

# Universidad de las Ciencias Informáticas

## Facultad 2



**Título:** Sistema para la gestión de grabaciones de llamadas que se transmiten mediante el protocolo RTP.

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

**Autor(es):**

Neyke Rodriguez Lima  
Norbel González Peña

**Tutor(es):**

Ing. Arianna Pérez Carmenates  
Ing. Yanerys Gourrie Fernández

**Co-tutor:**

Ciudad de la Habana, Cuba

2011



## Pensamiento

“Para llegar donde he llegado me he parado en hombros de gigantes.”

Albert Einstein.



**DECLARACIÓN DE AUTORÍA**

Declaramos que somos los únicos autoras del trabajo titulado: "Sistema para la gestión de grabaciones de llamadas que se transmiten mediante el protocolo RTP", y otorgamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Neyke Rodriguez Lima

\_\_\_\_\_  
Norbel Gonzalez Peña

\_\_\_\_\_  
Ing. Arianna Pérez Carmenates

\_\_\_\_\_  
Ing. Yanerys Gourrie Fernández



Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

---

**DATOS DE CONTACTO**



## AGRADECIMIENTOS

Quiero dar las gracias sinceras a mis padres que han sido unos padres maravillosos y han dado todo de sí para que yo llegara a este resultado, a mi amada esposa que ha trabajado a la par mía desde el punto de vista emocional y de apoyo. Quiero agradecer a mi familia por parte materna porque siempre han estado ahí y estoy seguro se sienten orgullosos de mí. A mis hermanos Denia y Roberto, a una parte de mi familia paterna, a la familia de mi esposa que también es mi familia.

Quiero agradecer a mi compañero de tesis, por su apoyo y contribución y a la ingeniera Yarismay su ayuda ha sido invaluable.

Gracias a las tutoras, a los profes del proyecto Rainer, Danae, Yordan y los demás por toda la ayuda y la atención prestada además al profe Ernesto Avilés ya que siempre que tuve dudas no dudó en ayudarme a comprender las cosas, a mis colegas Yury, Nestor, Yasiel Cabrera, Evis su ayuda y el tiempo que me dedicaron fue muy importante para lograr terminar este trabajo, a mis hermanos de aquí de la UCI Jose, Alejandro, David, Rolando, Rafael y a las niñas Leisy, Yenia, Lisbet, Maricé que siempre estuvieron preocupados por el avance y los resultados que iba obteniendo con este trabajo.

Gracias a todos los que de una forma u otra han contribuido al desarrollo de este trabajo de diploma, casi no he mencionado nombres para que no se me quede nadie.

Gracias a todos, los quiero y los llevaré conmigo en mis pensamientos. Gracias

Norbel



Quiero agradecer de manera muy especial a mis padres que no me han fallado ni un solo momento y siempre han estado para apoyarme y guiarme para que yo llegara a este resultado, a mi querida novia Yaris que ha sido un eslabón muy fuerte y ha estado conmigo en las buenas y las malas, su ayuda, su apoyo, comprensión y paciencia han sido inigualables.

Quiero agradecer a mi compañero de tesis, ya que desde el principio él estaba seguro que terminaríamos el presente trabajo y ha hecho un gran esfuerzo para que esto resultara.

Gracias a mis colegas de residencia y a los profes que nos han ayudado.

Gracias a todos los que de una forma u otra han contribuido al desarrollo de este trabajo de diploma.

Gracias a todos. Gracias

Neyke



**DEDICATORIA**

A mis padres que con tanto amor siempre me apoyaron en el transcurso de la carrera. A mi amada esposa Norys por toda su comprensión y ayuda, a mis niños y que esto les sirva de guía e inspiración.

Norbel

A mis padres que con tanto amor siempre me apoyaron en el transcurso de la carrera. A mi novia Yaris que ha sido clave en el desarrollo de este trabajo por toda su comprensión y ayuda, te quiero.

Neyke



## RESUMEN

La Empresa de Telecomunicaciones de Cuba<sup>1</sup>, rectora de las telecomunicaciones en el país, se encuentra inmersa en un proceso de informatización de sus servicios y áreas. Para el logro de tal objetivo se encuentra la compra de una Central Telefónica al proveedor de soluciones de comunicaciones Mitel Networks, pero sólo realiza el pago de la licencia para ciertas funcionalidades debido a la no existencia de presupuesto para la total adquisición, pero sobre todo por ser tecnología propietaria, en esta situación se encuentra el módulo de grabaciones de llamadas telefónicas, quedando actualmente sin utilización.

La situación antes descrita afecta al centro de atención de llamadas de ETECSA debido a que es un requisito indispensable la grabación de las mismas, para de esta forma poder darle solución a las quejas asociadas a los servicios prestados, pudiendo así corroborar la información tanto del operador (a) como del cliente, además de constituir prueba para las auditorías realizadas por las instancias superiores.

El estudio y la investigación realizada arrojaron como resultado la necesidad de realizar la grabación de las llamadas que se transmiten mediante el protocolo RTP para garantizar mayor control en el proceso de atención a los clientes.

El presente trabajo tiene como objetivo desarrollar un sistema para informatizar los procesos relacionados con la gestión de la grabación de llamadas en el centro de atención de llamadas de ETECSA.

**Palabras claves:** grabación, llamadas, protocolo RTP.

---

<sup>1</sup> ETECSA



## Índice

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
<i>Introducción</i> .....	5
1.1 <i>Marco conceptual</i> .....	5
1.2 <i>Estudio de sistemas de grabación de llamadas telefónicas en el mundo</i> .....	6
1.2.1 TRx, grabadora telefónica.....	6
1.2.2 Software de grabación de llamadas de Dharma Call Center .....	7
1.2.3 Call Corder.....	7
1.2.4 CenterWare AVRS.....	8
1.2.5 Grabador de llamadas digitales .....	8
1.2.6 Oreka .....	9
1.2.7 Wireshark.....	9
1.2.8 Asterisk .....	10
1.3 <i>Estudio de sistemas de grabación de llamadas telefónicas en Cuba</i> .....	11
1.4 <i>Análisis crítico de los sistemas estudiados</i> .....	11
1.5 <i>Propuesta de solución</i> .....	12
1.6 <i>Ambiente de desarrollo</i> .....	12
1.6.1 Metodología de desarrollo .....	13
1.6.2 Modelado de procesos de negocio.....	15
1.6.2.1 Notación para el modelado de procesos de negocio: BPM.....	15
1.6.2.2 Lenguaje de modelado: Lenguaje Unificado de Modelado.....	15
1.6.2.3 Herramienta para el modelado.....	15
1.6.2.3.1 Visual paradigm .....	16
1.6.3 Herramientas y tecnologías para el entorno de desarrollo integrado.....	16
1.6.3.1 Lenguajes de desarrollo .....	16
1.6.3.1.1 Java.....	16
1.6.3.1.2 Python versión 2.7 .....	18
1.6.3.2 Herramientas de desarrollo.....	18
1.6.3.2.1 NeatBeans versión 6.9.....	18
1.6.3.2.2 Eclipse versión 3.1 .....	19
1.6.3.3 Framework Django versión 1.2.5 .....	20
1.6.4 Herramientas para el acceso a datos .....	21
1.6.4.1 Sistema gestor de base de datos.....	21
1.6.4.1.1 PostgreSQL versión 8.4.....	21
1.6.4.2 Lenguaje de consulta a base de datos: Lenguaje de consulta estructurado.....	22
1.7 <i>Conclusiones parciales</i> .....	22
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>24</b>
<i>Introducción</i> .....	24



---

<b>2.1</b>	<b>Análisis crítico del flujo actual y objeto de automatización</b>	<b>24</b>
<b>2.2</b>	<b>Descripción del sistema</b>	<b>25</b>
2.2.1	Subsistema de captura y grabación en disco duro	25
2.2.2	Subsistema de filtrado y decodificación	26
2.2.3	Subsistema de gestión de grabación de las llamadas	26
<b>2.3</b>	<b>Requerimientos de software</b>	<b>27</b>
2.3.1	Características funcionales del sistema	27
2.3.2	Características no funcionales del sistema	27
<b>2.4</b>	<b>Planificación</b>	<b>28</b>
2.4.1	Historias de Usuario	28
2.4.2	Estimación de esfuerzo por historias de usuarios	31
2.4.3	Plan de entregas	33
2.4.4	Plan de iteraciones	33
2.4.4.1	Iteración 1	34
2.4.4.2	Iteración 2	34
2.4.4.3	Iteración 3	34
2.4.4.4	Iteración 4	34
2.4.4.5	Iteración 5	34
2.4.4.6	Iteración 6	35
2.4.4.7	Iteración 7	35
2.4.4.8	Iteración 8	35
2.4.4.9	Iteración 9	35
2.4.5	Plan de duración de las iteraciones	35
<b>2.5</b>	<b>Roles del sistema</b>	<b>36</b>
<b>2.6</b>	<b>Descripción de la seguridad</b>	<b>37</b>
<b>2.7</b>	<b>Conclusiones parciales</b>	<b>38</b>
<b>CAPÍTULO 3: DISEÑO, DESARROLLO Y PRUEBAS DEL SISTEMA</b>		<b>39</b>
Introducción		39
3.1	Diseño	39
3.1.1	Arquitectura de la solución	40
3.1.1.1	Patrón arquitectónico: Modelo Vista Controlador	40
3.1.2	Patrones de diseño	41
3.1.2.1	Patrones GOF	42
3.1.2.2	Patrones GRASP	42
3.1.3	Tarjetas CRC	44



3.1.4	Definiciones del diseño .....	45
	Pantalla de Inicio de Sesión.....	46
3.1.4.1	Interfaces del sistema .....	47
3.1.5	Modelo de datos .....	48
3.2	Desarrollo .....	49
3.2.1	Estándares de codificación .....	49
3.2.2	Tareas de ingeniería .....	50
3.2.3	Tratamiento de errores .....	52
3.3	Distribución del sistema .....	52
3.4	Diseño de casos de prueba .....	52
3.4.1	Pruebas unitarias .....	53
3.4.2	Pruebas de aceptación .....	54
3.4.2.1	Pruebas al subsistema de captura y grabación en disco duro .....	55
3.4.2.2	Pruebas al subsistema de filtrado y decodificación .....	57
3.4.2.3	Pruebas al subsistema de gestión de la grabación de las llamadas .....	60
3.4.2.4	Conclusiones parciales .....	61
<b>CONCLUSIONES .....</b>		<b>64</b>
<b>RECOMENDACIONES.....</b>		<b>65</b>
<b>Referencias Bibliográficas .....</b>		<b>66</b>



---

## Índice de Figuras y Tablas

### Figuras

Figura 1. 1 Propuesta de solución.....	12
Figura 2. 1 Flujo actual del proceso .....	25
Figura 3. 1 Interfaz de usuario HU Listar agentes .....	48
Figura 3. 2 Interfaz de usuario HU Buscar .....	48
Figura 3. 3 Modelo de datos .....	49
Figura 3. 4 No conformidades .....	55
Figura 3. 5 Resultado Iteración 1 de prueba al subsistema I.....	55
Figura 3. 6 Resultado Iteración 2 de prueba al subsistema I.....	56
Figura 3. 7 Resultado Iteración 3 de prueba al subsistema I.....	57
Figura 3. 8 Resultado Iteración 1 de prueba al subsistema II.....	58
Figura 3. 9 Resultado Iteración 2 de prueba al subsistema II.....	58
Figura 3. 10 Resultado Iteración 3 de prueba al subsistema II.....	59
Figura 3. 11 Resultado Iteración 1 de prueba al subsistema I.....	60

### Tablas

Tabla 2. 1 Descripción HU Gestionar información de logueo .....	30
Tabla 2. 2 Descripción HU Registrar agentes logueados .....	30
Tabla 2. 3 Descripción HU Capturar paquetes .....	31
Tabla 2. 4 Descripción HU Grabar en disco duro .....	31
Tabla 2. 6 Estimación de esfuerzo por HU .....	32
Tabla 2. 8 Plan de entregas .....	33
Tabla 2. 7 Plan de duración de las iteraciones .....	36
Tabla 2. 9 Roles del sistema .....	37
Tabla 3. 1 Tarjeta CRC Conexión .....	45
Tabla 3. 2 Tarjeta CRC Modelo_Configuración .....	45
Tabla 3. 3 Tarjeta CRC Modelo_Llamada .....	45
Tabla 3. 4 Tarjeta CRC Modelo_Fichero .....	45
Tabla 3. 5 Tarea de ingeniería 1 correspondiente a la HU1 .....	51
Tabla 3. 6 Tarea de ingeniería 2 correspondiente a la HU1 .....	52
Tabla 3. 7 CPR HU Listar agentes .....	61



### INTRODUCCIÓN

El nuevo milenio ha marcado un cambio significativo en la forma en que las administraciones operan y entregan sus servicios, debido, en gran medida al acelerado desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC). El uso de la tecnología para informatizar dichos servicios puede ahorrar a los países considerables sumas de dinero y permitir a los ciudadanos un acceso mucho más rápido y sencillo. Un aspecto importante dentro de la modernización de la sociedad lo constituye la comunicación, la cual ha sido una necesidad desde el surgimiento de la humanidad misma. En la era de la información el conjunto de avances que proporciona la informática, las telecomunicaciones y las tecnologías audiovisuales básicamente proporcionan información, herramientas para su proceso y canales de comunicación.

La Empresa de Telecomunicaciones de Cuba, conocida como ETECSA, rectora de los servicios de telecomunicaciones, para asegurar el proceso de investigación, inversión, producción, prestación de servicios y su comercialización en el país y en el exterior, se encuentra inmersa en un proceso de informatización de sus servicios y áreas. Para el logro de tal objetivo entre los pasos ya dados por ETECSA se encuentra la compra, al proveedor de soluciones de comunicaciones MITEL *Networks*<sup>2</sup>, de un sistema integrado de telefonía y computación orientado a potenciar las labores de la entidad por medio de la comunicación telefónica, o como comúnmente se conoce Central Telefónica, pero sólo se realiza el pago de la licencia para ciertas funcionalidades debido a la no existencia de presupuesto para la total adquisición, en esta situación se encuentra el módulo de grabaciones de llamadas telefónicas quedando actualmente sin utilización.

La situación antes descrito afecta al centro de atención de llamadas de ETECSA, el cual tiene como misión la atención de los clientes a través de las llamadas telefónicas, pero el cliente puede presentar quejas del/la operador(a) alegando calidad percibida en el servicio prestado, por parto resulta de vital importancia la grabación de las llamadas para así corroborar la información tanto del operador (a) como del cliente. El hecho que no se realice el proceso de grabación de llamadas y sólo se establezca la

---

<sup>2</sup> Empresa canadiense con más de 35 años de experiencia como fabricante de sistemas de comunicaciones. Se ha destacado como un líder tecnológico, consiguiendo hitos como el desarrollo e implantación de la tecnología de los tonos DTMF en los sistemas telefónicos o siendo pionera al implantar un microprocesador en una centralita electromecánica. Actualmente Mitel *Networks* se focaliza completamente en la tecnología IP como factor clave para internacionalizarse y adquirir una dimensión a nivel global. [1]



supervisión de algunas por parte de una persona designada para ello realizando la escucha de las mismas, provoca que no se pueda establecer el control de todas las llamadas que se generan en el día, ni lograr comprobar la veracidad de lo planteado por las partes involucradas en caso que se recepcionen quejas sobre el operador (a) generadas en el proceso de atención a los clientes, obstruyendo la misión del centro de atención de llamadas, lo que provoca que no se realiza de manera óptima y eficiente el tratamiento de las quejas.

Por otra parte, es conveniente destacar que tener grabadas todas las llamadas telefónicas entrantes y/o salientes que son realizadas es evidencia física para las auditorías que periódicamente tienen lugar por parte de instancias superiores.

Lo antes expuesto, es lo que genera la no grabación de las llamadas telefónicas, pero resulta de vital importancia explicar que estas llamadas se transmiten mediante el protocolo RTP<sup>3</sup>, que es además el protocolo por el cual la central telefónica adquirida a Mitel realiza la grabación de llamadas telefónicas, que conlleva a la búsqueda de una solución al siguiente **problema científico**: ¿Cómo grabar y gestionar las llamadas que se transmiten por el protocolo?

Partiendo del problema expuesto se propone que el **objeto de estudio** se enfocará a los sistemas para la grabación de llamadas telefónicas, constituyendo el **campo de acción** de la investigación la grabación de las llamadas que se transmiten mediante el protocolo RTP.

Para resolver el problema planteado con anterioridad se establece como **objetivo general**: desarrollar un sistema para informatizar los procesos relacionados con la gestión de grabaciones de llamadas que son transmitidas mediante el protocolo RTP.

