizar mejoras en futuras versiones de la Herramienta para el diseño y soporte online de un Sistema de Respuesta de Voz Interactiva (PLATEL-IVR).

Referencias Bibliográficas

[1] Jacobson, Ivar y Booch, Grady y Rumbaugh, James. *El proceso unificado de software*. Primera edición. Pearson Educación, S.A. 2000.

[2] García Rubio, Félix Óscar. S/A. *Metodologías de desarrollo de software. S/A*. 2002.

[3] Fernández Escribano, Gerardo, *Introducción a Extreme Programming, Ingeniería de Software II*, 2002.

[4] Ferrer, G. R. Y. J. *Programación Extrema y Software Libre*. 2002.

[5] Schwaber K., Beedle M., Martin R.C. *Agile Software Development with SCRUM*. Prentice Hall. 2001.

[6] *Ventajas e Inconvenientes de las aplicaciones Web*. 2007. Página web. Disponible en: http://www.avidos.net/blogold/aplicaciones-web/. Consultado 5-3-2009.

[7] Allen Downey, Jeffrey Elkner, Chris Meyers. *Aprenda a pensar como un programador con Python* .Primera Edición, Estados Unidos de América: 2002.

[8] The Apache Software Foundation. *Documentación del Servidor HTTP Apache 2.0*. 2007. Página web. Disponible en: http://httpd.apache.org/docs/2.0/es/. Consultado 20-3-2009.

[9] *PostgreSQL. Tutorial de PostgreSQL*. 2007. Página web. Disponible en: http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html. Consultado 20-3-2009.

[10] *AJAX un nuevo acercamiento a aplicaciones Web*. Página web. Disponible en: http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php. Consultado 10-3-2009.

[11] Beck, K, and Wesley Addison. *Extreme Programming Explained*. 2000.

[12] Crispin, L. y House, T. (2002). *Probando la Programación Extrema*. Addison Wesley. Título original: *Testing Extreme Programming*.

Bibliografía

Bibliografía consultada

1. The Apache Software Foundation. Documentación del Servidor HTTP Apache 2.0. 2007. Página web. Disponible en: http://httpd.apache.org/docs/2.0/es/. Consultado 20-3-2009.

2. PostgreSQL. Tutorial de PostgreSQL. 2007. Página web. Disponible en: http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html. Consultado 20-3-2009.

3. Silva, Moises. Instalación y Configuración de Asterisk. 2002.

4. Amitelo Wireless Asterisk. Realtime Administration (Awara). 2001

5. Ventajas e Inconvenientes de las aplicaciones Web. 2007. Página web. Disponible en: http://www.avidos.net/blogold/aplicaciones-web/. Consultado 5-3-2009.

6. César F. Acebal, Juan M. Cueva Lovelle*. Extreme Programming (XP): un nuevo método de desarrollo de software*. Depto. de Informática, Área de Lenguajes y Sistemas Informáticos, Universidad de Oviedo. NOVATICA, Marzo-Abril 2002, pp 8-12.

7. Estrada Terence, Jimmy. Asterisk en español. 2005.

8. *AJAX un nuevo acercamiento a aplicaciones Web*. Página web. Disponible en: http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php. Consultado 10-3-2009.

9. Van Meggelen, Jim. Madsen, Leif. Smith, Jared. "*Asterisk The Future of Telephony". Versión 2,* Estados Unidos de América: O'Reilly Media, 2007.

10. Flavio E. Goncalves. 2007. *Asterisk PBX. Guía de configuración*. 1ra Edición. Brasil, pp. 362.

11. Gomillon, David. Dempster, Barrie. Build Telephony System with Asterisk, Primera Edición, Estados Unidos de Norte América USA: Editorial Pack Publishing LTD, 2005,290p, ISBN-1-904811-15-9.

12. Sitio oficial de Asterisk PBX. Disponible en: http://www.asterisk.org. Consultado 10-12-2008.

13. Sitio oficial de Python. Disponible en: http://www.python.org/. Consultado 20-2-2009.

14. Sitio oficial de Voice-Guide. Disponible en: http:// www.VoiceGuide.com. Consultado 10-2-2009.

15. Sitio oficial de Calixta-IVR. Disponible en: http:// www.Calixta.com. Consultado10-2-2009.

Anexo I

Pruebas de Aceptación

Tabla 4.5.1 Prueba de aceptación # 1 para la historia de usuario Extensión

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de usuario: Extensión.
Nombre: Introducir número de extensión en tabla de la base de datos.	
Descripción: Probar que se escriba dato en tabla.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, enviando el dato introducido para escribir en la tabla de configuración de IVR de la base de datos.	
Entrada / Pasos de ejecución: Se introduce dato válido.	
Resultado esperado: El dato es reconocido y se escribe en campo correspondiente de la tabla.	
Evaluación de la prueba: Satisfactoria.	

