



# BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad CIENCIAS DE LA COMPUTACIÓN

## TESIS PROFESIONAL

Para Obtener El Título De:

## ING. EN CIENCIAS DE LA COMPUTACIÓN

### Sistema de Facturación sobre una plataforma Asterisk (VoIP)

**Presenta: MISAEL OCTAVIO RODRÍGUEZ DE LA ROSA**

**Asesor: DR. MANUEL I. MARTIN ORTIZ**

**Coasesor: DR. IVO PINEDA TORRES**

Junio del 2009  
Puebla, Puebla.



# RESUMEN

La tecnología de VoIP se ha consolidado enormemente en ámbitos empresariales, institucionales y académicos gracias a varias plataformas como lo es Asterisk. VoIP no es solo acerca de hacer y recibir llamadas; es una forma totalmente nueva de comunicarse. Con *VoIP*, una compañía puede disfrutar del incremento de productividad y satisfacción de consumo.

Asterisk es una plataforma telefónica de código abierto (GPL) que pretende revolucionar el mundo de las comunicaciones IP frente las tradicionales soluciones de grandes corporaciones como Cisco, Nortel, Ericsson, Siemens, etc. También Asterisk unifica muchos de los servicios que, tradicionalmente, se ofrecían por separado y/o en sistemas no integrados.

Desde que se empezó a usar la tecnología de VoIP sobre la facultad para poder ofrecer telefonía a departamentos que no poseían una extensión telefónica, se abrió la posibilidad de ampliar las extensiones telefónicas para la facultad. Como en cualquier servicio que se ofrezca se requiere un control administrativo para mantener un registro del uso del mismo. Para ello el sistema Asterisk se convierte en un factor esencial, ya que sobre esta plataforma se puede desarrollar e implementar un amplísimo abanico de aplicaciones para su administración.

Con este trabajo se pretende dar a conocer algunos fundamentos acerca de lo que es voz sobre IP, el sistema Asterisk y algunos sistemas IPBX comerciales. Además de cómo crear una aplicación de facturación para un sistema Asterisk, el cual permite obtener información de las llamadas registradas en Asterisk, crear facturas para los usuarios que usan este servicio y realizar procesos de actualización de datos de forma automatizada.



# Agradecimientos.

*A mis padres*

*Arturo y Ana por el apoyo que me han otorgado y enseñarme a seguir adelante.*

*A mis amigos de rancho*

*A todos ellos que entre desveladas, carcajadas y desmanes, hicieron que la vida no fuera de un solo color.*

*A mis amigos de uni*

*A ellos que me dieron su confianza y apoyo en estos años de carrera.*

*A mis Asesores.*

*Dr. Manuel y Dr. Ivo, que a pesar de que parecía fantasma, me dieron el apoyo para realizar este trabajo.*

*Al Jurado.*

*Al M.C. Esteban y M.C. Hilda, que es la primera vez que tengo oportunidad de conocerlos y me hubiera gustado tener clases con ellos.*

*Y en especial a Carmen*

*Por apoyarme y darme fuerzas para ser aun mejor, TE AMO.*



# Índice.

<b>Introducción</b>	<b>9</b>
<b>CAPÍTULO 1</b>	<b>11</b>
<b>1.1 Planteamiento del Problema</b>	<b>11</b>
<b>1.2 Objetivos.</b>	<b>11</b>
1.2.1 Objetivo general	11
1.2.2 Objetivos específicos:	11
<b>1.3 Solución propuesta.</b>	<b>12</b>
<b>CAPÍTULO 2. Conceptos de VoIP, Asterisk y Facturación</b>	<b>14</b>
<b>2.1 Introducción</b>	<b>14</b>
<b>2.2 VoIP</b>	<b>15</b>
2.2.1 Elementos de la Voz sobre IP	16
2.2.2 Codificadores de audio.	17
2.2.3 Protocolos de Voz sobre IP	19
2.2.3.1 H.323	19
2.2.3.2 Session Initiation Protocol (SIP)	19
2.2.3.3 MGCP-MEGACO	20
2.2.3.4 IAX	20
<b>2.3 ASTERISK</b>	<b>20</b>
2.3.1 Arquitectura de Asterisk	23
2.3.1.1 Aplicaciones	25
2.3.2 Alternativas	26
2.3.2.1 Introducción	27
2.3.2.2 Características básicas a considerar	27
2.3.2.2.1 Características para Enrutamiento de Llamadas	28
2.3.2.2.2 Características de Mensajería y de Gestión	28
2.3.2.3 Características Avanzadas.	29
2.3.2.3.1 Gestión	29
2.3.2.3.2 Programabilidad en el enrutamiento y Scheduling	29
2.3.2.3.3 Características para Usuarios Remotos.	30
2.3.2.3.4 Unificar / Integración de Sistemas de Oficina	30
2.3.2.4 Características Únicas	30
2.3.2.5 Algunas Alternativas	31
<b>2.4 Factura</b>	<b>35</b>
<b>2.5 E-Factura</b>	<b>36</b>
2.5.1 Beneficios	38
<b>2.6 Lenguajes usados.</b>	<b>39</b>
2.6.1 XML	39
2.6.2 XSLT	40