Partiendo del análisis del objetivo general se derivan los siguientes **objetivos específicos**:

- ✓ Desarrollar el subsistema para la captura y grabación en disco físico de llamadas que se transmiten por el protocolo RTP.
- ✓ Desarrollar el subsistema para el filtrado y decodificación de llamadas que se transmiten por el protocolo RTP.
- ✓ Desarrollar el subsistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP.

---

<sup>3</sup> Del inglés *Real-time Transport Protocol* o Protocolo de Tiempo Real.



La investigación se basa en la siguiente **idea a defender**: la gestión de la grabación de las llamadas que se transmiten por el protocolo RTP en el centro de llamadas de ETECSA permitirá el tratamiento de las quejas emitidas por los clientes sobre el operador (a) durante el proceso de atención a las llamadas.

Para dar cumplimiento a los objetivos planteados en la investigación, se trazan las siguientes **tareas de investigación**:

- ✓ Realización de entrevistas con especialistas del centro de llamadas para conocer cómo se realizan los procesos asociados a gestión de las quejas del servicio prestado por el operador (a).
- ✓ Estudio de sistemas similares a nivel internacional.
- ✓ Investigación asociada a la captura de llamadas transmitidas por protocolo RTP.
- ✓ Investigación asociada a la decodificación, filtrado y reproducción de llamadas transmitidas por protocolo RTP.

Al ser necesario un mejor entendimiento por parte de los investigadores con el tema objeto de estudio, conocer su estado actual y analizar la factibilidad de una posible solución, para así determinar métodos y técnicas a utilizar, se decide llevar a cabo la investigación exploratoria como estrategia de investigación.

Para el desarrollo de la investigación se utilizaron los siguientes métodos científicos:

Métodos teóricos:

- ✓ Analítico – Sintético: permite efectuar un análisis de la bibliografía disponible para realizar un estudio lo más completo posible del estado del problema a resolver. Posibilita definir los conceptos principales y analizar otras soluciones existentes. Se sintetizan las principales características de las herramientas para el desarrollo del sistema y las ventajas del uso de las mismas.
- ✓ Histórico – Lógico: permite consultar y analizar la información almacenada de sistemas desarrollados anteriormente, sirviendo de ayuda para la implementación de un nuevo sistema con una calidad superior.
- ✓ Modelación: es utilizado para la creación de abstracciones con vistas a explicar la realidad, es decir, modelar las historias de usuario.

Métodos empíricos:



- ✓ Observación científica: permite obtener la información tal y como se observa en la realidad dentro del centro de atención de llamadas de ETECSA, de una forma directa y concreta, para poder de modo objetivo percibir las necesidades reales que debe satisfacer el sistema a implementar.
- ✓ Entrevista: aplicada a personas con amplios conocimientos sobre el proceso de tratamiento de las quejas por parte de los clientes sobre las operadoras durante la atención de las llamadas, conociendo así las características del negocio y los posibles aspectos a informatizar.

El presente documento consta de tres (3) capítulos desarrollados a partir de los estudios realizados sobre la información bibliográfica existente y el encuentro frecuente con directivos de ETECSA y funcionarios del centro de atención de llamadas. A continuación se describe detalladamente el contenido que se aborda en cada uno de ellos:

**Capítulo 1. Fundamentación Teórica:** se detallan los conceptos fundamentales asociados al dominio del problema y al objeto de estudio, realizándose un análisis de la situación actual y de otras soluciones existentes a nivel nacional e internacional. Se muestra el respaldo teórico de los temas tratados en el informe, necesarios para el correcto entendimiento de la solución propuesta. También se describen las tecnologías utilizadas y las herramientas necesarias para realizar el diseño y la implementación del sistema.

**Capítulo 2. Características del Sistema:** se presenta una vista general de la propuesta del sistema. Para ello se realiza un análisis del proceso actual del negocio y un modelado del mismo, analizando sus componentes y cómo interactúan entre sí. Se presentan las historias de usuario.

**Capítulo 3. Diseño, Desarrollo y Prueba del sistema:** se realiza un análisis de cómo debe funcionar el sistema. Se define un diseño simple, siendo la llave para lograr el producto deseado. Se selecciona la metáfora para el sistema y se realizan las tarjetas CRC. Además se describe la solución propuesta con los estándares de implementación. Se establecen los elementos asociados a las unidades de *test* o prueba de calidad que se le realizarán al sistema y se analizan los elementos fundamentales en la solución de la aplicación.



# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

## Introducción

Toda investigación científica requiere de un análisis previo de referentes teóricos que existen sobre el tema en cuestión. Precisamente en el presente capítulo se abordan los principales conceptos referentes a la grabación de llamadas que son transmitidas por el protocolo RTP, además se plasma el resultado del estudio de sistemas que actualmente tienen implementadas funcionalidades relacionadas con los procesos de grabación de llamadas, análisis que permitirá ampliar conocimientos sobre dichos procesos y valorar posibles vías de solución a la problemática planteada. También se realiza la fundamentación de la selección de las herramientas de desarrollo y modelado para la propuesta de solución.

### 1.1 Marco conceptual

Desde la antigüedad existieron medios de comunicación que se han ido perfeccionando con el paso de los años hasta llegar al teléfono, dispositivo indispensable para la razón de ser de muchas entidades en el mundo. En el caso particular de los centros de atención de llamadas el teléfono es siempre el primer punto de contacto personal del cliente con la empresa y la primera imagen que el cliente se lleva, por lo que es indispensable la grabación telefónica, ayudando a supervisar y mejorar continuamente su atención. Pero para poder entender los procesos de grabación de llamadas telefónicas y la presente investigación a continuación se explican conceptos asociados al dominio del problema planteado:

**Llamada telefónica:** en términos de telefonía, una llamada telefónica es la acción de emitir una señal o aviso por medio de un teléfono. Las llamadas telefónicas pueden ser internas, es decir, se realizan dentro de una determinada área que no requieren de costo alguno y las externas se realizan hacia o desde un número que no se encuentra en la red privada de la empresa. Estas pueden ser entrantes y salientes. Las llamadas externas-entrantes son aquellas llamadas realizadas desde un número ubicado en la red pública hacia un número que se encuentra en la red privada y las llamadas externas-salientes son las realizadas desde un número que se encuentra dentro de la red privada hacia un número que se encuentra en la red pública.

**Grabación de llamadas telefónicas:** recogida o registro de las llamadas telefónicas en un soporte para su posterior reproducción.



**Protocolo RTP:** “protocolo específico para la gran demanda de recursos en tiempo real por parte de los usuarios. Algunos de estos recursos son la música, videoconferencia, video, telefonía en Internet y más aplicaciones multimedia. Está formado conjuntamente con el protocolo RTCP<sup>4</sup>, cuya función principal es proporcionar mecanismos de realimentación para informar sobre la calidad en la distribución de los datos. El protocolo RTP se establece en el espacio de usuario y se ejecuta, por lo general, sobre UDP<sup>5</sup>, ya que posee menor retardo que TCP<sup>6</sup>. La función básica de RTP es multiplexar varios flujos de datos en tiempo real en un solo flujo de paquetes UDP, pudiéndose enviar tanto a un solo destino (*unicast*) o múltiples destinos (*multicast*).” [2]

**Voz sobre Protocolo de Internet (VoIP):** grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP. Esto significa que se envía la señal de voz en forma digital, en paquetes de datos, en lugar de enviarla en forma analógica a través de circuitos utilizables sólo por telefonía convencional como las redes PSTN<sup>7</sup>. El tráfico de Voz sobre IP puede circular por cualquier red IP, incluyendo aquellas conectadas a Internet, como por ejemplo las redes de área local (LAN). VoIP es el conjunto de normas, dispositivos, protocolos, en definitiva la tecnología que permite comunicar voz sobre el protocolo IP.

### 1.2 Estudio de sistemas de grabación de llamadas telefónicas en el mundo

En el mundo existe un gran número de sistemas dedicados a la grabación de llamadas por lo que se procede al estudio de los mismos para determinar si pueden ser utilizados en el centro de llamadas de ETECSA. A continuación se muestra el resultado:

#### 1.2.1 TRx, grabadora telefónica

“TRx es un programa que graba llamadas telefónicas en Windows y Mac OS X. Permite grabar manualmente llamadas telefónicas y funciona como un sistema para visualizar la identificación de las llamadas, así como también le permite tener llamadas en espera y reproducirles música y mensajes. TRx fue diseñado para centros de llamadas empresariales donde una copia de TRx es instalada en cada

---

<sup>4</sup> Protocolo de Control RTP.

<sup>5</sup> Del inglés *User Datagram Protocol*.

<sup>6</sup> Del inglés *Transmission Control Protocol* o Protocolo de Control de Transmisión.

<sup>7</sup> Del inglés *Public Switched Telephone Network* o Red Telefónica Pública Conmutada.



estación de trabajo y conectada a cada uno de los teléfonos usando un conector del aparato telefónico para grabar llamadas, módem de voz o una tarjeta de telefonía profesional. También puede ser usado de manera similar en otros ambientes de negocios o en el hogar, donde necesita grabar manualmente una línea telefónica.” [3]

#### 1.2.2 Software de grabación de llamadas de Dharma Call Center

“Es un *software* integrado al *Dharma Call Center*<sup>8</sup> de DataVoice<sup>9</sup> para la grabación de llamadas, para la realización de funciones avanzadas de un centro de atención al cliente. Esta nueva integración, permite al Dharma de DataVoice, mientras los agentes estén ocupados, grabar las llamadas telefónicas que reciba el Centro de Llamadas, las que luego podrán escucharse de manera *on-line* desde cualquier PC conectado a la misma red que los grabadores del sistema.” [4]

“De igual forma, se podrán escuchar las grabaciones maliciosas (intencionadas), así como la de escucha y últimas llamadas, todo ello, desde el mismo terminal del agente que inicie la grabación. Dharma integra la modalidad de grabación, denominada *Voice-Mail*, que realiza la función de buzón de voz personalizado por terminal/agente, con *password*, grabación de mensajes salientes, con consulta por llamadas desde el exterior, etc., facilidad que permite crear un sistema homogéneo de funcionamiento entre las plataformas telefónicas e informáticas, pre requisito principal para convertir el *Call Center* en un sistema multimedia. Además de la particularidad *Voice-Mail*, cuando entra una llamada en condiciones de saturación o fuera de horario, el o los mensajes dejados podrían grabarse en el buzón de un determinado grupo de agentes.” [4]

Dharma también puede grabar una muestra aleatoria de llamadas bajo los criterios de filtros que se configuren. Se pueden aplicar varios filtros simultáneamente, y cada uno de ellos se puede componer de:

- ✓ Grabar una llamada cada cierto tiempo o cada determinada cantidad de llamadas.
- ✓ Grabar cualquier tipo de llamada, o sólo salientes, o sólo entrantes.
- ✓ Grabar las llamadas de un determinado puesto, agente, grupo de agentes, ruta, etc.[4]

#### 1.2.3 Call Corder

---

<sup>8</sup> Producto estrella de DataVoice, un sistema capaz de gestionar hasta 160.000 llamadas a la hora.

<sup>9</sup> Empresa española líder en investigación, desarrollo, fabricación y comercialización de productos y soluciones para Call Centers. Creada en 1987, desde su inicio se ha dedicado exclusivamente a la tecnología de los Call Centers.



“Permite grabar todas las llamadas telefónicas en el disco duro usando un módem. Este *software* es ideal para uso empresarial y totalmente compatible con identificación de llamadas. Almacena las llamadas como archivos de sonido estándar de Windows (WAV) y es muy fácil de utilizar, con Call Corder se puede grabar la llamada telefónica simplemente pulsando un botón. Pero también brinda la opción de grabar las llamadas automáticamente. Los mensajes se pueden editar y reproducir con cualquier *software* de grabación de sonido, también puede trabajar en cooperación con el contestador automático, es decir, inicia la grabación inmediatamente después del mensaje del contestador. El programa es compatible con Windows 98, ME, y Windows NT y 2000 y la licencia es de tipo *shareware*<sup>10</sup>.” [5]

#### 1.2.4 CenterWare AVRS

“CenterWare AVRS es un sistema de grabación de llamadas telefónicas de alto desempeño, cuya principal ventaja es que fue diseñado con los nuevos avances tecnológicos, que le permiten integrarse fácilmente a sus conmutadores telefónicos y a las redes de datos de las empresas, para finalmente aprovechar su infraestructura actual con el menor costo de inversión.” [6] CenterWare AVRS cuenta con herramientas que le permiten centralizar, almacenar, recuperar, reproducir y verificar todas las llamadas telefónicas de un centro de contacto y/o de una empresa. “Cuenta, además, con una interfaz gráfica de ambiente web y de acceso remoto vía internet que facilita el uso de sus aplicaciones de vanguardia, para poder configurar, administrar y monitorear todos los procesos de grabación desde cualquier sitio con acceso a Internet las 24 horas.” [6]

#### 1.2.5 Grabador de llamadas digitales

Es un producto de Telesoft<sup>11</sup> que puede utilizarse en instituciones, empresas y residencias, permite la grabación de llamadas entrantes y salientes, su duración, fecha, y hora de realización, además de controlar tiempos de llamadas y tarifas cobradas por las empresas de telecomunicaciones. “El grabador digital de llamadas se adapta a cualquier computador, admite operaciones en red y convierte las grabaciones a formato WAV para escucharlas en distintos reproductores multimedia y llevarlas en CD o DVD. Este sistema cuenta con un promedio de almacenamiento de 80 gigabytes por cada 5600 horas de

---

<sup>10</sup> Modalidad de distribución de *software*, en la que el usuario puede evaluar de forma gratuita el producto, pero con limitaciones en el tiempo de uso o en algunas de las formas de uso o con restricciones en las capacidades finales.

<sup>11</sup> Compañía que tiene como objetivo brindar soluciones en el ámbito de la integración telefónica al mundo de la computación.



grabación y con la función de grabación avanzada de extensiones PBX, adecuado para soportar la función CRM<sup>12</sup>.” [7]

#### 1.2.6 Oreka

“*Software* que permite la gestión avanzada de grabaciones de las llamadas que se hacen en una centralita Asterisk. El sistema de grabación Oreka funciona grabando llamadas telefónicas tomando como fuente la escucha pasiva del tráfico de voz sobre la red, o directamente capturando el audio del dispositivo de sonido de la PC del agente. De esta forma, se consigue un sistema altamente escalable, porque es independiente de la centralita. Oreka funciona grabando llamadas telefónicas y realiza las características del grabador de llamadas Oreka GPL<sup>13</sup>, funciona en todos los sistemas operativos y se integra con cualquier sistema de telefonía de VoIP.” [8]

“Oreka se distribuye en varias versiones, tanto *open-source* como comerciales, añadiendo esta última opción características como grabación *on-demand*, *multi-site* o diversos módulos para captura de puesto (*Oreka Screen Capture*), monitorización en tiempo real (*Oreka Live Monitoring*) o evaluación de agente (*Oreka Quality Management*). El acceso a los registros de llamadas se realiza utilizando *browsers* estándares como el *Internet Explorer* o *Mozilla Firefox*, sin la instalación de ningún *software* adicional. La escucha de las llamadas se realiza por medio de un Media Player estándar. Además permite al cliente buscar y categorizar grabaciones por fecha y hora de las llamadas, número telefónico origen o destino o cualquier otro parámetro asociado a la llamada. Cuenta con niveles de operación para restringir el acceso a las grabaciones.” [8]

#### 1.2.7 Wireshark

“Wireshark, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de *software* y protocolos, y como una herramienta didáctica para educación. Cuenta con todas las características estándar de un analizador de protocolos.” [9]

“La funcionalidad que provee es similar a la de *tcpdump*, pero añade una interfaz gráfica y muchas opciones de organización y filtrado de información. Así, permite ver todo el tráfico que pasa a través de

---

<sup>12</sup> *Customer Relation Managment.*

<sup>13</sup> *General Public License* por sus siglas del inglés, es una licencia creada por la *Free Software Foundation*, y está orientada principalmente a proteger la libre distribución, modificación y uso de *software*.



una red (usualmente una red Ethernet, aunque es compatible con algunas otras) estableciendo la configuración en modo promiscuo. También incluye una versión basada en texto llamada tshark. Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete. Wireshark incluye un completo lenguaje para filtrar lo que se quiere ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP. Wireshark es *software* libre, y se ejecuta sobre la mayoría de sistemas operativos Unix así como en *Microsoft Windows*.” [9]

#### 1.2.8 Asterisk

“Asterisk es un *software* completo en PBX<sup>14</sup>, *Open Source*<sup>15</sup> en su totalidad, liberado bajo licencia GPL. Asterisk hace VoIP en tres protocolos y puede interoperar con equipos de telefonía estándar básicas usando un *hardware* relativamente sin costo, no necesita ningún hardware adicional para el VoIP. Para interconectarse con algún tipo de telefonía digital o análoga, se apoya de hardware, el más notable de los hardware manufacturado por Asterisk es DIGIUM™.” [10]

“Asterisk soporta una amplia gama de protocolos TMD para el manejo y transmisión de interfaces de telefonía tradicional. Además implementa el tipo de señalización estándar americano y europeo de sistemas de telefonía, permitiendo ser un nexo entre las redes integradas VoIP de siguiente generación y la infraestructura existente.” [11]

“Asterisk dispone de todas las funcionalidades de las grandes centralitas propietarias (Mitel, Panasonic, Alcatel, etc), desde las más básicas (desvíos, capturas, transferencias, multi conferencias) hasta las más avanzadas (Buzones de voz, IVR, etc.). Ha incorporado la mayoría de estándares de telefonía del mercado, tanto los tradicionales con el soporte de puertos de interfaz analógicos y RDSI, como los de telefonía IP. Eso le permite conectarse a las redes públicas de telefonía tradicional e integrarse fácilmente con centralitas tradicionales (no IP) y otras centralitas IP. Utiliza plataforma servidor estándar (de propósito no específico) y tarjetas PCI para los interfaces de telefonía. El sistema puede dar servicio desde 10 usuarios en una sede de una pequeña empresa, hasta 10.000 de una multinacional repartidos en múltiples

---

<sup>14</sup>Del inglés *Private Branch Exchange*, una central telefónica privada.