Tabla 4.5.2 Prueba de aceptación # 2 para la historia de usuario Extensión

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de usuario: Extensión.
Nombre: Introducir número de extensión no válido en tabla de la base de datos.	
Descripción: Probar que no se escriba dato en tabla.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, el dato no válido introducido (número de más de 4 dígitos, letras, componente visual en blanco o extensión existente en tabla) no es correcto para escribir en la tabla de configuración de IVR de la base de datos.	
Entrada / Pasos de ejecución: Se introduce dato no válido.	

Resultado esperado: El dato no es reconocido y no se escribe en campo correspondiente de la tabla, emitiendo un mensaje de error dando la posibilidad de intentarlo de nuevo.
Evaluación de la prueba: Satisfactoria.

Tabla 4.5.3 Prueba de aceptación # 1 para la historia de usuario Prioridad

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de usuario: Prioridad.
Nombre: Introducir prioridad en tabla de la base de datos.	
Descripción: Probar que se escriba dato en tabla.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, enviando el dato introducido para escribir en la tabla de configuración de IVR de la base de datos.	
Entrada / Pasos de ejecución: Se introduce dato válido.	
Resultado esperado: El dato es reconocido y se escribe en campo correspondiente de la tabla.	
Evaluación de la prueba: Satisfactoria.	

Tabla 4.5.4 Prueba de aceptación # 2 para la historia de usuario Prioridad

Caso de Prueba de Aceptación	
Código: HU3_P2	Historia de usuario: Prioridad.
Nombre: Introducir prioridad no válida en tabla de la base de datos.	
Descripción: Probar que no se escriba dato en tabla.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, el dato no válido introducido (letras, componente en blanco o prioridad existente en contexto en uso) no es	

correcto para escribir en la tabla de configuración de IVR de la base de datos.
Entrada / Pasos de ejecución: Se introduce dato no válido.
Resultado esperado: El dato no es reconocido y no se escribe en campo correspondiente de la tabla, emitiendo un mensaje de error dando la posibilidad de intentarlo de nuevo.
Evaluación de la prueba: Satisfactoria.

Tabla 4.5.5 Prueba de aceptación # 1 para la historia de usuario Servicio

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de usuario: Servicio.
Nombre: Introducir el servicio en tabla de la base de datos.	
Descripción: Probar que se escriba dato en tabla.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, enviando el dato seleccionado para escribir en la tabla de configuración de IVR de la base de datos.	
Entrada / Pasos de ejecución: Se selecciona dato válido.	
Resultado esperado: El dato es reconocido y se escribe en campo correspondiente de la tabla.	
Evaluación de la prueba: Satisfactoria.	

Tabla 4.5.6 Prueba de aceptación # 2 para la historia de usuario Servicio

Caso de Prueba de Aceptación	
Código: HU4_P2	Historia de usuario: Servicio.
Nombre: Introducir servicio no válido en tabla de la base de datos.	
Descripción: Probar que no se escriba dato en tabla.	

Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, sin dato seleccionado (componente visual en blanco) para escribir en la tabla de configuración de IVR de la base de datos.
Entrada / Pasos de ejecución: No selecciona dato alguno.
Resultado esperado: El dato no es reconocido y no se escribe en campo correspondiente de la tabla, mensaje de error y posibilidad de intento nuevamente.
Evaluación de la prueba: Satisfactoria.

Tabla 4.5.7 Prueba de aceptación # 1 para la historia de usuario Datos del Servicio

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de usuario: Datos del Servicio.
Nombre: Introducir los datos del servicio seleccionado en tabla de la base de datos.	
Descripción: Probar que se escriba dato en tabla.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, enviando el dato para escribir en la tabla de configuración de IVR de la base de datos.	
Entrada / Pasos de ejecución: Se selecciona o se introduce dato válido.	
Resultado esperado: El dato es reconocido y se escribe en campo correspondiente de la tabla.	
Evaluación de la prueba: Satisfactoria.	

Tabla 4.5.8 Prueba de aceptación # 2 para la historia de usuario Datos del Servicio

Caso de Prueba de Aceptación	
Código: HU5_P2	Historia de usuario: Datos del Servicio.
Nombre: Introducir datos del servicio no válido en tabla de la base de datos.	

Descripción: Probar que no se escriba dato en tabla.
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, sin dato seleccionado (componente visual en blanco) para escribir en la tabla de configuración de IVR de la base de datos.
Entrada / Pasos de ejecución: No selecciona o se introduce dato alguno.
Resultado esperado: El dato no es reconocido y no se escribe en campo correspondiente de la tabla, se emite mensaje de error y se da la posibilidad de intentarlo de nuevo.
Evaluación de la prueba: Satisfactoria.

Tabla 4.5.9 Prueba de aceptación # 1 para la historia de usuario Actualizar IVR

Caso de prueba de Aceptación	
Código: HU6_P1	Historia de usuario: Actualizar IVR.
Nombre: Introducir los datos para actualizar IVR en tabla de la base de datos.	
Descripción: Probar que se escriba dato en tabla.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, enviando el dato para escribir en la tabla de configuración de IVR de la base de datos.	
Entrada / Pasos de ejecución: Se selecciona o se introduce dato válido.	
Resultado esperado: El dato es reconocido y se escribe en campo correspondiente de la tabla.	
Evaluación de la prueba: Satisfactoria.	