<sup>15</sup> Código abierto.



sedes. Además que de estudios realizados se estima que puede manejar hasta 300 llamadas a la vez, esto sobre un hardware sencillo como una Pentium IV.” [10]

#### **1.3 Estudio de sistemas de grabación de llamadas telefónicas en Cuba**

En Cuba el proceso de grabación de llamadas se realiza por la vía tradicional y obsoleta, es decir con dispositivos agregados de pequeñas dimensiones con los cuales es posible pinchar el teléfono y así grabar de forma automática todas las conversaciones. Pero vale destacar que se posee experiencia con la grabación de llamadas desde el *software* en la Universidad de las Ciencias Informáticas (UCI) con el proyecto Identidad Venezuela con la implantación de Asterisk.

#### **1.4 Análisis crítico de los sistemas estudiados**

Todas las soluciones antes mencionadas permiten la gestión de grabación de llamadas pero se detectaron inconformidades por las cuales no pueden ser aplicadas en el centro de atención de llamadas de ETECSA, siendo sobresaliente que en todas excepto Wireshark y Asterisk el proceso de grabación en sí no lo realizan mediante el protocolo RTP, tampoco son compatibles con la central telefónica adquirida por el cliente a MITEL, siendo estas, precisamente, las principales exigencias argumentadas por los usuarios finales del producto.

Por otra parte, en todos los sistemas las llamadas son convertidas a formato de sonido y guardadas como tal por lo que habría que tener disponible gran cantidad de recursos para efectuar las salvadas de la información, según las políticas establecidas en el centro de llamadas de ETECSA. También se destaca el problema de que ninguno de estos sistemas incluye su código fuente, siendo esto un riesgo para la seguridad, porque no se conoce verdaderamente cuál es el funcionamiento de los mismos. Amén de lo anterior en algunos de los casos para el funcionamiento óptimo requieren dispositivos agregados, lo que trae como consecuencia una inversión adicional.

A partir del estudio realizado se destaca el hecho que en su generalidad son sistemas que muestran soluciones muy particulares asociadas a la gestión empresarial por lo que tienen un conjunto de funcionalidades innecesarias.

En el caso de Wireshark al necesitar el cliente como mínimo 120 canales simultáneos es un gran volumen de datos por tanto sobrecarga la red, además no es configurable. En el caso de Asterisk , simula la planta



telefónica en su totalidad siendo una opción excelente pero según las políticas de ETECSA tiene el soporte de Mitel entonces Asterisk se convertiría en un obstáculo en la red.

Concluyéndose entonces que son sistemas incapaces de cumplir con las exigencias establecidas para la gestión de grabación de llamadas para el centro de llamadas de ETECSA, lo que se traduce en la necesidad real de un sistema de producción nacional capaz de cumplir con las funcionalidades establecidas por el cliente.

#### 1.5 Propuesta de solución

Para lograr el sistema que el cliente solicita se presenta una propuesta de solución basada en tres aplicaciones, una transparente al usuario encargada de controlar el flujo de llamadas en la pc del/la operador (a), realizar la grabación y guardarlas en el servidor en texto plano. En el caso del segundo subsistema también transparente al usuario, tiene como principal función la de convertir el texto plano en audio. Por su parte la tercera aplicación interactúa con el usuario a través de una interfaz web. Por medio de esta se realiza todo el proceso de gestión de grabación y reproducción de llamadas. Para mayor entendimiento dirigirse al acápite 2.2.

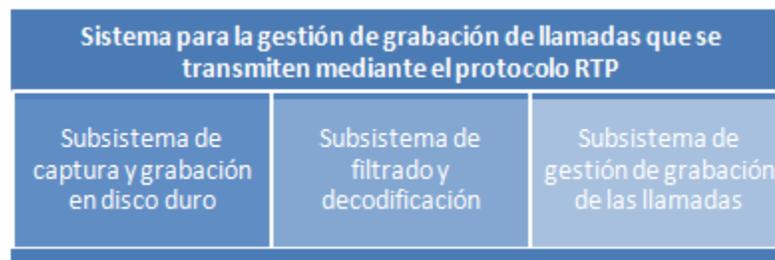


Figura 1. 1 Propuesta de solución

#### 1.6 Ambiente de desarrollo

Con el objetivo de lograr el producto antes mencionado, que no sólo solucione los problemas existentes sino que además proponga nuevas funcionalidades y permita unificar calidad y eficiencia, el mismo se desarrolla sobre la base del proyecto Línea de Desarrollo de Servicios para *Call Center*, el cual ha realizado estudios sobre las herramientas y metodologías de desarrollo a utilizar basándose siempre en las facilidades que ofrecen para el ambiente de desarrollo y las exigencias del cliente. Seguidamente se



exponen características y ventajas de las mismas para poder tener un mayor conocimiento de sus prestaciones y valorar su utilidad.

#### 1.6.1 Metodología de desarrollo

Para desarrollar un *software* es necesario guiar el proceso a través de una metodología, la cual será la encargada de elaborar “el plano” sobre el que se apoyará el equipo de desarrollo. Permitiendo construir *software* de calidad, en el tiempo y con el costo esperado, además proporcionan un sistema flexible ante los cambios que solicite el usuario, ya sea cuando el sistema esté en desarrollo o una vez terminado, contribuyendo a que no se produzcan atrasos en los proyectos por cuestiones organizativas, de entendimiento y de comunicación entre los miembros del equipo de desarrollo, eliminando así la incomodidad de los clientes y desarrolladores ante los cambios que puedan existir. Para pilotar el desarrollo de la solución propuesta se utiliza la metodología Programación Extrema (XP<sup>16</sup>).

“XP es una metodología ágil<sup>17</sup> de desarrollo de *software* utilizada para proyectos de corto plazo, pequeño equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Los principios de XP, se basan en:” [12]

**El juego de planeamiento:** rápidamente determinar el alcance del próximo reléase mediante la combinación de prioridades del negocio y estimaciones técnicas. A medida que la realidad va cambiando el plan, actualizar el mismo.

**Pequeños releases:** poner un sistema simple en producción rápidamente, luego liberar nuevas versiones en ciclos muy cortos.

**Metáfora:** guiar todo el desarrollo con una historia simple y compartida de cómo funciona todo el sistema.

**Diseño simple:** el sistema deberá ser diseñado tan simple como sea posible en cada momento. Complejidad extra es removida apenas es descubierta.

---

<sup>16</sup> arquetipo: XP - eXtreme Programming.

<sup>17</sup> Tipo de metodología que se fundamenta en: la comunicación, entre los usuarios y los desarrolladores; la simplicidad, al desarrollar y codificar los módulos del sistema y la retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.



**Testing:** los programadores continuamente escriben pruebas unitarias, las cuales deben correr sin problemas para que el desarrollo continúe. Los clientes escriben pruebas demostrando que las funcionalidades están terminadas.

**Refactoring:** los programadores reestructuran el sistema sin cambiar su comportamiento para remover duplicación, mejorar la comunicación, simplificar o añadir flexibilidad.

**Programación de a pares:** todo el código de producción es escrito por dos programadores en una máquina.

**Propiedad colectiva del código:** cualquiera puede cambiar código en cualquier parte del sistema en cualquier momento.

**Integración continua:** integrar y hacer *builds* del sistema varias veces por día, cada vez que una tarea se completa.

**Semana de 40-horas:** trabajar no más de 40 horas semanales como regla. Nunca trabajar horas extras durante dos semanas consecutivas.

**Cliente en el lugar de desarrollo:** incluir un cliente real en el equipo, disponible de forma *full-time* para responder preguntas.

**Estándares de codificación:** los programadores escriben todo el código de acuerdo con reglas que enfatizan la comunicación a través del mismo. [12]

XP establece los siguientes derechos del cliente:

- ✓ Decidir qué se implementa.
- ✓ Saber el estado real y el progreso del proyecto.
- ✓ Añadir, cambiar o quitar requerimientos en cualquier momento.
- ✓ Obtener lo máximo de cada semana de trabajo.
- ✓ Obtener un sistema funcionando cada 3 o 4 meses. [12]

Y los derechos del desarrollador:

- ✓ Decidir cómo 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.
- ✓ Cambiar los requerimientos en base a nuevos descubrimientos.[12]



#### 1.6.2 Modelado de procesos de negocio

Seleccionada la metodología de desarrollo a utilizar se recomienda comenzar a modelar los términos de negocio presentes en el área donde se utilizará la solución propuesta. De manera general existen múltiples herramientas que facilitan la modelación de dichos procesos, siendo de gran ayuda para la comprensión de las posibles funcionalidades del sistema.

##### 1.6.2.1 Notación para el modelado de procesos de negocio: BPM

“*Business Process Management* (BPM, por sus siglas en inglés) es una metodología empresarial cuyo objetivo es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio, que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua. Se enfoca en la administración de los procesos del negocio.” [13]

A través del modelado de las actividades y procesos se logra un mejor entendimiento del negocio y muchas veces esto presenta la oportunidad de mejorarlos. “La administración de los procesos permite asegurar que estén ejecutándose eficientemente y obtener información que luego puede ser usada para mejorarlos. Es a través de la información que se obtiene de la ejecución diaria de los procesos, que se puede identificar posibles ineficiencias en los mismos y de esta forma optimizarlos. Para soportar esta estrategia es importante contar con un conjunto de herramientas que den el soporte necesario para cumplir con el ciclo de vida de BPM.” [13]

##### 1.6.2.2 Lenguaje de modelado: Lenguaje Unificado de Modelado

“Lenguaje Unificado de Modelado o *Unified Modeling Language* (UML, por sus siglas en inglés), es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema. Proporciona una forma estándar de escribir los planos de un sistema, tales como procesos del negocio, funciones del sistema, clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes *software* reutilizables. Es la sucesión de una serie de métodos de análisis y diseño orientados a objetos. Está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requisitos y estructuras necesarias para plasmar un sistema de *software* previo al proceso intensivo de escribir código.” [14]

##### 1.6.2.3 Herramienta para el modelado



Las herramientas de modelado permiten incrementar la productividad y el control de la calidad en cualquier proceso de elaboración de *software*, debido que transforman la actividad de desarrollar en un proceso automatizado. A medida que los sistemas que hoy se construyen se tornan más complejos, las herramientas de modelado con UML ofrecen muchos beneficios para todos los involucrados en un proyecto.

#### 1.6.2.3.1 Visual paradigm

“*Visual Paradigm* para UML es una herramienta CASE<sup>18</sup> profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño, construcción, pruebas y despliegue. El *software* de modelado UML ayuda a una rápida construcción de aplicaciones de calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación. Constituye una herramienta de *software* libre de probada utilidad para el analista, fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de *software* de forma fiable a través de la utilización de un enfoque orientado a objetos.” [15] Se caracteriza por:

- ✓ Disponibilidad en múltiples plataformas.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Soporta aplicaciones web.
- ✓ Fácil de instalar y actualizar.
- ✓ Generación de bases de datos. Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- ✓ Ingeniería inversa de bases de datos. Desde Sistemas Gestores de Bases de Datos existentes a diagramas de Entidad-Relación. [15]

### 1.6.3 Herramientas y tecnologías para el entorno de desarrollo integrado

#### 1.6.3.1 Lenguajes de desarrollo

##### 1.6.3.1.1 Java

---

<sup>18</sup> Por sus siglas en inglés *Computer Aided Software Engineering* o Ingeniería de Software Asistida por Computadora



“Java es un lenguaje de Programación Orientado a Objetos (POO), desarrollado por *Sun Microsystems*. El lenguaje en sí mismo toma mucho de la sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Con respecto a la memoria, su gestión no es un problema debido que es gestionada por el propio lenguaje y no por el programador. El lenguaje en su generalidad presenta las siguientes características:

**Lenguaje simple:** Java posee una curva de aprendizaje muy rápida. Debido a su semejanza con C y C++, y dado que la mayoría de los programadores los conoce aunque sea de forma elemental, resulta muy fácil aprender Java.

**Orientado a objetos:** fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.

**Distribuido:** proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir *sockets* y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

**Interpretado y compilado a la vez:** Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los *bytecodes*, semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, debido que los *bytecodes* se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (*run-time*).

**Robusto:** Java fue diseñado para crear *software* altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros).

**Indiferente a la arquitectura:** Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows Nt, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos.

**Portable:** la indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos,



de manera que los programas son iguales en todas las plataformas. Estas dos últimas características se conocen como la *Máquina Virtual Java* (JVM).

**Multihebra:** Java soporta sincronización de múltiples hilos de ejecución (*multithreading*) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.

**Dinámico:** el lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la red.”[16]

#### 1.6.3.1.2 Python versión 2.7

“Python es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Es un lenguaje de programación multiparadigma, lo que significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Python usa tipado dinámico y conteo de referencias para la administración de memoria. Una característica importante de Python es la resolución dinámica de nombres, es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado ligadura dinámica de métodos). Por otra parte Python funciona en Windows, Linux / Unix, Mac OS X, y ha sido adaptada a las aplicaciones Java y .NET máquinas virtuales. Python es libre de usar, incluso para los productos comerciales, debido a su, aprobada por OSI<sup>19</sup>, licencia de código abierto.” [17]

#### 1.6.3.2 Herramientas de desarrollo

##### 1.6.3.2.1 NeatBeans versión 6.9

“NetBeans IDE es un entorno de desarrollo integrado (IDE, por sus siglas en inglés), modular, de base estándar (normalizado), escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una

---

<sup>19</sup> Del inglés *Open System Interconnection* o Modelo de Interconexión de Sistemas Abiertos.



estructura de soporte general (*framework*<sup>20</sup>) para compilar cualquier tipo de aplicación. Se ejecuta en Windows, Linux, Mac OS X y Solaris y además es un producto de código abierto y gratuito sin restricciones de uso.” [18]

“La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs<sup>21</sup> de NetBeans y un archivo especial (*manifest file*) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de *software*.” [18]

#### 1.6.3.2.2 Eclipse versión 3.1

“Eclipse es una plataforma universal para integrar herramientas de desarrollo, con una arquitectura abierta y basada en plug-ins. Eclipse es sustentado por la Fundación Eclipse, una organización independiente sin ánimos de lucro que fomenta una comunidad de código abierto.” [19]

Características:

- ✓ Editor visual con sintaxis coloreada.
- ✓ Compilación incremental de código.
- ✓ Modifica e inspecciona valores de variables.
- ✓ Avisa de los errores cometidos mediante una ventana secundaria.
- ✓ Depura código que resida en una máquina remota.
- ✓ Es soportado en los sistemas operativos, Linux, Windows, Solaris 8 (SPARC/GTK 2), Mac OSX – Mac/Carbon. [19]

“La característica clave de Eclipse es la extensibilidad de sus utilidades con la inclusión de nuevos plug-in.

Eclipse IDE ofrece a los desarrolladores las siguientes ventajas:

- ✓ Información de elementos al posicionar el puntero encima.
- ✓ Documentación de funcionalidades heredadas.

---

<sup>20</sup> Del inglés marco de trabajo.

<sup>21</sup> Del inglés *Application Programming Interface* o interfaz para la programación de aplicaciones.



- ✓ Detección de errores en tiempo real.
- ✓ Facilita plantillas para secciones de código.
- ✓ Formato automático del código.” [19]

#### 1.6.3.3 Framework Django versión 1.2.5

“Django es un *framework* de desarrollo web de código abierto, escrito en Python, que cumple en cierta medida el paradigma del Modelo Vista Controlador, abstrae problemas comunes al desarrollo web y proporciona atajos para tareas de programación frecuentes. La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés *Don't Repeat Yourself*). Python es usado en todas las partes del *framework*, incluso en configuraciones, archivos, y en los modelos de datos.” [20]

“La distribución principal de Django también aglutina aplicaciones que proporcionan un sistema de comentarios, herramientas para syndicar contenido vía RSS y/o Atom, páginas planas que permiten gestionar páginas de contenido sin necesidad de escribir controladores o vistas para esas páginas, y un sistema de redirección de URLs.” [20]

Otras características de Django son:

- ✓ Un mapeador objeto-relacional.
- ✓ Aplicaciones enchufables que pueden instalarse en cualquier página gestionada con Django.
- ✓ Una API de base de datos robusta.
- ✓ Un sistema incorporado de vistas genéricas que ahorra tener que escribir la lógica de ciertas tareas comunes.
- ✓ Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.
- ✓ Un despachador de URLs basado en expresiones regulares.
- ✓ Un sistema *middleware* para desarrollar características adicionales.
- ✓ Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- ✓ Documentación incorporada accesible a través de la aplicación administrativa. [20]



### 1.6.4 Herramientas para el acceso a datos

#### 1.6.4.1 Sistema gestor de base de datos

“Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- ✓ Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- ✓ Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD
- ✓ Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Algunas de las características deseables en un SGBD son:

- ✓ Control de la redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- ✓ Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- ✓ Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad.” [21]

##### 1.6.4.1.1 PostgreSQL versión 8.4

“PostgreSQL es un Sistema de Gestión de Base de Datos Relacional o *Relational Data Base Management System* (RDBMS, por sus siglas en inglés) libre, que está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.”[22] Las principales características de PostgreSQL son:

- ✓ Implementación de estándar SQL92/SQL99.
- ✓ Soporta distintos tipos de datos, además del soporte para los tipos base, también soporta datos de tipo fecha, geométricas, monetarios, elementos gráficos, datos sobre redes, cadenas de bits, etc. Permite la creación de tipos propios.
- ✓ Interfaz con diversos lenguajes: C, C++, Java, Delphi, Python, Perl, PHP, entre otros.



- ✓ Incorpora una estructura de datos *array*.
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Extensible.
- ✓ Multiplataforma.
- ✓ Diseñado para ambientes de alto volumen.
- ✓ Herramientas gráficas de diseño y administración de bases de datos.[22]

### 1.6.4.2 Lenguaje de consulta a base de datos: Lenguaje de consulta estructurado

“El Lenguaje de Consulta Estructurado o *Structured Query Language* (SQL, por sus siglas en inglés) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella. Explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones en éstos últimos. Es un lenguaje declarativo de alto nivel o de no procedimiento, que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación y la orientación a objetos. De esta forma una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros. SQL es mucho más que un lenguaje de consulta de base de datos, ya que permite:

- ✓ Definir la estructura de los datos.
- ✓ Recuperar y manipular datos.
- ✓ Administrar y controlar el acceso a los datos.
- ✓ Compartir datos de forma concurrente.
- ✓ Asegurar su integridad.” [23]

### 1.7 Conclusiones parciales



## Capítulo 1: Fundamentación Teórica

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

---

Luego de definir el marco conceptual es posible un mejor entendimiento de los procesos asociados a la grabación de llamadas. Una vez concluido el análisis de diferentes sistemas que gestionan la grabación de llamadas, se observan sus beneficios y carencias con respecto a requisitos imprescindibles para lograr la gestión de grabación de llamadas en el centro de llamadas de ETECSA, se demostró la necesidad de un sistema de producción nacional que se adapte a los requerimientos dictados por el cliente y satisfaga las expectativas esperadas. Teniendo en cuenta que con el empleo de las herramientas y tecnologías seleccionadas se da la posibilidad de desarrollar el sistema requerido.



## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### Introducción

La informatización de las áreas y servicios en ETECSA se traduce en un funcionamiento superior y mayor organización en la entidad. El centro de atención de llamadas se adapta a las transformaciones tecnológicas con el fin de aprovechar sus beneficios, priorizando, precisamente la gestión de la grabación de llamadas.

Aunque hay que enfatizar que para que esta sea exitosa y se alcance el producto esperado resulta obligatorio el análisis de los aspectos que intervienen en los procesos objetos de automatización. Dicho examen es realizado en el presente capítulo, atendiendo a los procesos relacionados al campo de acción, profundizándose en la fase de planificación propia de la metodología de desarrollo de *software* utilizada, donde se confeccionan las Historias de Usuario (HU).

### 2.1 Análisis crítico del flujo actual y objeto de automatización

El centro de atención de llamadas de ETECSA constituye una herramienta de comunicación y relación con los clientes, que utiliza el teléfono como medio de comunicación básico, gestionado por personas en conjunto a los recursos humanos, físicos y tecnológicos necesarios y disponibles, basados en metodologías de trabajo y procesos determinados y adecuados, para satisfacer las necesidades y dar servicio a cada cliente único con el objeto de atenderlos de la manera que se merecen y se sientan a gusto con la organización.

Al existir el constante intercambio operador (a) con el cliente y realizarse numerosas llamadas en un día pueden producirse quejas por parte del cliente respecto al servicio y la atención prestada, provocando conflictos en la institución sobre la veracidad de lo planteado y la responsabilidad de proporcionar una respuesta a quien la presenta. Para establecer cierto control actualmente la plantilla laboral de la entidad cuenta con un (a) supervisor (a) encargado (a) de efectuar la escucha de algunas llamadas, seleccionadas según los criterios que entienda conveniente, pero no queda registro físico de lo que realmente sucede. Pero tampoco hay evidencia para las auditorías que tienen lugar periódicamente según los procedimientos y políticas de la entidad y las instancias superiores de la misma.



Derivándose de lo antes planteado que el proceso objeto de automatización es la gestión de la grabación de llamadas en tiempo real desde el IP donde está teniendo lugar y la reproducción de las mismas.

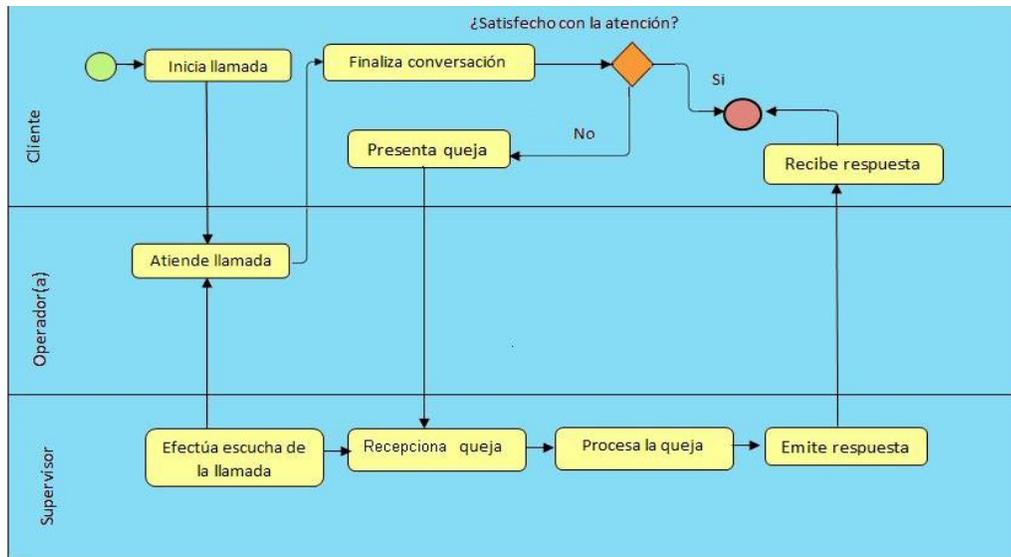


Figura 2. 1 Flujo actual del proceso

## 2.2 Descripción del sistema

En el epígrafe 1.5 se presentó a grandes rasgos el Sistema para la gestión de grabación de llamadas que se transmiten mediante el protocolo RTP, a continuación se detallan las características de cada uno de los subsistemas:

### 2.2.1 Subsistema de captura y grabación en disco duro

Este subsistema es transparente al usuario, se inicia con el Sistema Operativo del servidor, por tanto es un proceso que se encuentra en ejecución todo el tiempo. Para lograr mejores resultados se definen los siguientes módulos dentro de la aplicación:

**Base de datos:** es el encargado de concentrar las clases asociadas a la base de datos, es decir la clase modelo, la de conexión y la de acceso a datos.

**Común:** responsable de establecer la comunicación entre las aplicaciones: Subsistema de captura y grabación y el Subsistema de filtrado y decodificación, traduciéndose en los recursos compartidos, es decir, paquetes, acceso a ficheros de escritura/lectura y rendimiento.



**Hilos:** es el encargado de realizar la captura y la grabación de llamadas en disco duro, mediante los recursos compartidos. Está basado en una pareja de hilos, de los cuales uno captura todos los paquetes que se transmiten por el protocolo RTP y escribe en el recurso compartido o lista de paquetes y el otro recoge el recurso y lo escribe en disco, serializándose en un fichero binario. Se asigna una pareja de hilos para cada canal o cada operador (a).

**Agente:** este módulo tiene la intención de cometer el tratamiento de los agentes y los vínculos a los puestos de trabajo o extensiones.

### 2.2.2 Subsistema de filtrado y decodificación

El subsistema de filtrado y decodificación es transparente al usuario, se inicia a petición del usuario a través de funcionalidades seleccionadas por medio de la interfaz web. Este subsistema también está estructurado por módulos, dos (2) de los cuales son Base de datos y Común ya explicados anteriormente. El resto de los módulos se detallan a continuación:

**Filtrado:** tiene como función principal verificar de todos los paquetes capturados cuales son de audio y los selecciona.

**Audio:** su propósito es crear el audio con las propiedades adecuadas para la reproducción (frecuencia de muestreo 9 200 Hz, la señal mono y formato de audio WAV). Además disminuye el porcentaje de ruido que es producido por interferencias.

### 2.2.3 Subsistema de gestión de grabación de las llamadas

Este subsistema es el que interactúa con el usuario a través de una interfaz web, al igual que las aplicaciones anteriores posee el módulo de Base de datos, propios del sistema, pero el usuario establece comunicación directa con los siguientes módulos:

**Configuración:** es el encargado de establecer la configuración del sistema, así como de los dos (2) subsistemas anteriores.

**Administración:** su objetivo es establecer los permisos de usuario y el acceso a todas las funcionalidades del sistema.



**Gestión y reproducción de las llamadas:** encargado de realizar la gestión de la grabación de llamadas. Incorpora los criterios de grabación y de búsqueda, además permite la confección y visualización de diferentes reportes.

### 2.3 Requerimientos de *software*

“Los requerimientos para un sistema de *software* determinan lo que hará el sistema y definen las restricciones de su operación e implementación. Los mismos se clasifican en funcionales y no funcionales. Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir y se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.” [24]

#### 2.3.1 Características funcionales del sistema

“Las características funcionales son características requeridas del sistema que expresan una capacidad de acción del mismo – una funcionalidad- generalmente expresada en una declaración en forma verbal.” [25] Para describir los servicios que se espera que el sistema propuesto suministre las funcionalidades se detallan en las HU del presente capítulo. Ver acápite 2.4.1 *Historias de Usuario*.

#### 2.3.2 Características no funcionales del sistema

Para definir los requisitos no funcionales se tienen en cuenta aspectos como la usabilidad, la seguridad, el *hardware*, la eficiencia, la fiabilidad, el soporte y las restricciones del diseño, que determinan condiciones y características a tener en cuenta para el sistema. A continuación se muestran los requisitos no funcionales de fiabilidad:

**Fiabilidad** El sistema debe estar disponible las 24 horas durante los 7 días de la semana.

**RnF 1.** No se realizarán mantenimientos preventivos en horario laboral, estos deberán ejecutarse en un horario estipulado.

**RnF 2.** Las fallas del *software* se dividirán en dos categorías:



- ✓ **Simple:** la solución y la actualización se realizarán en línea en un período inferior a cuatro horas.
- ✓ **Complejas:** la solución y actualización se realizarán en un tiempo que se definirá posterior a una evaluación detallada.

**RnF 3.** Sólo se accederá a la BD desde la aplicación, nunca directamente desde el gestor de BD.

**RnF 4.** Se garantizará la consistencia de los datos, se realizarán comprobaciones y validaciones automáticas en todos los casos posibles.

### 2.4 Planificación

En esta fase se define el alcance general del proyecto, el cliente delimita lo que necesita mediante la redacción, a grandes rasgos, de sencillas HU que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto y se estima el tiempo de implementación en base a la información que proporcionan las HU. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración. También se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. “La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando las HU, las que sustituyen a los tradicionales casos de uso.” [26] Una vez obtenidas las HU, “los programadores evalúan rápidamente el esfuerzo que costará la implementación de cada una de ellas. Si alguna de ellas tiene riesgos que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba (*spikes*), para reducir estos riesgos. Luego de realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas (*Release Plan*) en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, dónde en cada una de ellas se desarrolla, prueba e instala unas pocas HU.” [26]

#### 2.4.1 Historias de Usuario



“Las historias de usuario son la forma en que se especifican en *Extreme Programming* los requisitos del sistema. Estas se escriben desde la perspectiva del cliente aunque los desarrolladores pueden brindar ayuda en su identificación. El contenido de las mismas debe ser concreto y sencillo.” [12]

“Las HU también conducen el proceso de los *test* de aceptación, que son empleados para verificar que las historias han sido implementadas correctamente. Existen diferencias entre las HU y la tradicional especificación de requisitos, siendo su principal diferencia el nivel de detalle. Las historias de usuario solamente proporcionan los detalles sobre la estimación de riesgo y cuánto tiempo conllevará la implementación de dicha historia.”[12]

Esta técnica utilizada para especificar los requisitos del *software*. Se registran en tablas en las cuales el cliente describe brevemente las características que el sistema debe poseer. El tratamiento de las HU es muy dinámico y flexible. Cada una debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

Las HU se clasifican de la siguiente manera:

1. Teniendo en cuenta la *Escala Nominal de Prioridad en el Negocio*:

- ✓ **Alta:** se le otorga a las HU que constituyen funcionalidades de vital importancia en el desarrollo del proyecto.
- ✓ **Media:** se le otorga a las HU que para el cliente constituyen funcionalidades a tener en cuenta sin que tengan una afectación directa sobre el proyecto que se está desarrollando.
- ✓ **Baja:** se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de la estructura y que no tienen nada que ver con el proyecto en desarrollo.

2. Teniendo en cuenta la *Escala Nominal de Riesgo en Desarrollo*:

- ✓ **Alta:** se otorga cuando para la implementación de la HU se considera la posible existencia de errores que lleven a la inoperatividad del código.
- ✓ **Media:** se otorga cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.
- ✓ **Baja:** se otorga cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Durante el análisis en la fase de planificación se identificaron un total de diecinueve (19) HU, de las cuales una muestra es descrita a continuación:



## Capítulo 2: Características del sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

Historia de Usuario	
<b>Número:</b> 1	<b>Usuario:</b> Sistema.
<b>Nombre historia:</b> Gestionar información de logueo.	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Medio.
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Neyke Rodriguez y Norbel González	
<b>Descripción:</b> Realiza peticiones al servidor a través de un telnet para detectar el logueo de los usuarios, una vez obtenida la respuesta del servidor: si existen nuevos agentes logueados se crean los agentes y manda a ejecutar HU2, sino manda a ejecutar HU5.	
<b>Prototipo de interfaz de usuario:</b>	
<b>Observaciones</b> Respuesta del servidor: comando que contiene número de agente, nombre, extensión donde se logueó, el número de la extensión y la dirección ip.	