Tabla 4.5.10 Prueba de aceptación # 2 para la historia de usuario Actualizar IVR

Caso de Prueba de Aceptación	
Código: HU6_P2	Historia de usuario: Actualizar IVR.
Nombre: Introducir datos no válidos para actualizar IVR en tabla de la base de datos.	
Descripción: Probar que no se escriba dato en tabla.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, sin dato seleccionado (componente visual en blanco) para escribir en la tabla de configuración de IVR de la base de datos.	
Entrada / Pasos de ejecución: No selecciona o se introduce dato alguno.	
Resultado esperado: El dato no es reconocido y no se escribe en campo correspondiente de la tabla, se emite mensaje de error y se da la posibilidad de intentarlo de nuevo.	
Evaluación de la prueba: Satisfactoria.	

Tabla 4.5.11 Prueba de aceptación # 1 para la historia de usuario Eliminar IVR

Caso de prueba de Aceptación	
Código: HU7_P1	Historia de usuario: Eliminar IVR.
Nombre: Seleccionar el IVR a eliminar en tabla de la base de datos.	
Descripción: Probar que se elimine IVR en tabla.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, enviando el dato seleccionado para eliminar IVR en la tabla de configuración de IVR de la base de datos.	
Entrada / Pasos de ejecución: Se selecciona IVR existente en la base de datos.	
Resultado esperado: El dato es reconocido y se elimina IVR de la tabla.	

Evaluación de la prueba: Satisfactoria.

Tabla 4.5.12 Prueba de aceptación # 2 para la historia de usuario Eliminar IVR

Caso de Prueba de Aceptación	
Código: HU7_P2	Historia de usuario: Eliminar IVR.
Nombre: Seleccionar dato no válido para eliminar IVR en tabla de la base de datos.	
Descripción: Probar que no se elimine IVR en tabla.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, sin dato seleccionado (componente visual en blanco) para eliminar IVR en la tabla de configuración de IVR de la base de datos.	
Entrada / Pasos de ejecución: No selecciona o se introduce dato alguno.	
Resultado esperado: El dato no es reconocido y no se elimina IVR correspondiente de la tabla, se emite mensaje de error y se da la posibilidad de intentarlo de nuevo.	
Evaluación de la prueba: Satisfactoria.	

Tabla 4.5.13 Prueba de aceptación # 1 para la historia de usuario Árbol de acciones

Caso de Prueba de Aceptación	
Código: HU8_P1	Historia de usuario: Árbol de acciones.
Nombre: Escribir en fichero de configuración del Árbol.	
Descripción: Probar que se escriba dato en fichero.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, escribiendo dato correspondiente a servicio seleccionado en fichero de configuración del Árbol.	
Entrada / Pasos de ejecución: Se escribe dato en fichero.	
Resultado esperado: El dato se escribe en fichero de configuración del Árbol.	

Evaluación de la prueba: Satisfactoria.

Tabla 4.5.14 Prueba de aceptación # 2 para la historia de usuario Árbol de acciones

Caso de Prueba de Aceptación	
Código: HU8_P2	Historia de usuario: Árbol de acciones.
Nombre: Mostrar Árbol de acciones en interfaz visual.	
Descripción: Probar que se muestre servicio seleccionado en interfaz visual.	
Condiciones de ejecución: La aplicación debe ser ejecutada normalmente, mostrando el servicio seleccionado en interfaz visual.	
Entrada / Pasos de ejecución: Se muestra servicio en interfaz visual.	
Resultado esperado: El servicio es mostrado en interfaz visual.	
Evaluación de la prueba: Satisfactoria.	

Anexo II

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “Herramienta para el diseño y soporte online de un Sistema de Respuesta de Voz Interactiva (PLATEL-IVR) “, fue realizado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, correspondiendo con los objetivos trazados, el trabajo realizado le satisface

□ Totalmente
□ Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

_____
_____
_____
_____
_____
_____
_____
_____
_____

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

_____ _____

Representante de la entidad Cargo

_____ _____

Firma Cuño

Anexo II

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: “Herramienta para el diseño y soporte online de un Sistema de Respuesta de Voz Interactiva (PLATEL-IVR) “

Autor: Héctor Alexander Pérez Coello

Yaidel Alfredo Carbajal Rondón

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

_____
_____
_____
_____
_____
_____
_____
_____
_____

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros Informáticos; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