Tabla 2. 1 Descripción HU Gestionar información de logueo

Historia de Usuario	
<b>Número:</b> 2	<b>Usuario:</b> Sistema.
<b>Nombre historia:</b> Registrar agentes logueados.	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Medio.
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Neyke Rodriguez y Norbel González	
<b>Descripción:</b> Registra los nuevos agentes que se han logueado en el sistema. Manda a ejecutar las HU3 y HU4.	
<b>Prototipo de interfaz de usuario:</b>	
<b>Observaciones</b>	

Tabla 2. 2 Descripción HU Registrar agentes logueados



## Capítulo 2: Características del sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

Historia de Usuario	
<b>Número:</b> 3	<b>Usuario:</b> Sistema.
<b>Nombre historia:</b> Capturar paquetes.	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Medio.
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Neyke Rodriguez y Norbel González.	
<b>Descripción:</b> Captura todos los paquetes que se transmiten mediante el protocolo RTP y están asociados a un IP específico.	
<b>Observaciones:</b>	

Tabla 2. 3 Descripción HU Capturar paquetes

Historia de Usuario	
<b>Número:</b> 4	<b>Usuario:</b> Sistema.
<b>Nombre historia:</b> Grabar en disco duro.	
<b>Prioridad en negocio:</b> Alta.	<b>Riesgo en desarrollo:</b> Medio.
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Neyke Rodriguez y Norbel González.	
<b>Descripción:</b> Guarda en disco duro un fichero binario con las capturas realizadas.	
<b>Observaciones:</b>	

Tabla 2. 4 Descripción HU Grabar en disco duro

### 2.4.2 Estimación de esfuerzo por historias de usuarios

“Las estimaciones de esfuerzo asociado a la implementación de las HU se establecen utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente



## Capítulo 2: Características del sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

valen de 1 a 3 puntos. Por otra parte, se mantiene un registro de la velocidad de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.” [28]

Para el desarrollo de la aplicación propuesta se tienen en cuenta situaciones relativas a HU tales como: aparición de nuevas, postergación en cuanto a la iteración en la cual se implementa, no finalización en la iteración planificada y modificación durante la iteración o una vez completada en una iteración previa, realizándose una estimación del esfuerzo para cada una de las HU identificadas, los resultados de este estudio se muestran en la siguiente tabla:

No. HU	Historia de Usuario	Puntos de estimación
1.	Gestionar información de logueo.	2
2.	Registrar agentes logueados.	1
3.	Capturar paquetes.	2
4.	Grabar en disco duro.	1
5.	Detener capturas.	2
6.	Recibir información del subsistema III	1
7.	Obtener interfaces de red activas.	1
8.	Cerrar conexiones.	1
9.	Abrir fichero binario.	1
10.	Filtrar paquetes.	2
11.	Decodificar.	2
12.	Crear fichero de audio.	1
13.	Guardar fichero de audio en disco duro.	1
14.	Listar agentes.	1
15.	Buscar.	1
16.	Grabar llamadas.	1
17.	Grabar llamadas por rango de ip.	1
18.	Mostrar llamadas.	2
19.	Reproducir llamada.	2

Tabla 2. 5 Estimación de esfuerzo por HU



#### 2.4.3 Plan de entregas

“Las HU servirán para crear el plan estimado de entregas. Este plan se usa para crear los planes de iteración para cada iteración. Con cada HU previamente evaluada en tiempo de desarrollo ideal, el cliente las agrupará en orden de importancia.” [26]

Para trazar el plan de entregas del Sistema para la gestión de grabación de llamadas que se transmiten mediante el protocolo RTP se realiza en función de dos parámetros: tiempo de desarrollo ideal y grado de importancia para el cliente. Por otra parte las iteraciones individuales son planificadas en detalle justo antes de que comience cada iteración.

Seguidamente se presenta el plan de entregas estimado para la fase de implementación, como producto del presente plan se realizarán versiones entregables del sistema al finalizar cada una de las iteraciones, teniendo en cuenta que la entrega de estas versiones del *software* ocurre muy temprano y en intervalos muy cortos para maximizar la interacción con el usuario.

Sistema para la gestión de grabación de llamadas que se transmiten mediante el protocolo RTP								
Fin iter. 1	Fin iter. 2	Fin iter. 3	Fin iter. 4	Fin iter. 5	Fin iter. 6	Fin iter. 7	Fin iter. 8	Fin iter. 9
Nov 2011	Dic 2012	Ene 2012	Ene 2012	Feb 2012	Mar 2012	Abr 2012	May 2012	May 2012
Capturav 0.1	Capturav 0.2	Capturav 0.3	Filtradov0 .4	Filtradov0 .5	Filtradov0 .6	Grabació nv0.7	Grabació nv0.8	Grabació nv1.0

Tabla 2. 6 Plan de entregas

#### 2.4.4 Plan de iteraciones

El plan de iteración consiste en seleccionar las HU que, según el plan de entregas, corresponderían a cada iteración. También se eligen qué pruebas de aceptación fallidas se corregirán. Las HU seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración.

“Cada HU se traduce en tareas específicas de programación. Asimismo, para cada HU se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a



las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir.” [28]

“Cada iteración corresponde a un período de tiempo de desarrollo del proyecto de entre una (1) y cuatro (4) semanas.” [28] Para un mejor desempeño y resultados satisfactorios del equipo de desarrollo, se han definido las siguientes iteraciones:

#### **2.4.4.1 Iteración 1**

En la iteración 1 se implementan de las HU correspondientes al subsistema de captura y grabación en disco la HU1 y la HU2, las mismas tienen una prioridad alta dada la complejidad del sistema en su conjunto, obteniéndose así la versión 0.1 del producto dándole la posibilidad al cliente de probar dichas funcionalidades, que son de carácter de crítico para él.

#### **2.4.4.2 Iteración 2**

En esta iteración se desarrollan de las HU restantes del primer subsistema la HU3 y HU4, una vez finalizadas le son mostradas al cliente para que verifique si se ajusta a sus requerimientos.

#### **2.4.4.3 Iteración 3**

Se concluye la implementación del primer subsistema, facilitando y sentando las bases para el desarrollo de los dos (2) subsistemas restantes. Al terminar, el cliente debe realizar el proceso de pruebas correspondientes para lograr el valor agregado que realmente presenta el sistema en su totalidad. En este momento se consigue la versión 0.3.

#### **2.4.4.4 Iteración 4**

El objetivo de esta iteración es la implementación de las HU 6, 7 y 8 propias del subsistema filtrado y la codificación de las llamadas telefónicas grabadas en disco. También al concluir el cliente puede verificar y realizarle las pruebas correspondientes al sistema.

#### **2.4.4.5 Iteración 5**



Al finalizar la iteración será posible abrir el fichero binario y filtrar los paquetes, es decir las HU 9 y 10 serán efectuadas.

#### **2.4.4.6 Iteración 6**

Durante la sexta iteración se ultiman las HU 11, 12 y 13, las que proporcionan al segundo subsistema el terminado que exige el cliente. Una vez concluida se contará con una nueva versión de prueba, la cual será mostrada al cliente con el objetivo de retroalimentar al grupo de trabajo.

#### **2.4.4.7 Iteración 7**

Durante el transcurso de esta iteración se implementan las HU 14,15 y 16, las que dan inicio al Subsistema de gestión de la grabación de llamadas. Al terminar se muestran al cliente para lograr mayor comunicación entre las partes.

#### **2.4.4.8 Iteración 8**

Las HU 17 y 18 son las que se implementan en esta iteración, las cuales también forman parte del tercer subsistema a desarrollar. Una vez concluida la iteración se muestran al cliente porque el mismo tiene el derecho de conocer el estado real y el progreso del proyecto.

#### **2.4.4.9 Iteración 9**

En la novena iteración implementan las HU que terminan con los objetivos propuestos para el sistema, es decir la HU 19. Al finalizar la misma se contará con la versión 1.0 del producto final, por tanto el sistema será puesto en funcionamiento en el centro de atención de llamadas de ETECSA durante un período de tiempo para valorar su desempeño.

### **2.4.5 Plan de duración de las iteraciones**

Con la utilización de la metodología de desarrollo XP y para lograr una mejor distribución del trabajo resulta obligatorio la creación de un plan de duración de iteraciones como parte del ciclo de vida del proyecto; el mismo tiene como propósito exponer la duración de cada iteración así como el orden en que serán implementadas las HU en cada una de ellas. En este caso se realiza un solo plan debido a que



existe un único equipo de desarrolladores. A continuación se muestra cómo quedan distribuidas las HU según el orden en que serán abordadas en cada iteración teniendo en cuenta su prioridad y el tiempo de duración de las mismas.

Iteración	Orden de las HU a implementar	Duración (Semanas)
Iteración 1	1. Gestionar información de logueo.	3
	2. Registrar agentes logueados.	
Iteración 2	1. Capturar paquetes.	3
	2. Grabar en disco duro.	
Iteración 3	1. Detener capturas.	2
Iteración 4	1. Recibir información del subsistema III	3
	2. Obtener interfaces de red activas.	
	3. Cerrar conexiones.	
Iteración 5	1. Abrir fichero binario.	3
	2. Filtrar paquetes.	
Iteración 6	1. Decodificar.	4
	2. Crear fichero de audio.	
	3. Guardar fichero de audio en disco duro.	
Iteración 7	1. Grabar llamadas.	3
	2. Grabar llamadas por rango de ip.	
	3. Buscar llamadas.	
Iteración 8	1. Mostrar llamadas.	3
	2. Seleccionar llamada.	
Iteración 9	1. Reproducir llamada.	2

Tabla 2. 7 Plan de duración de las iteraciones

### 2.5 Roles del sistema

Una vez presentadas las características del sistema es válido destacar que se procede a la ingeniería de roles debido que la información del sistema sólo puede ser accedida y modificada por las personas autorizadas y de la forma autorizada, por lo que se definen roles para acceder al sistema, estos roles



deben asignarse adecuadamente a las diferentes tipos de personas, según sus capacidades y puestos de cada una de ellas. Los permisos se encuentran asociados con los roles y los usuarios son miembros de los roles, de ésta manera se adquieren permisos a los roles. La interacción directa con el sistema lo realizarán los siguientes tipos de persona:

Persona que interactúa con el sistema	Descripción
<b>Administrador</b>	Establece los permisos del sistema y tiene acceso a todas las funcionalidades del sistema con autorización a efectuar modificaciones.
<b>Supervisor</b>	Puede establecer configuraciones en el sistema y es el encargado de dar seguimiento a las llamadas efectuadas.
<b>Usuario (Operador(a))</b>	Tiene permiso de sólo lectura sobre sus llamadas.

Tabla 2. 8 Roles del sistema

### 2.6 Descripción de la seguridad

La seguridad es una característica de cualquier aplicación informática, que indica que está libre de todo peligro, daño o riesgo y que es en cierta manera infalible. Mantener un sistema seguro consiste básicamente en garantizar tres aspectos: confidencialidad, integridad y disponibilidad. La confidencialidad indica que la información de una aplicación ha de ser accedida únicamente por quien está autorizado y de forma controlada; la integridad significa que los datos solamente pueden ser modificados por las personas autorizadas y la disponibilidad se refiere a que la información tiene que estar disponible cuando se necesite.

Para autenticarse en el portal se comprueba el nombre del usuario y la contraseña en la base de datos. En caso de que el usuario no haya cambiado su contraseña en un período de 30 días, al autenticarse en la aplicación se le brinda la opción de cambiarla antes de que pueda realizar cualquier acción. Una vez autenticado, se verifica qué roles tiene asignado el usuario y en dependencia de los permisos que tenga, podrá ejecutar o no las posibles acciones en la aplicación. Por otra parte, el sistema inhabilita una cuenta si se han realizado 5 intentos fallidos de acceso al sistema.



## Capítulo 2: Características del sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

---

Además el *framework* Django intenta mitigar las vulnerabilidades que puedan surgir durante la programación web. Entre los problemas de seguridad que son relevantes para Django se encuentran las inyecciones sql, *Cross-site scripting* (XSS) (Scripting inter-sitio).

### 2.7 Conclusiones parciales

El análisis previo indica que con un mejor entendimiento del flujo actual de las actividades y el objeto de automatización se crean las bases para la comprensión del sistema a implementar. La descripción de las HU, como paso fundamental en el ciclo de desarrollo del proyecto, contribuye a la identificación de los requerimientos del sistema, los cuales sirven de base para la generación de un conjunto de artefactos necesarios en fases posteriores.



### CAPÍTULO 3: DISEÑO, DESARROLLO Y PRUEBAS DEL SISTEMA

#### Introducción

Cada paso durante el ciclo de desarrollo del proyecto cumple un importante papel para obtener el producto final. En el caso del diseño tiene el propósito de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución, que preparan la implementación y prueba del sistema. Como resultado del diseño se procede a la implementación del sistema en términos de componentes: ejecutables, ficheros de código fuente, *scripts*, entre otros. Tiene como objetivo desarrollar la arquitectura y el sistema como un todo, así como definir la organización del código.

Para lograr una correcta implementación, es necesario que la solución propuesta satisfaga las expectativas del usuario final, asegurando que sea operacional o que funcione de acuerdo a los requerimientos. Derivándose entonces la incorporación de pruebas al sistema, actividad para garantizar la calidad del *software*. En el presente capítulo, se aborda lo referente a las fases de diseño, construcción y prueba del sistema.

#### 3.1 Diseño

El diseño es, durante el ciclo de vida del *software* “el proceso de aplicar distintas técnicas y principios con el propósito de definir un producto con los suficientes detalles como para permitir su realización física. Con el diseño se pretende construir un sistema que se ajuste a las limitaciones impuestas por el medio de destino y respete requisitos sobre forma, rendimiento, utilización de recursos, coste, etc.”[30] Para producir el modelo o representación técnica del *software* a desarrollar se plantea el diseño evolutivo, es decir, se realiza a medida que el sistema evoluciona, formando parte de la programación. “En el diseño evolutivo XP aboga por hacer la cosa más simple que pueda funcionar y que no lleve a cabo todo aquello que no sea necesario”. [12]

En el caso del Sistema para la gestión de grabación de llamadas que se transmiten mediante el protocolo RTP, se basa en un diseño simple y claro, es decir, la simplicidad es la llave. En las partes de la implementación que resultan especialmente compleja, se replantea: divide y vencerás. Así, cualquier cambio y modificación es mucho más fácil. En el presente epígrafe se exponen los elementos relevantes para el diseño del sistema.



#### 3.1.1 Arquitectura de la solución

“Desde los sistemas más pequeños hasta los más grandes poseen una estructura y un comportamiento que los hace clasificables según su arquitectura, la cual establece el funcionamiento e interacción entre las partes del *software*.” [29] La propuesta de solución está basada en la arquitectura Cliente/Servidor, una extensión de programación modular que separa los módulos del *software*, brindando una mayor usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones.

“En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un solo programa.” [30]

En esta arquitectura el remitente de una solicitud es conocido como cliente. Sus principales características son:

- ✓ Inicia solicitudes o peticiones.
- ✓ Espera y recibe las respuestas del servidor.
- ✓ Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario. [30]

Al receptor de la solicitud enviada por el cliente se conoce como servidor. Sus principales características son:

- ✓ Al iniciarse esperan a que llegue las solicitudes de los clientes.
- ✓ Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- ✓ Por lo general, aceptan conexiones desde un gran número de clientes.
- ✓ No es frecuente que interactúen directamente con los usuarios finales. [30]

##### 3.1.1.1 Patrón arquitectónico: Modelo Vista Controlador

Para la solución de la aplicación se propone la utilización del patrón arquitectónico Modelo Vista Controlador (MVC). El MVC es aplicable al desarrollo de cualquier aplicación independientemente del lenguaje de programación elegido y separa los datos, la interfaz de usuario y la lógica de control en tres componentes distintos:



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

---

La capa modelo es la representación específica de la información con la cual se operará en el sistema, siendo la encargada de todo el acceso a los datos y las funciones; llamada también lógica de negocio. Esta capa lleva un registro de cada acceso a los datos, poniéndolo en una función individual para garantizar que si cambia el gestor de bases de datos sólo se afecte a este tipo de funciones y no al resto de la aplicación, maximizando de esta forma la adaptabilidad a los cambios de la aplicación.

La capa Vista es la encargada de la interacción con el usuario, mostrando la información de la capa modelo al cliente. Al tener la capa vista separada de la capa controladora se pueden realizar cambios en esta sin tener que tocar nada más que una parte delimitada de código.

“La capa controlador es la responsable de la unión de las capas vista y modelo, esta capa es la que escucha los cambios en la vista y se los envía al modelo, el cual le regresa los datos a la vista.” [31]

La utilización del patrón arquitectónico MVC se puede observar en cada uno de los subsistemas por separado y en el sistema como un todo.

### 3.1.2 Patrones de diseño

El desarrollo de sistemas informáticos motiva la búsqueda de soluciones a problemas comunes en el desarrollo de *software* y otros ámbitos referentes al diseño de interacción o interfaces pues facilitan la reusabilidad, extensibilidad y mantenimiento, problemas muy complejos, pero resueltos con anterioridad. Por esta razón uno de los pasos a tener en cuenta cuando se decide construir un proyecto de *software* es identificar que patrones pueden ser utilizados.

“Los patrones son directrices y principios estructurados que describen un problema común y entregan una buena solución ya probada a la que le dan un nombre. Es una descripción de las clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular.”

[32]

Christopher Alexander, arquitecto destacado, conocido por uno de sus libros “El lenguaje de patrones (*A Pattern Language*)”, expresa que “cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma”. [33]

Para estandarizar el modo en que se realiza el diseño en la aplicación se han tenido en cuenta patrones que logran el equilibrio necesario entre flexibilidad y rendimiento, los cuales se explican a continuación:



#### 3.1.2.1 Patrones GOF<sup>22</sup>

Los patrones de diseño del grupo de GOF se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

**Singleton:** “asegura que sólo se pueda crear una instancia de la clase y ofrece un punto global de acceso a esta instancia.” [33] El uso de este patrón permite que los servicios puedan ser creados sólo una vez. Un ejemplo de su uso es en la clase Diccionario perteneciente al subsistema de captura y grabación en disco duro:

```
private static Diccionario inst = null;
public static Diccionario getInst(){
    if (inst == null)
        inst = new Diccionario();
    return inst;
}
```

#### 3.1.2.2 Patrones GRASP<sup>23</sup>

“Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.” [32]

El uso de estos patrones está totalmente ligado a cada componente desarrollado en el sistema, donde cada uno de ellos posee sólo las funcionalidades acorde a las particularidades que lo caracterizan:

**Creador:** “se aplica para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello.” [32] El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, lo que favorece al mantenimiento del sistema y ofrece mejores oportunidades de reutilización, evidenciándose su uso en la clase Hilo\_Pareja:

```
private Hilo_Capturar productor = null;
private Hilo_Guardar consumidor = null;
```

<sup>22</sup> Del inglés *Gang of Four*, Banda de Cuatro.

<sup>23</sup> Del inglés *General Responsibility Assignment Software Patterns*, Patrones Generales de Software para asignación de Responsabilidades.



```
public Hilo_Pareja(String Ip_Operadora, jpcap.NetworkInterface dispositivo, int
snaplen, boolean promiscuo, int to_ms) {
    Capturas ubicacionCompartida = new Capturas();
    this.productor = new Hilo_Capturar(Ip_Operadora, ubicacionCompartida,
dispositivo, snaplen, promiscuo, to_ms);
    this.consumidor = new Hilo_Guardar(Ip_Operadora, ubicacionCompartida);
}
```

**Experto:** “se aplica para la asignación de responsabilidades a las clases de forma tal que las mismas contengan la información necesaria para poder ejecutar una acción específica. El uso de este patrón permitirá a los objetos valerse de su propia información para hacer lo que se les pide, favorece la existencia de mínimas relaciones entre las clases, lo que permite contar con un sistema robusto y fácil de mantener.” [32] En la clase Hilo\_Capturar se muestra el ejemplo del uso de este patrón:

```
public Hilo_Capturar(String Ip_Operadora, Capturas ubicacionCompartida,
jpcap.NetworkInterface dispositivo, int snaplen, boolean promiscuo, int to_ms) {
    super(Ip_Operadora, ubicacionCompartida);
    try {
        this.filtro = "proto \\udp and host "+getIp_Operadora();
        this.dispositivo = dispositivo;
        this.snaplen = snaplen;
        this.to_ms = to_ms;
        this.promiscuo = promiscuo;
        captor = JpcapCaptor.openDevice(dispositivo, snaplen, promiscuo, to_ms);
        captor.setFilter(filtro, true);
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
```

**Bajo acoplamiento:** “el acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Las diferentes clases controladoras sólo dependen de un único controlador frontal para realizar sus funcionalidades.” [32] Este patrón se tuvo presente debido a la importancia que se le atribuye a realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y que a su vez permitan la reutilización. En las clases Diccionario, Hilo\_Pareja, Hilo\_Capturar e Hilo\_Guardar se usa este patrón.



**Alta cohesión:** “se aplica para realizar un diseño que evite contener clases con un alto grado de abstracción, que asuman responsabilidades que podían haber delegado a otros objetos o que tengan responsabilidades muy complejas.” [32] Se tienen las clases controladoras que se encargan de ejecutar acciones de acuerdo a las peticiones que le llegan y las clases de acceso a datos que interactúan con el modelo, de forma tal que se elimina la sobrecarga de funcionalidades en las clases controladoras.

La arquitectura base y el diseño flexible a través del correcto uso de patrones de diseño en la generación de los artefactos vitales para el desarrollo del sistema; posibilitan crear una entrada apropiada como punto de partida a las actividades de implementación, con la máxima de lograr una mayor calidad del producto y el agrado del cliente.

### 3.1.3 Tarjetas CRC

“La metodología XP para el diseño de las aplicaciones no requiere la representación del sistema mediante diagramas de clases. En su lugar se usan variantes como las tarjetas CRC<sup>24</sup>. Las tarjetas CRC permiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos. Las tarjetas CRC permiten que el equipo completo contribuya en la tarea del diseño. Una tarjeta CRC representa un objeto.” [34]

Las tarjetas CRC son una técnica simple e informal pero efectiva que ha sido propuesta tanto para el modelado conceptual como para el diseño detallado de sistemas Orientado a Objetos, pues cada tarjeta representa un objeto. Se analizan basándose en sus responsabilidades con respecto al sistema. Una tarjeta CRC establece 3 dimensiones las cuales identifican el rol de un objeto en análisis y/o diseño: nombre de la clase (a modo de título en la tarjeta), responsabilidades (se colocan a la izquierda) y las clases que se implican en cada responsabilidad (se colocan a la derecha, en la misma línea que su requerimiento correspondiente). Las mismas determinan el comportamiento de cada actividad.

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

Seguidamente se muestran las tarjetas CRC correspondientes a las clases más relevantes para las funcionalidades del Subsistema de captura y grabación en disco duro:

<sup>24</sup> Cargo o Clase, Responsabilidad y Colaboración.



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

Clase Conexión	
<b>Descripción:</b> clase encargada de realizar la conexión a la BD.	
Responsabilidad	Colaborador
Establecer conexión con la BD.	Modelo_Configuracion.
Cerrar conexión con la BD.	Modelo_Llamada.
Verificar estado de la conexión.	Modelo_Fichero.

Tabla 3. 1 Tarjeta CRC Conexión

Clase Modelo_Configuración	
<b>Descripción:</b> clase encargada de acceder a la BD.	
Responsabilidad	Colaborador
Acceder a la BD.	Conexión.
Consultar datos de configuración.	

Tabla 3. 2 Tarjeta CRC Modelo\_Configuración

Clase Modelo_Llamada	
<b>Descripción:</b> clase encargada de acceder a la BD.	
Responsabilidad	Colaborador
Insertar llamadas.	Conexión.
Consultar datos de Llamada.	

Tabla 3. 3 Tarjeta CRC Modelo\_Llamada

Clase Modelo_Fichero	
<b>Descripción:</b> clase encargada de acceder a la BD.	
Responsabilidad	Colaborador
Insertar fichero.	Conexión.
Consultar datos de Llamada y Fichero.	Modelo_Llamada.
Consultar datos de Fichero.	Fichero.

Tabla 3. 4 Tarjeta CRC Modelo\_Fichero

### 3.1.4 Definiciones del diseño

Para lograr un buen diseño del sistema se deben tener en cuenta determinados aspectos entre los que se encuentran: facilitar al usuario el uso de los objetos con los que tiene que interactuar, brindar información de retorno sobre las acciones que se realizan, que todos los elementos de navegación y la información



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

---

importante se muestren con sólo cargar la página, no utilizar una carga excesiva de imágenes, no incluir sentencias largas y tener en cuenta el contraste entre texto y fondo.

Teniendo en cuenta estos aspectos y las definiciones de diseño establecidas por el proyecto, se muestran a continuación algunas de las pautas empleadas.

### Pantalla de Inicio de Sesión

Es la primera pantalla genérica de la aplicación y contiene:

- ✓ Campos para ingresar a la aplicación: usuario, contraseña, seguridad con su campo de entrada y el botón de acceder a la aplicación.
- ✓ Dimensiones
  - Área total: Largo: 578 px; Altura: 287 px
  - Campo de usuario y contraseña: Largo: 175 px; Altura: 19 px
  - Tipografías: *Tahoma* a 12 puntos, a diferencia del botón entrar, que estará a 14 puntos. Se justificarán a la derecha.

### Tipografía

Se utilizará la tipografía *Tahoma* en todas sus variantes, la utilización de cada una de ellas se pauta en cada una de las secciones en particular. En las entradas de datos y dentro de los cuadros de texto se usará la Tipografía *Arial* en su variante normal.

### Botones con texto

#### Botones de contenido

- ✓ Justificados a la derecha de la acción correspondiente.
- ✓ Los botones con flechas para adicionar o eliminar un término se justificarán al centro de los marcos de contenido, y estarán separados del resto de los botones a 7 px.
- ✓ Cuando se ubica a un lado de la caja de texto, el botón se separa a 25 px de ella.
- ✓ El tamaño depende de la cantidad de caracteres que presente la acción.
- ✓ Tipografía: *Tahoma* 12 puntos Color: R: 255 G:255 B:255 #FFFFFF

#### Botones aceptar y cancelar



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

---

- ✓ Justificados en la parte inferior derecha del área de trabajo, con una separación del borde cuadro de 15 px, y/o justificados a la derecha del cuadro de texto, separados entre ellos a 12 px. Excepto el botón “Cancelar” que siempre debe estar en la extrema derecha.
- ✓ En los casos de aumentar el número de botones en el mismo nivel se ubicarán a la izquierda del botón “aceptar”, separados a 12 px igualmente entre ellos.
- ✓ Tipografía: *Tahoma* 12 puntos Color: R: 0 G: 0 B:0 #000000

### 3.1.4.1 Interfaces del sistema

Las interfaces de un sistema permiten que el usuario interactúe y establezca un contacto más fácil e intuitivo con el mismo. Es por ello que se debe tener en cuenta un conjunto de herramientas y utilidades que permitan al usuario interactuar con los contenidos.

La interfaz proporcionará al usuario el conjunto de posibilidades que utilizará durante todo el tiempo que se relacione con el programa. El usuario, además de entender el mensaje, ha de comprender la mecánica operativa que se le ofrece (sintaxis, órdenes, códigos, abreviaturas, íconos, etc.). Una buena interfaz requiere poco esfuerzo por parte del usuario, simplicidad y funcionalidad. [35]

Para la creación de las interfaces del sistema propuesto se siguieron las pautas y definiciones del diseño trazadas por la dirección del proyecto. A continuación se muestran algunas interfaces del sistema:



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP



Figura 3. 1 Interfaz de usuario HU Listar agentes



Figura 3. 2 Interfaz de usuario HU Buscar

### 3.1.5 Modelo de datos

Un modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Por lo general, un modelo de datos permite describir las estructuras de datos de la base (el tipo de los datos que incluye la base y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregado, borrado, modificación y recuperación de los datos de la base).

“En un enfoque más amplio, un modelo de datos permite describir los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí.” [35]

“Por lo general, un modelo de datos presenta dos sublenguajes: un Lenguaje de Definición de Datos o



*Data Definition Language* (DDL, por sus siglas en inglés), cuya función es describir, de una forma abstracta, las estructuras de datos y las restricciones de integridad; y un Lenguaje de Manipulación de Datos o *Data Manipulation Language*, (DML, por sus siglas en inglés) que se orienta a describir las operaciones de manipulación de los datos. A la parte del DML enfocada a la recuperación de datos, se le suele conocer como Lenguaje de Consulta o *Query Language*.” [35]

En la siguiente figura se muestra el modelo de datos de las clases persistentes que almacenan los datos de la gestión llamadas que se transmiten por el protocolo RTP:

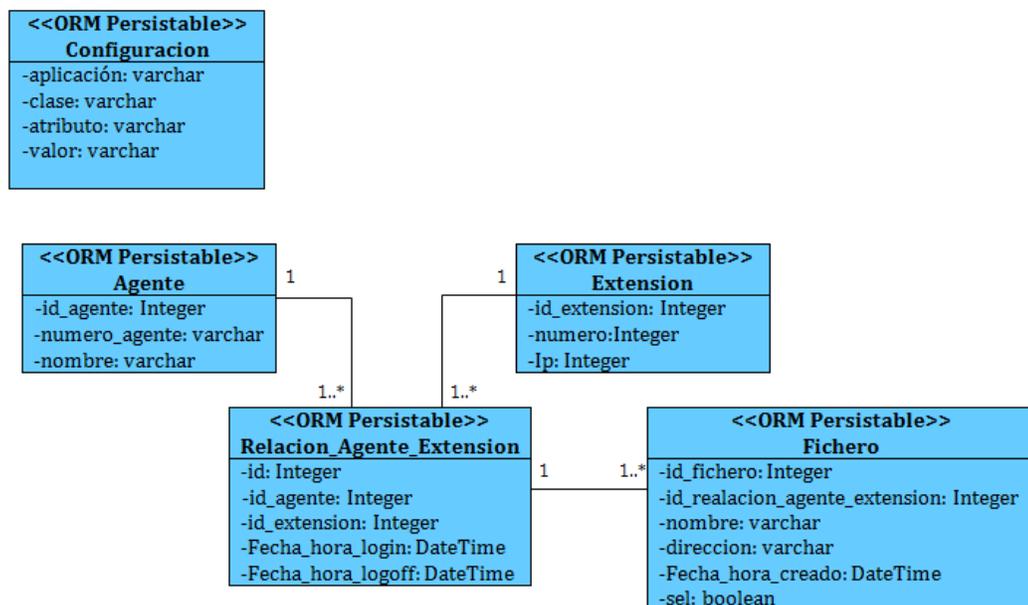


Figura 3. 3 Modelo de datos

## 3.2 Desarrollo

Partiendo de los resultados hasta el momento, con el propósito de clarificar los requisitos restantes y completar el desarrollo del sistema basándose en la línea base de la arquitectura, se procede a la implementación, desarrollo o codificación del sistema. Esta fase se puede ver como un proceso de fabricación, en el que se pone énfasis a la gestión de los recursos y el control de las operaciones para optimizar los costes, la planificación y la calidad del producto final.

### 3.2.1 Estándares de codificación



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

### Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

Uno de los instrumentos que facilitan la tarea de asegurar la calidad del *software* es la adopción de estilos y estándares de codificación, es decir, técnicas de codificación sólidas apoyadas por buenas prácticas de programación. Con el uso de un estándar de codificación bien definido junto a revisiones del código se logra alcanzar un mejor rendimiento para determinada aplicación y caben muchas posibilidades de que un proyecto de *software* se convierta en una aplicación fácil de comprender y de mantener.

Hay que tener en cuenta que para el uso de un estándar de codificación se debe aplicar el mismo en todo momento, o sea, desde el principio hasta el final del proyecto de *software*. El uso de estos estándares tiene innumerables ventajas entre ellas lograr un estilo de código homogéneo asegurando su legibilidad y proveer una guía para el encargado de mantenimiento/actualización del sistema, con código claro y bien documentado. Además ayuda a mejorar el proceso de codificación haciéndolo en gran medida eficiente y en muchos casos reutilizables.

Es debido a estos argumentos que la codificación de los módulos del Sistema para la gestión de grabación de llamadas que se transmiten mediante el protocolo RTP cumple con determinados requisitos, algunos de los cuales son detallados seguidamente:

- ✓ El código está estructurado en forma de bloques, esto favorece la mejor comprensión y lectura del mismo.
- ✓ Para evitar confusión de nombre y tipo se utilizan nombres que describen a los identificadores, en vez de nombres que describen el tipo de identificador.
- ✓ Se definen nombres de clases significativos, que expresan total o parcialmente su significado.
- ✓ Se emplean comentarios en determinadas declaraciones de clases y funciones más complejas.
- ✓ Para la capitalización de los nombres de las clases se utiliza el estilo Pascal y para los atributos el estilo Camello):
  - Pascal: la primera letra en el identificador y la primera letra de cada subsiguiente palabra concatenada se capitalizan. Puede utilizar los identificadores de Pascal case en caso de tres o más caracteres. Por ejemplo: *TiempoEjecucion*.
  - Camello: La primera letra en el identificador está en minúscula y la primera letra de cada subsiguiente palabra concatenada en mayúscula. Por ejemplo: *cuentaBuferOcupado*.

#### 3.2.2 Tareas de ingeniería

A lo largo del desarrollo de las iteraciones se realizará la implementación de las HU que componen cada una de estas. Se llevará a cabo una revisión del plan de iteraciones y se modificará este de ser necesario.



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

De acorde con el plan de desarrollo se descomponen las UH en tareas asignándolas posteriormente a un equipo (o una persona) responsable de su implementación. Estas tareas serán usadas por los programadores, pueden ser descritas utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente.