_____ Firma

_____ Fecha

Anexo II

Interfaz de usuario Iniciar_IVR

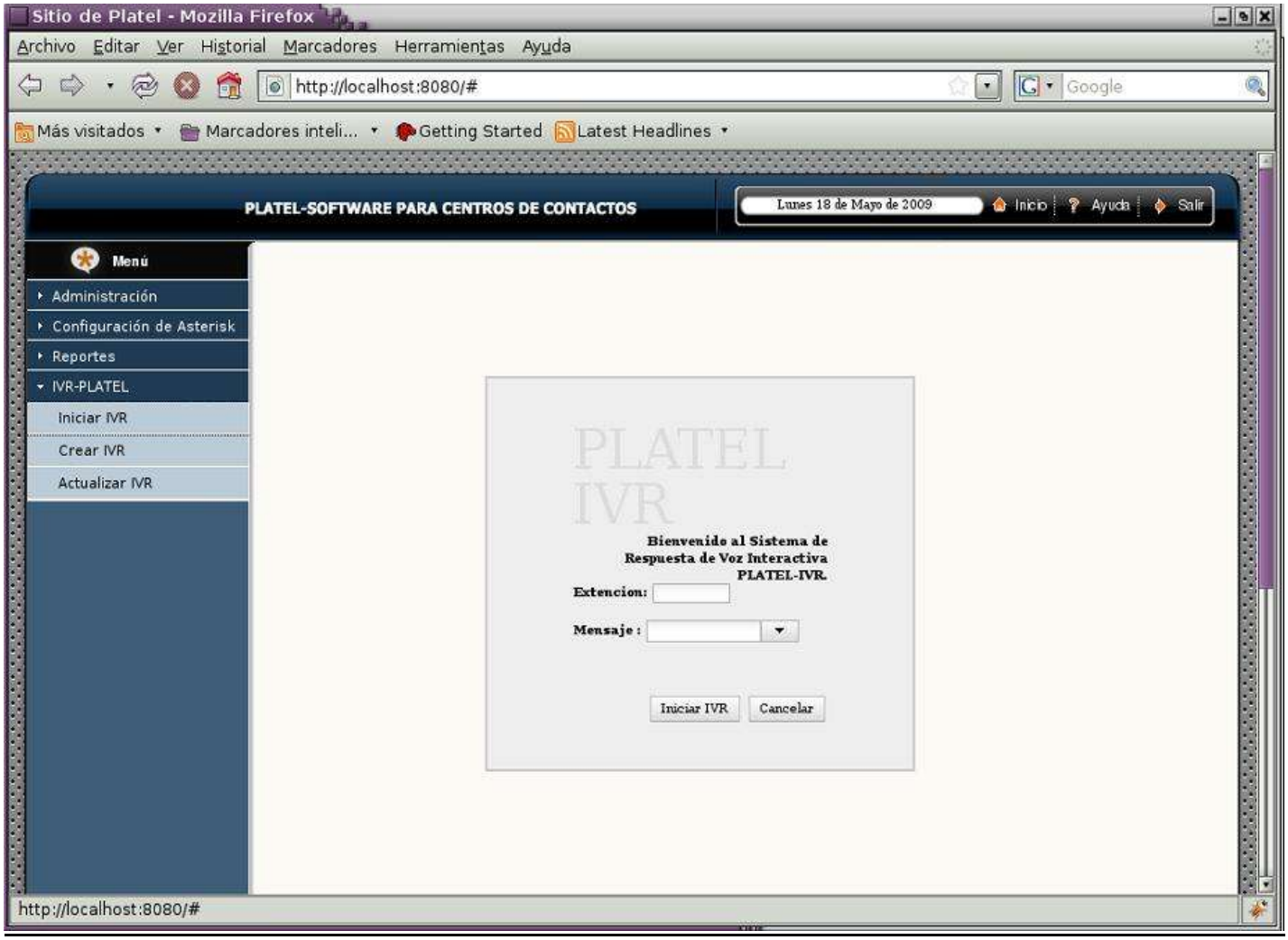

Fig. 3 Iniciar IVR

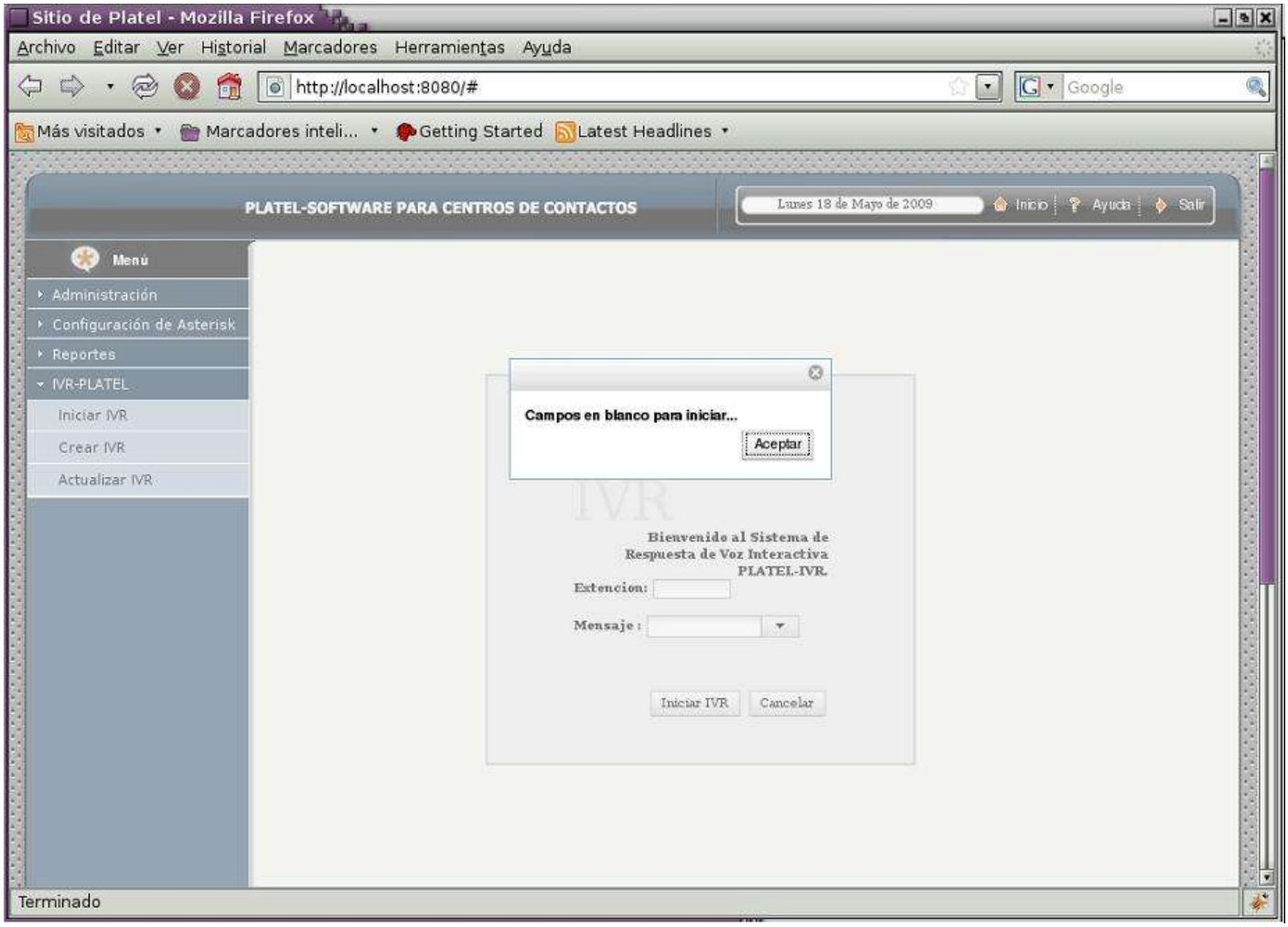

Fig. 4 Validación Iniciar IVR

Anexo II

Interfaz de usuario Configurar_IVR

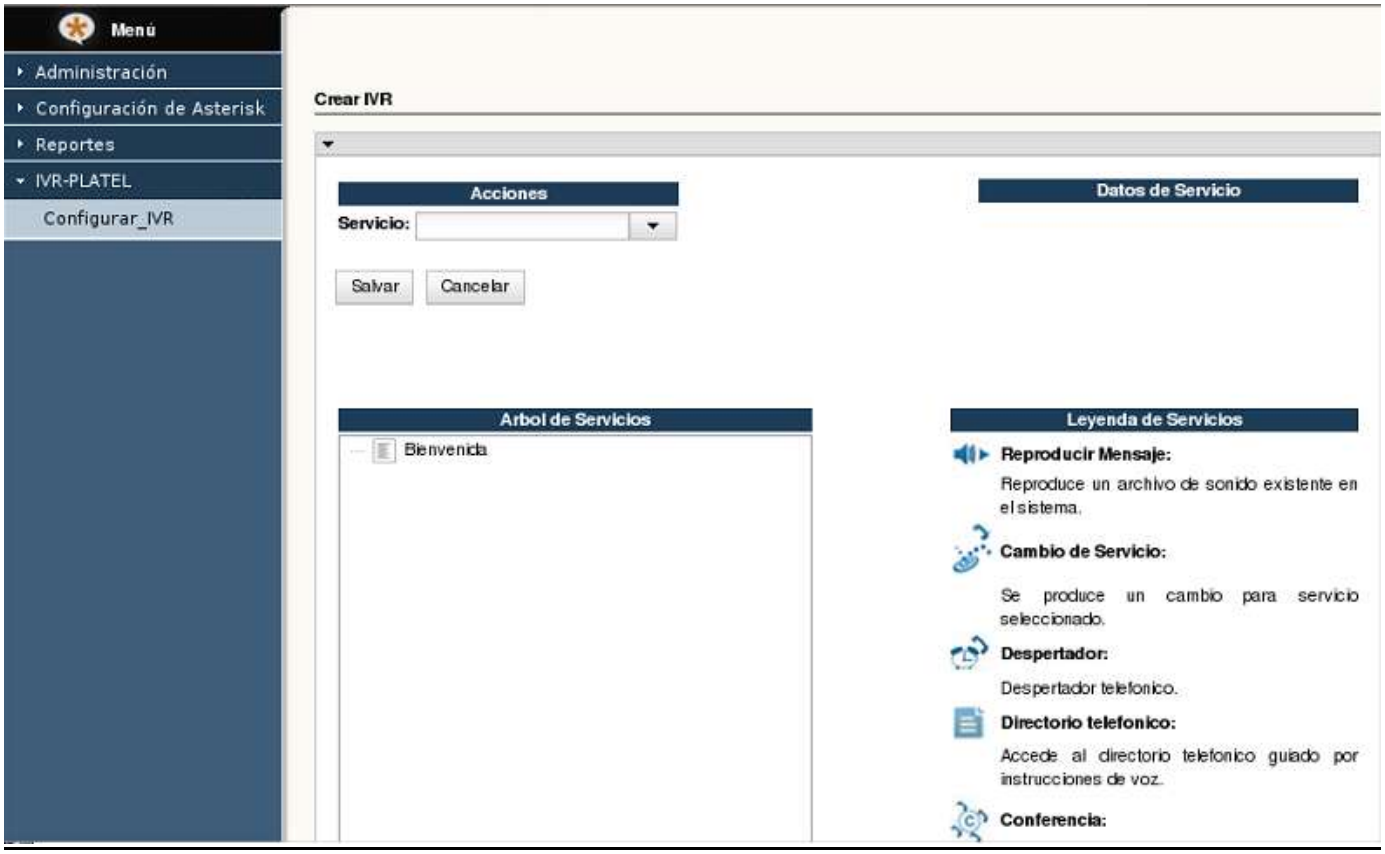

Fig. 5 Configurar IVR

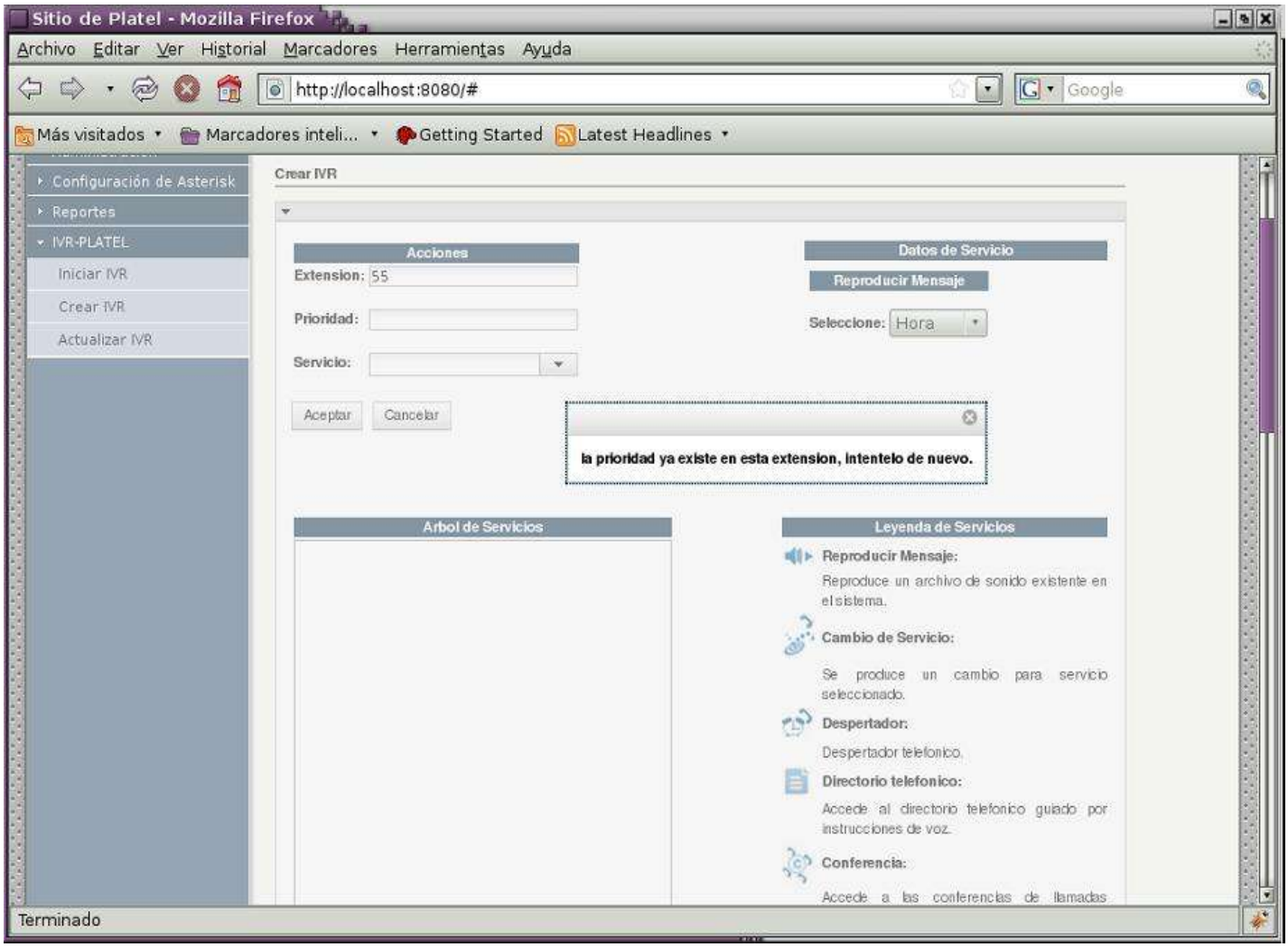

Fig. 6 Validación Configurar IVR

Anexo II

Interfaz de usuario Actualizar_IVR

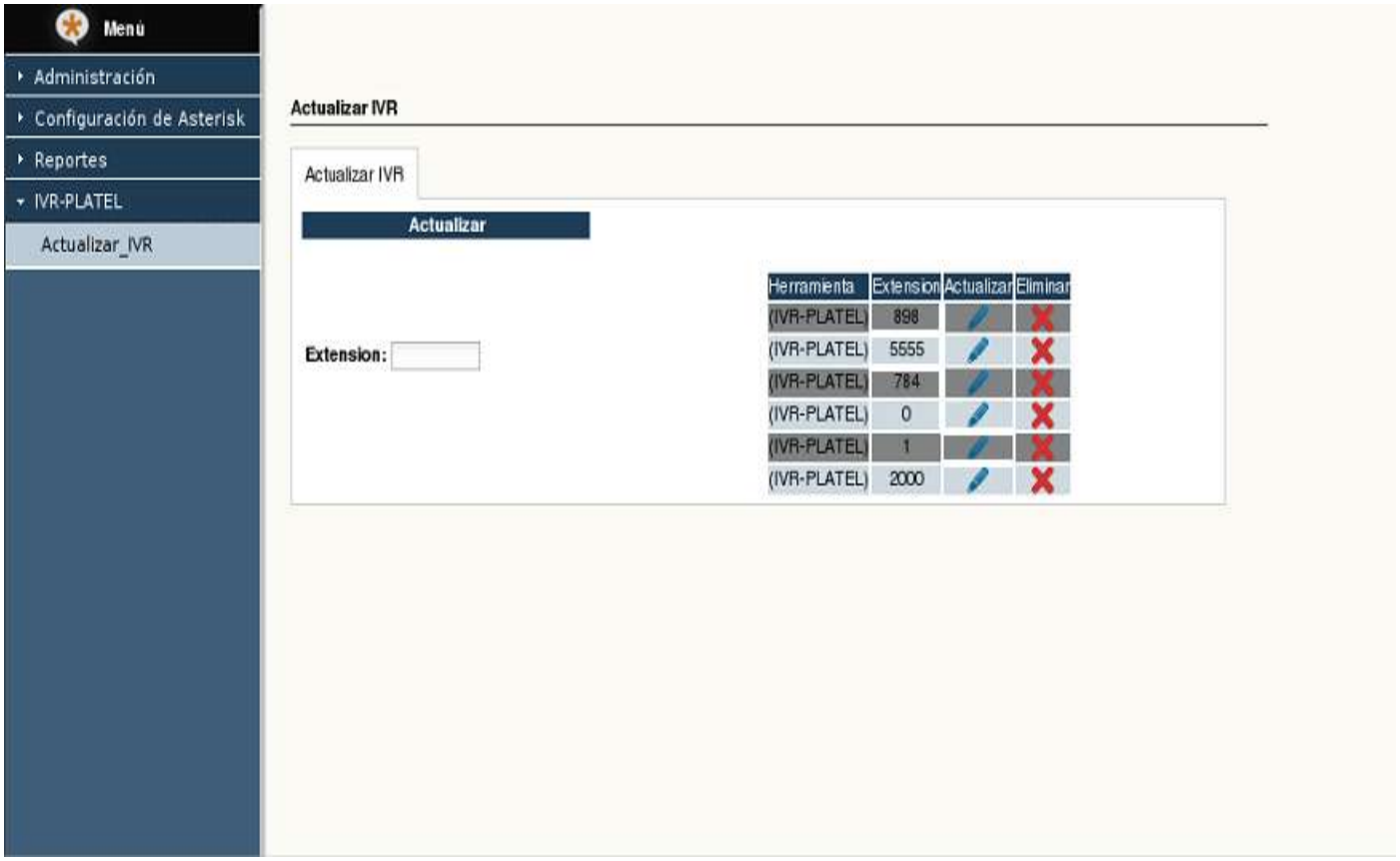

Fig. 7 Actualizar IVR

Fig. 8 Validación Actualizar IVR

Glosario de Términos

Asterisk: Es una aplicación de software libre (bajo licencia GPL) que proporciona funcionalidades de una central telefónica (PBX). Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una tarjeta RDSI tanto básicos como primarios.

Array: Se suele utilizar el termino array (o arreglo) para referirse de forma genérica a matrices de cualquier número de dimensiones.

APIs: Es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Bookmarking: Enlaces a páginas web que se marcan de alguna forma (en un navegador, en una herramienta dedicada, etc) por su interés o para su posterior visualización.