- ✓ **Número Tarea:** los números deben ser consecutivos.
- ✓ **Número Historia de Usuario:** número de la historia de usuario a la que pertenece la tarea.
- ✓ **Nombre Tarea:** nombre que identifica a la tarea.
- ✓ **Tipo de Tarea:** las tareas pueden ser de: Desarrollo, Corrección, Mejora, Otra (Especificar)
- ✓ **Puntos Estimados:** tiempo estimado en semanas que se le asignará a su desarrollo.
- ✓ **Fecha Inicio:** fecha en que inicia el desarrollo de la tarea.
- ✓ **Fecha Fin:** fecha en que finaliza el desarrollo de la tarea.
- ✓ **Programador Responsable:** nombre y apellidos del programador.
- ✓ **Descripción:** breve descripción de la tarea.

A continuación se presenta una muestra:

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Configurar servidor.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Neyke Rodriguez y Norbel González.	
Descripción: Se establecen los permisos de usuario y contraseña en la planta telefónica.	

Tabla 3. 5 Tarea de ingeniería 1 correspondiente a la HU1

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 1
Nombre Tarea: Implementar telnet en java.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Neyke Rodriguez y Norbel González.	
Descripción: Conectar la aplicación través del socket 5838 a la planta telefónica. Una vez conectado enviar	



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

comandos para obtener los datos: número de agente, nombre, extensión donde se logueó, el número de la extensión y la dirección ip.

Tabla 3. 6 Tarea de ingeniería 2 correspondiente a la HU1

### 3.2.3 Tratamiento de errores

Con el objetivo de garantizar la mayor confiabilidad e integridad en los datos que utiliza el sistema se adoptan las siguientes estrategias para el tratamiento de errores:

- ✓ Se validan los campos de los formularios de las interfaces pertenecientes a los módulos del subsistema de gestión de grabación de las llamadas.
- ✓ La información que contiene los mensajes de errores detectados se muestra con un lenguaje claro, legible y sencillo de entender.
- ✓ Dado el caso que la información introducida en un formulario sea incorrecta o incompleta, se señalarán los campos que contienen el error señalados con una breve descripción.

### 3.3 Distribución del sistema

El Sistema para la gestión de la grabación de llamadas que se transmiten mediante el protocolo RTP estará disponible en el centro de atención de llamadas de ETECSA. Las estaciones de trabajo se comunicarán mediante el protocolo HTTP (contenido dentro del protocolo TCP/IP) con el servidor de aplicaciones donde se encuentra la aplicación web y el mismo estará en constante intercambio de información con el servidor de base de datos general por medio del protocolo TPC/IP.

### 3.4 Diseño de casos de prueba

El desarrollo del *software* implica una serie de actividades de producción en las que las posibilidades de que aparezca la falibilidad humana son comunes. Los errores pueden empezar a darse desde el primer momento del proceso, en el que los objetivos pueden estar especificados de forma errónea e imperfecta; así como en los posteriores pasos del diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del *software* ha de ir acompañado de una actividad que garantice la calidad. Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

### Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

La prueba de *software* es un elemento crítico para la garantía de la calidad del mismo y representa una revisión final de las especificaciones del diseño y de la codificación. “El objetivo de la prueba es: diseñar pruebas que saquen a la luz diferentes clases de errores con la menor cantidad de tiempo y espacio.” (24) El diseño de casos de prueba consiste en la confección de los distintos casos de prueba según la técnica identificada previamente. Cada uno de estos va acompañado del resultado que ha de producir el *software* al ejecutarlo para detectar un posible fallo en el programa. Definen un conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular. Cada técnica de pruebas proporciona unos criterios distintos para generar estos casos o datos de prueba.

En el caso específico de XP, es una metodología que “anima a probar tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones. XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y son diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguieron las funcionalidades requeridas, diseñadas por el cliente final.” [36]

#### 3.4.1 Pruebas unitarias

Las pruebas unitarias son las encargadas de verificar el código y son diseñadas por los programadores, es decir, se basan en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del *software* proponiendo casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado. Esto genera gran cantidad de caminos posibles por lo que hay que dedicar esfuerzos a la determinación de las condiciones de prueba que se van a verificar. Cada uno de los desarrolladores tiene que ir probando constantemente lo que va obteniendo en el transcurso de la implementación del sistema, para garantizar que las funcionalidades exigidas por el cliente estén siendo implementadas correctamente. Las pruebas unitarias al Sistema para la gestión de la grabación de llamadas que se transmiten mediante el protocolo RTP fueron desarrolladas constantemente cada vez que se terminaba de implementar alguna funcionalidad, probándola directamente en el entorno real. Estas pruebas realizadas proporcionan beneficios tales como:



- ✓ El programador puede realizar cambios de forma segura respaldado por efectivos casos de prueba.
- ✓ Permite saber si una determinada funcionalidad se puede agregar al sistema existente sin alterar el funcionamiento actual del mismo.
- ✓ Brindan al programador una inmediata retroalimentación de cómo está realizando su trabajo.

#### 3.4.2 Pruebas de aceptación

Las pruebas funcionales son la mejor forma de probar la aplicación de extremo a extremo: desde la petición realizada por un navegador hasta la respuesta enviada por el servidor. Las pruebas funcionales prueban todas las capas de la aplicación. En realidad, son muy similares a lo que se hace manualmente cada vez que se añade o modifica una acción y se prueban dichos cambios en el navegador para comprobar que todo funciona bien al pulsar sobre los enlaces y botones y que todos los elementos se muestran correctamente en la página. En otras palabras, lo que se hace es probar un escenario correspondiente a la historia de usuario que se acaba de implementar en la aplicación.

“Las pruebas de aceptación son consideradas como pruebas de caja negra (*Black box system tests*). Las pruebas de aceptación son más importantes que las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente.” [37] Las pruebas de aceptación se elaboran a lo largo de la iteración, en paralelo con el desarrollo del sistema, y adaptándose a los cambios que el mismo sufra. Por las características específicas del Sistema para la gestión de grabación de llamadas que se transmiten mediante el protocolo RTP que está constituido por tres subsistemas, dos de los cuales son transparentes al usuario, las pruebas de aceptación se realizaron por separado a cada subsistema. El siguiente gráfico ilustra cómo se comportaron las iteraciones de pruebas:

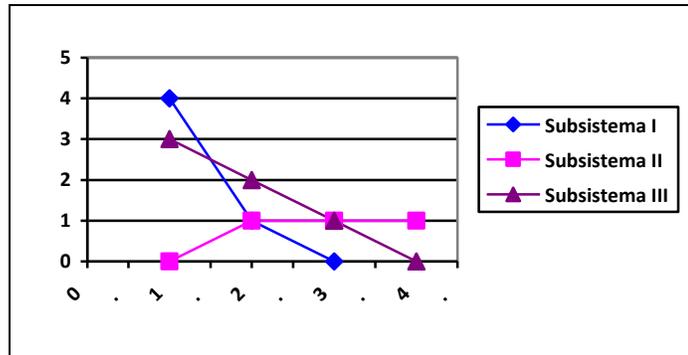


Figura 3. 4 No conformidades

### 3.4.2.1 Pruebas al subsistema de captura y grabación en disco duro

Las pruebas al subsistema de captura y grabación en disco duro están condicionadas por el hecho que información debe ser mostrada por consola debido que no hay interfaces de usuarios. Se realizaron un total de tres (3) iteraciones de pruebas en las que el cliente pudo determinar las no conformidades.

En la primera iteración los datos de entrada (dirección IP de los teléfonos) al subsistema se introducen manualmente, el sistema efectúa la captura desde esas direcciones y los datos de salida son impresos en consola. En este momento fueron detectadas cuatro (4) no conformidades.

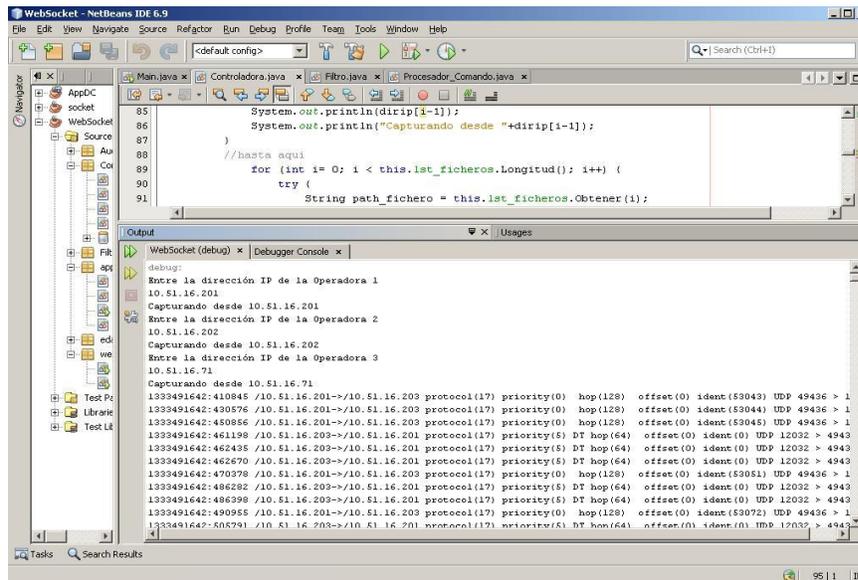


Figura 3. 5 Resultado Iteración 1 de prueba al subsistema I



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

### Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

Para una segunda iteración ya se habían corregido las no conformidades encontradas en la primera. El proceso se realizó de la siguiente forma: una persona logueada como agente en la plataforma telefónica atiende las llamadas entrantes de los clientes. El sistema inicia la captura desde el teléfono del agente y guarda las llamadas según sean atendidos. Se loguearon un total de tres (3) agentes porque el sistema no soportaba número mayor, siendo esta la única no conformidad revelada.

```
164     Datos datos = new Datos(buffer_listado);
165
166     datosVoz = datos.convertirByteADouble();
167     Ruido rd = new Ruido(datosVoz, 500, FrecuenciaDeSampleo);
168     datosVoz = rd.eliminarSeñalesInutiles();
169
170     buffer_listado = datos.convertirDoubleAByte(datosVoz);
171
172     datosVoz = null;
173     System.gc();
174     //hasta aqui
175
176     String path_fichero= SerializarAudio(ubicacionAudio, nombreBuffer, buffer_listado);
177     buffer_listado = null;
178     System.gc();
```

Output: WebSocket (debug) x | Debugger Console x | Usages

```
debug:
Agente logueado: 75 (Norbel González Peña) en 10.51.16.201
Capturando desde 10.51.16.201
Agente logueado: 76 (Neyke Rodriguez) en 10.51.16.202
Capturando desde 10.51.16.202
Agente logueado: 77 (Maria Pérez) en 10.51.16.71
Capturando desde 10.51.16.71
Guardando captura en: D:\Grabaciones\10.51.16.201\10.51.16.201_3-4-2012_10-00-48.mcap
Guardando captura en: D:\Grabaciones\10.51.16.201\10.51.16.201_3-4-2012_10-02-05.mcap
Guardando captura en: D:\Grabaciones\10.51.16.201\10.51.16.201_3-4-2012_10-03-27.mcap
Guardando captura en: D:\Grabaciones\10.51.16.201\10.51.16.201_3-4-2012_10-05-40.mcap
Guardando captura en: D:\Grabaciones\10.51.16.201\10.51.16.201_3-4-2012_10-06-51.mcap
```

Figura 3. 6 Resultado Iteración 2 de prueba al subsistema I

En un tercer momento de iteración de las pruebas al subsistema se tuvo en cuenta una mayor magnitud de las mismas, por lo tanto fueron materializadas en el local correspondiente al laboratorio 301 del docente 1 en la UCI, para ello se seleccionaron un total de 20 estaciones de trabajo, las cuales incluyen 5 teléfonos y las restantes en computadoras con simulador de teléfono (XLite<sup>25</sup>). En un momento determinado se loguean en la plataforma telefónica e inician conversación, en consola se va mostrando de forma actualizada lo ocurrido en dicho proceso:

<sup>25</sup> Programa para hacer llamadas desde la PC, X-Lite es una práctica aplicación de las denominadas *softphone*, es decir, es un *software* que hace una simulación de teléfono convencional por computadora. Permite usar la computadora para hacer llamadas a otros *softphones* o a otros teléfonos convencionales usando un VoIP.



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

id_fi	id_rama	localizacion	nombre	expira
35	D:/Test/Grab10.51.16.71/		10.51.16.71_74-2012_0-36-49.mcap	07/05/2012 12:36
36	D:/Test/Grab10.51.16.71/		10.51.16.71_74-2012_0-36-55.mcap	07/05/2012 12:36
37	D:/Test/Grab10.51.16.71/		10.51.16.71_74-2012_0-37-1.mcap	07/05/2012 12:37
38	D:/Test/Grab10.51.16.71/		10.51.16.71_74-2012_0-37-6.mcap	07/05/2012 12:37
39	D:/Test/Grab10.51.16.71/		10.51.16.71_74-2012_0-37-12.mcap	07/05/2012 12:37
40	D:/Test/Grab10.51.16.71/		10.51.16.71_74-2012_0-37-18.mcap	07/05/2012 12:37
41	D:/Test/Grab10.51.16.71/		10.51.16.71_74-2012_0-37-24.mcap	07/05/2012 12:37
42	D:/Test/Grab10.51.16.71/		10.51.16.71_74-2012_0-37-30.mcap	07/05/2012 12:37
43	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-50-25.mcap	10/05/2012 10:50
44	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-50-30.mcap	10/05/2012 10:50
45	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-50-36.mcap	10/05/2012 10:50
46	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-50-42.mcap	10/05/2012 10:50
47	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-50-48.mcap	10/05/2012 10:50
48	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-50-53.mcap	10/05/2012 10:50
49	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-50-59.mcap	10/05/2012 10:51
50	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-51-5.mcap	10/05/2012 10:51
51	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-51-12.mcap	10/05/2012 10:51
52	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-51-17.mcap	10/05/2012 10:51
53	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-51-23.mcap	10/05/2012 10:51
54	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-51-29.mcap	10/05/2012 10:51
55	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-51-35.mcap	10/05/2012 10:51
56	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-51-41.mcap	10/05/2012 10:51
57	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-51-47.mcap	10/05/2012 10:51
58	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-51-53.mcap	10/05/2012 10:51
59	D:/Test/Grab10.51.16.71/		10.51.16.71_104-2012_10-51-59.mcap	10/05/2012 10:52

Figura 3. 7 Resultado Iteración 3 de prueba al subsistema I

Una vez concluido se accede a la base de datos y se comprueba en la tabla Fichero cada una de las capturas realizadas. Demostrando que lo mostrado en consola coincide con lo guardado en la base de datos.

### 3.4.2.2 Pruebas al subsistema de filtrado y decodificación

Al contar con los ficheros de captura del primer subsistema se puede proceder a realizarle las pruebas al subsistema de filtrado y decodificación. Para hacer efectivas estas pruebas se formalizaron 4 iteraciones. La primera iteración tuvo como principal propósito comprobar el filtrado de los paquetes. Se abrieron un total de 3 ficheros de captura y se le efectuó a cada uno el filtrado. En consola se mostró la cantidad inicial de paquetes en cada fichero y al concluir se pudo comprobar que se habían eliminado un porcentaje porque no coincidían con las características de los paquetes de audio.



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

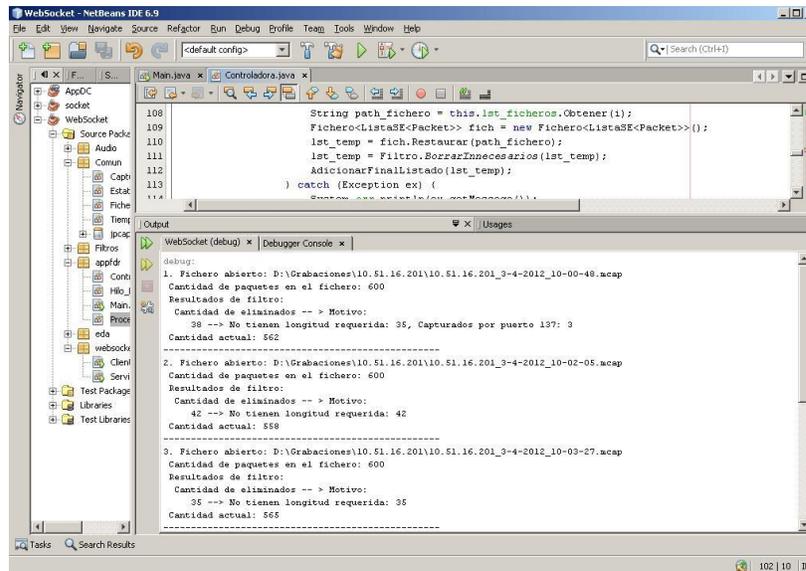


Figura 3. 8 Resultado Iteración 1 de prueba al subsistema II

La segunda iteración tuvo también como objetivo el filtrado, pudiéndose detectar que al abrir más de 5 ficheros el sistema daba error por exceso de memoria. Aunque es válida la aclaración que una llamada según su duración puede tener más de 5 ficheros. El cliente pidió en este momento que la grabación no se ejecutase por llamada sino por un rango de tiempo establecido con anterioridad sin que fuese relevante cuantas llamadas ocurriesen en ese período de tiempo.

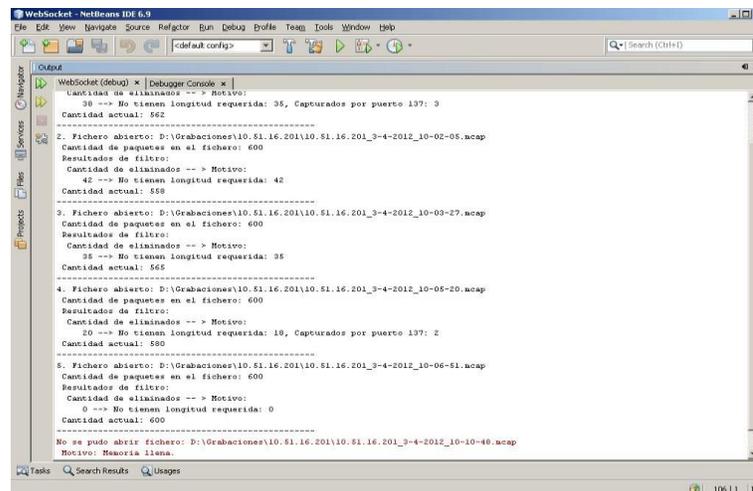


Figura 3. 9 Resultado Iteración 2 de prueba al subsistema II



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

Para una tercera iteración quedaba resuelta la situación anterior y se sumaba la decodificación. La prueba consistía básicamente en que una vez realizado el filtrado se debía obtener del listado la información relevante de cada paquete, para ello en consola se mostraba para cada uno la cantidad de datos iniciales y los datos eliminados. Teniendo al finalizar como no conformidad que entre los datos de la información relevante se conservaba el ruido.

```
89      System.out.println(" Descartados 54 bytes (Ethernet: 14, TCP/IP: 20, UDP: 8, RTP: 12)");
90      System.out.println(" Bytes: "+result[i-1]);
91      System.out.println(" Cantidad de bytes de información relevante: 160");
92      System.out.println("-----");
```

```
debug:
Paquete no. : 1 Cantidad de bytes: 214
Descartados 54 bytes (Ethernet: 14, TCP/IP: 20, UDP: 8, RTP: 12)
Bytes: [10,25,37,-14,115,-110,-108,78,64,-56,...]
Cantidad de bytes de información relevante: 160
-----
Paquete no. : 2 Cantidad de bytes: 214
Descartados 54 bytes (Ethernet: 14, TCP/IP: 20, UDP: 8, RTP: 12)
Bytes: [67,-78,84,83,-54,-20,0,0,114,111,...]
Cantidad de bytes de información relevante: 160
-----
Paquete no. : 3 Cantidad de bytes: 214
Descartados 54 bytes (Ethernet: 14, TCP/IP: 20, UDP: 8, RTP: 12)
Bytes: [30,-42,-112,-5,115,78,42,46,-80,-69,...]
Cantidad de bytes de información relevante: 160
-----
Paquete no. : 4 Cantidad de bytes: 214
Descartados 54 bytes (Ethernet: 14, TCP/IP: 20, UDP: 8, RTP: 12)
Bytes: [-1,43,29,0,0,-113,108,-78,-64,-56,...]
Cantidad de bytes de información relevante: 160
-----
Paquete no. : 5 Cantidad de bytes: 214
Descartados 54 bytes (Ethernet: 14, TCP/IP: 20, UDP: 8, RTP: 12)
Bytes: [-100,-125,-37,-114,-15,10,108,108,64,65,...]
```

Figura 3. 10 Resultado Iteración 3 de prueba al subsistema II

Al efectuar la cuarta iteración se había eliminado aproximadamente un 60 por ciento del ruido inicial y en consola se publicaban los datos necesarios. El cliente al tener la posibilidad de escuchar la reproducción pudo percibir que se llegaron a cumplir con las expectativas del segundo subsistema. Aún queda un 40 por ciento del ruido en la reproducción de las llamadas, se trabaja aún para que quede eliminado totalmente.

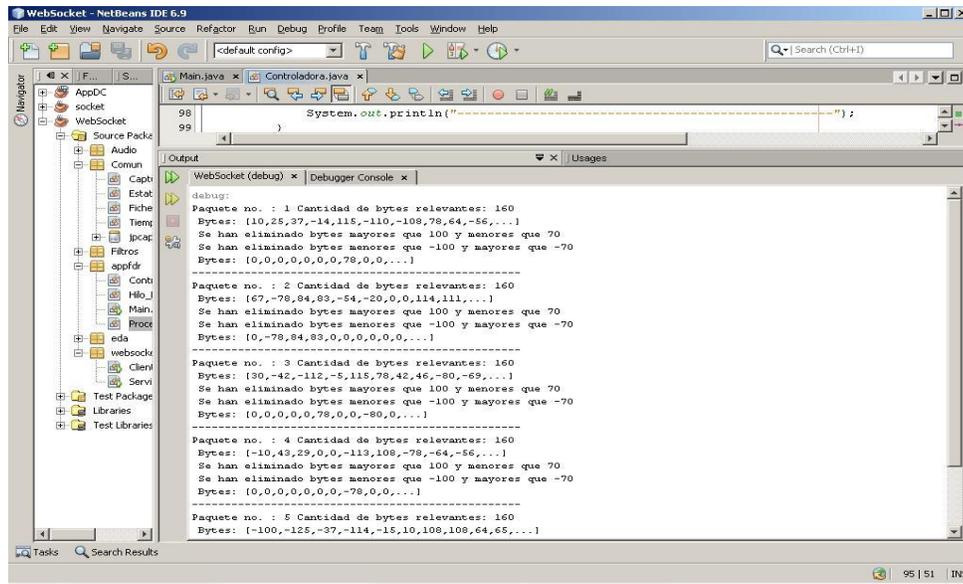


Figura 3. 11 Resultado Iteración 1 de prueba al subsistema I

#### 3.4.2.3 Pruebas al subsistema de gestión de la grabación de las llamadas

Al ser este el subsistema con el que el cliente interactúa a través de una interfaz web las pruebas que se realizan son las pruebas de caja negra. Dichas pruebas se registran en tablas, las cuales están divididas en las siguientes secciones:

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



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

Se muestra el caso de prueba a la HU 14, la primera definida para el subsistema web, a esta HU se le realizaron las siguientes pruebas:

Prueba #1: Se selecciona en el menú principal el submenú listado de agentes. Se espera un resultado y se califica de satisfactoria o no satisfactoria.

### CPR<sup>26</sup> 1: “Listar agentes”

#### Descripción de la funcionalidad

Mostrar el listado de los agentes logueados en la aplicación.

#### Flujo Central

Seleccionar del menú principal el submenú Listado de agentes

Esperar respuesta del sistema.

#### Condiciones de Ejecución

Existan agentes logueados

#### Iteraciones

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Se selecciona la opción “Listado de agentes”		El sistema muestra una tabla con el número y Nombre y Apellidos de los agentes autenticados en la planta telefónica	Satisfactorio	

Tabla 3. 7 CPR HU Listar agentes

### 3.4.2.4 Conclusiones parciales

La realización de las actividades correspondientes a la etapa del diseño expresa la organización del sistema y aumenta la facilidad de desarrollo en paralelo y las facilidades de modificación y desarrollo, necesarias para la implementación y mantenimiento. La determinación de las pautas de diseño a utilizar posibilita la obtención de una interfaz profesional y a la vez amigable. El uso de estándares de codificación en conjunto con la estrategia de tratamiento de errores permite implementar un código organizado, de alta calidad y fácil de mantener. Una vez finalizada la etapa de implementación se realizaron varias iteraciones de pruebas unitarias y de sistema que posibilitaron la detección de no conformidades que fueron

<sup>26</sup> Caso de prueba



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

---

corregidas, posibilitando que el sistema propuesto presente el funcionamiento requerido. Comprobándose finalmente que la aplicación está lista para su despliegue.



## Capítulo 3: Diseño, Desarrollo y Prueba del Sistema

Sistema para la gestión de grabaciones de llamadas que se transmiten por el protocolo RTP

---



### CONCLUSIONES

Una vez concluida la presente investigación queda desarrollado el Sistema para la gestión de grabaciones de llamadas que se transmiten mediante el protocolo RTP dando cumplimiento a los objetivos trazados y a través de las diferentes etapas se pudo palpar que:

- ✓ Con el análisis del estudio del marco teórico de la investigación se determinó la necesidad en el centro de atención de llamadas de ETECSA un sistema capaz de realizar la grabación de las llamadas que se transmiten mediante el protocolo RTP.
- ✓ Se diseñó e implementó un sistema que permite la grabación de las llamadas por el protocolo RTP.
- ✓ Se probaron las funcionalidades del sistema a través del diseño de casos de prueba de la aplicación, las pruebas unitarias y las pruebas de validación con el cliente, permitiendo la validación del funcionamiento de los requisitos que fueron definidos.



### RECOMENDACIONES

Con el objetivo de incorporar mejoras al sistema se recomienda para próximas iteraciones:

- ✓ Realizar iteración de prueba para comprobar que las no conformidades fueron solucionadas.



### Referencias Bibliográficas

1. Mitel Networks Corporation. [www.mitel.com](http://www.mitel.com). [www.mitel.com](http://www.mitel.com). [En línea] 2010. [Citado el: 2 de Enero de 2012.] <http://www.mitel.com/PortalController?country=US>.
2. Internet Engineering Task Force. RTP: A Transport Protocol for Real-Time Applications. Columbia University, 2003.
3. NCH **Software**. [www.nch.com.au](http://www.nch.com.au). [www.nch.com.au](http://www.nch.com.au). [En línea] 2011. [Citado el: 9 de Octubre de 2011.] <http://www.nch.com.au/vrs/es/index.html>.
4. **Datavoice**. [www.datavoices.com](http://www.datavoices.com). [www.datavoices.com](http://www.datavoices.com). [En línea] 2011. [Citado el: 9 de Octubre de 2011.] <http://www.datavoices.com/Grabacion.htm>.
5. **Pingram Software**. [www.voicecallcentral.com](http://www.voicecallcentral.com). [www.voicecallcentral.com](http://www.voicecallcentral.com). [En línea] 1999. [Citado el: 9 de Octubre de 2011.] <http://www.voicecallcentral.com/es/callcorder.htm>.
6. **Nuxiba Technologies**. [www.nuxiba.com](http://www.nuxiba.com). [www.nuxiba.com](http://www.nuxiba.com). [En línea] 2010. [Citado el: 9 de Octubre de 2011.] [http://www.nuxiba.com/es/grabacion\\_llamadas.html](http://www.nuxiba.com/es/grabacion_llamadas.html).
7. **WOLTERS KLUWER ESPAÑA, S.A.** [www.telesoftware.es](http://www.telesoftware.es). [www.telesoftware.es](http://www.telesoftware.es). [En línea] 2010. [Citado el: 9 de Octubre de 2011.] <http://www.telesoftware.es/Productos.asp>.
8. **EuropeSIP**. [www.europesip.com](http://www.europesip.com). [www.europesip.com](http://www.europesip.com). [En línea] 2011. [Citado el: 9 de Octubre de 2011.] <http://www.europesip.com/index.php/es/portfolio-soluciones-software/oreka>.
9. **Wireshark Foundation**. [www.wireshark.org](http://www.wireshark.org). [En línea] 2008. [Citado el: 2 de Enero de 2012.] <http://www.wireshark.org/>.
10. **Digium, Inc.** <http://www.asterisk.org/>. <http://www.asterisk.org/>. [En línea] 2010. [Citado el: 2 de Enero de 2012.] <http://www.asterisk.org/>.
11. **Digium, Inc.** [www.Asterisk-ES.org](http://www.Asterisk-ES.org). [www.Asterisk-ES.org](http://www.Asterisk-ES.org). [En línea] 2011. [Citado el: 2 de Enero de 2012.] <http://comunidad.asterisk-es.org/index.php?title=Asterisk>.
12. **Beck, Kent**. *Extreme Programming Explained*. s.l. : Addison Wesley The XP Series, 2000.
13. **Weske, Mathias**. *Business Process Management. Concepts, Languages, Architectures*. Potsdam : Springer, 2007. ISBN 978-3-540-73521-2.
14. Booch, Grady, Rumbaugh, James and Jacobson, Ivar. *El Lenguaje Unificado de Modelado*. Madrid: Addison Wesley, 2005.
15. **Visual Paradigm International**. [www.visual-paradigm.com](http://www.visual-paradigm.com). [www.visual-paradigm.com](http://www.visual-paradigm.com). [En línea] 26 de Noviembre de 2011. [Citado el: 2 de Diciembre de 2011.] <http://www.visual-paradigm.com/>.



16. **Oracle.** www.java.com. *www.java.com*. [En línea] 2010. [Citado el: 18 de Noviembre de 2011.] <http://java.com/es/about/>.
17. **Python Software Foundation.** www.python.org. *www.python.org*. [En línea] 1990. [Citado el: 19 de Noviembre de 2011.] <http://python.org>.
18. **Oracle Corporation.** www.neatbeans.org. *www.neatbeans.org*. [En línea] 2011. [Citado el: 19 de Noviembre de 2011.] <http://neatbeans.org/>.
19. **Eclipse Foundation.** plataformaeclipse. *plataformaeclipse.com*. [En línea] 2004. [Citado el: 19 de Diciembre de 2011.] <http://plataformaeclipse.com>.
20. **Django Software Foundation.** www.djangoproject.com. *www.djangoproject.com*. [En línea] 2005. [Citado el: 25 de Noviembre de 2011.] <https://www.djangoproject.com/>.
21. **garbagecollector.** Sistema Gestor de base de datos SGBD. [En línea] 01 de Noviembre de 2004. [Citado el: 25 de Noviembre de 2011.] [http://www.error500.net/garbagecollector/archives/categorias/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_datos\\_sqbd.php](http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sqbd.php).
22. **PostgreSQL Global Development Group.** www.postgresql.org. *www.postgresql.org*. [En línea] 1996. [Citado el: 2 de Diciembre de 2011.] <http://www.postgresql.org>.
23. <http://www.ecured.cu>. *http://www.ecured.cu*. [En línea] 2011. [Citado el: 2 de Diciembre de 2011.] <http://www.ecured.cu/index.php/SQL>.
24. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque Práctico*. Bogotá : McGraw-Hil, 2007. ISBN 978-0-07-337 597-7.
25. **Synergix.** Wordpress. [En línea] [Citado el: 9 de Febrero de 2012.] Disponible en: [<http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>].
26. **Acebal, César F. y Cueva Lovelle, Juan M.** *Extreme Programming (XP): un nuevo método de desarrollo de software*. Oviedo : Novatica, 2002.
27. **Steinberg, Daniel H; Palmer, Daniel W.** *Extreme Software Engineering*, Pearson Education, Inc., ISBN 0-13-047381-2
28. **Camacho, Erika; Cardeso ,Fabio; Nuñez, Gabriel.** *Arquitecturas de Software*. Abril, 2004.
29. **Kruchten, Philippe.** *Architectural Blueprints--The 4+1 View Model of Software Architecture*. New York: IEEE Software, 1995.
30. Bertrand Meyer. *Construcción de software orientado a objetos (Segunda edición)*. Prentice Hall.



31. Jacobson, Ivar. 2000. El Proceso Unificado de Desarrollo de Software. 2000.
32. **Larman, Craig. 2004.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* La Habana: Feliz Varela, 2004. pág. 189.
33. **Alexander, Christopher.** *A Pattern Language: Towns, Buildings, Construction.*
34. **.Escribano, G. F.** "eXtreme Programming / Programación Extrema." 2002.
35. **Marcos, Mari Carmen and Palma, María del Valle.** *Pautas para el diseño y la evaluación de interfaces de usuario.* Barcelona: Roviera, 2004.
36. **Crispin, Lisa y House, Tip. 2002.** *Testing Extreme Programming.* s.l.: Addison-Wesley, 2002. 0321113551.
37. **Pérez, Isaías Carrillo.** *METODOLOGIA DE DESARROLLO DE SOFTWARE.* s.l. : 2, 2005