DOM: Es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

DialPlan: Siglas correspondientes al plan de discado de una plataforma telefónica, en Asterisk corresponde a la particularidad de edición del fichero de configuración extensions.conf; contexto, extensión, prioridad, servicio y dentro de estos agrupados en () los datos del servicio.

FTP: En informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

Framework: es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Glosario de Términos

GPL: La GNU General Public License (inglés: Licencia Pública General) es una licencia creada por la Free Software Foundation 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.

HTTP: Define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

IP: Una dirección IP es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red o nivel 3 del modelo de referencia OSI.

IDE: Entorno de desarrollo integrado o en inglés Integrated Development Environment.

('IDE'). Es un programa compuesto por un conjunto de herramientas para un programador desde el que se pueden editar programas, compilarlos y depurarlos.

JSON: Es un formato ligero para el intercambio de datos. Es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Log: Es usado para registrar datos o información sobre quien, que, cuando, donde y por que un evento ocurre para un dispositivo en particular o aplicación.

Lenguajes interpretados: Los lenguajes interpretados (o lenguajes de script) forman un subconjunto de los lenguajes de programación, que incluye aquellos lenguajes cuyos programas son habitualmente ejecutados en un intérprete en vez de compilados.

MAC: En redes de computadoras la dirección MAC (Media Access Control address o dirección de control de acceso al medio) es un identificador de 48 bits (6 bytes) que corresponde de forma única a una tarjeta o interfaz de red.

MVC: Modelo Vista -Controlador.

Máquina Virtual: Es un software que crea un entorno virtual entre la plataforma de la computadora y el usuario final, permitiendo que este ejecute un software determinado.

Netware: Sistema operativo de red. Es una de las plataformas de servicio más fiable para ofrecer acceso seguro y continuado a la red y los recursos de información, sobre todo en cuanto a servidores de archivos.

Glosario de Términos

Open Source: Código abierto (Open Source por sus siglas en inglés) es el término con el que se conoce al software distribuido y desarrollado libremente. Free en inglés puede tener diferentes significados: gratuidad y libertad. Por ello, por un lado, permite pensar en "software por el que no hay que pagar" (software gratuito) y por otro, se adapta al significado que se pretendió originalmente (software que posee ciertas libertades).

ODBC: Es un estándar de acceso a Bases de datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación.

Protocolo: Conjunto de normas que rigen un determinado proceso de comunicación.

Programación multihilo: Un hilo de ejecución en Sistemas Operativos, es similar a un proceso en que ambos representan una secuencia simple de instrucciones ejecutadas en paralelo con otras secuencias. Los hilos permiten dividir un programa en dos o más tareas que corren simultáneamente por medio de la multiprogramación.

PBX: (Private Branch Exchange): Centralita Telefónica Privadas de tamaño particular (para casas u oficinas) que interconecta las extensiones internas de los teléfonos y proporciona conexión con la red telefónica exterior.

Release: Se refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan. En esta fase el producto implementa todas las funciones del diseño y se encuentra libre de cualquier error que suponga un punto muerto en el desarrollo.

RAM: Memoria de acceso aleatorio. Tipo de memoria donde la computadora guarda información para que pueda ser procesada más rápidamente. En la memoria RAM se almacena toda información que está siendo usada en el momento.

Scrum Máster: Es la persona responsable para asegurarse que el proceso de Scrum es utilizado correctamente. Es el responsable de enseñarles cómo hacer todo esto hasta que aprendan, llevándolos en el proceso de convertirse en un equipo auto-administrado.

Staging: Facilita la extracción de datos (los procesos ETL) desde las fuentes de origen de carácter múltiple realizando un pretratado, realiza lo que se conoce como data cleansing (limpieza de datos) y mejorar la calidad de datos.

SSL: Proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere un despliegue de infraestructura de claves públicas (o PKI) para los clientes.

SVN: Es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos.

Scripts: Permiten la automatización de tareas creando pequeñas utilidades. Es muy utilizado para la administración de sistemas UNIX. Son ejecutados por un intérprete de línea de órdenes y usualmente son archivos de texto. También un script puede considerarse una alteración o acción a una determinada plataforma.

Scrum: Define un marco para la gestión de proyectos. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones y la segunda característica importante son las reuniones a lo largo proyecto.

URL: Localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

VoIP: Voice over Internet Protocol, Voz sobre Protocolo de Internet. Es un grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP (Internet Protocol).

Webmail: Es un programa informático, concretamente un cliente de correo electrónico, que provee una interfaz web por la que acceder al correo electrónico.

Widgets: Pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine.

XML: Es un metalenguaje extensible de etiquetas desarrollado por el World Wide We Consortium (W3C). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

Glosario de Términos

XHTML: Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.

XSLT: Actualmente, XSLT es muy usado en la edición web, generando páginas HTML o XHTML. La unión de XML y XSLT permite separar contenido y presentación, aumentando así la productividad.

XMLHttpRequest: Es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores WEB.