2.6.3 XSL-FO	41
2.6.4 SQL	42
2.6.5 Shell Script	43
2.6.6 PHP	43
2.6.7 Ajax	45
<b>CAPÍTULO 3. ANALISIS Y DISEÑO</b>	<b>47</b>
3.1 Introducción	47
3.2 Especificación de Requerimientos	47
3.3 Objetivos del sistema	50
3.4 Requisitos de almacenamiento de información	50
3.5 Diagramas de casos de uso	52
3.5.1 Definición de actores	53
3.5.2 Casos de uso del Sistema	54
3.5.3 Especificación de casos de uso	56
3.6 Diagrama de Clases	64
3.6.1 Diccionario de Datos	64
3.7 Escenarios del Sistema	68
3.8 Diagramas de Secuencia	69
<b>CAPÍTULO 4. IMPLEMENTACIÓN</b>	<b>72</b>
4.1 Introducción	72
4.2 Estructura de los directorios del proyecto.	72
4.3 Herramientas usadas.	73
4.4 Subsistema de Gestión.	74
4.5 Subsistema de Facturación	75
4.5.1 La factura (reporte)	79
4.6 Subsistema de Información Vía Web	82
4.7 Pruebas y resultados	88
<b>CAPÍTULO 5. CONCLUSIONES Y PERSPECTIVAS</b>	<b>90</b>
5.1 Conclusiones.	90
5.2 Limitaciones.	90
5.3 Perspectivas.	91
<b>BIBLIOGRAFÍA</b>	<b>92</b>
<b>APÉNDICE I. MIGRACIÓN E INSTALACIÓN</b>	<b>93</b>
<b>APÉNDICE II. MANUAL DE USUARIO</b>	<b>102</b>

## Introducción

En la evolución de los sistemas de comunicación como es la telefonía, fue necesario tener un registro de las llamadas que se realizaban. En sus inicios el número de personas que poseían un servicio telefónico era mínimo con respecto a la población de un sector; esto implicaba un fácil control de las llamadas que se hacían, por lo que el cobro a los usuarios era mas sencillo. Con el paso del tiempo, aumentó el número de usuarios y servicios telefónicos que se ofrecían, esto fomentó el automatizar el control de llamadas realizadas por los usuarios, para ayudar en el cobro de estos servicios.

La telefonía ha sufrido varios cambios, y esto es debido a que con la revolución digital, ha facilitado el realizar varias actividades con la ayuda de una computadora, así también el uso de Internet que aumenta las posibilidades de comunicación y servicios que se pueden ofrecer a través de ella, por lo que la telefonía considere al Internet como un medio con gran potencial para su uso. En específico la comunicación por voz que convencionalmente se hacía por las redes telefónicas pase a un ámbito digital, a través de las redes de computadoras locales, privadas, corporativas y públicas.

Con la digitalización de la señal de voz, el fuerte crecimiento de las redes IP y la aparición de protocolos de transmisión en tiempo real se ha creado un nuevo entorno para la transmisión de voz sobre IP (VoIP). Hasta ahora las empresas operaban con dos infraestructuras separadas para el transporte de voz y el de datos, cuya dificultad de gestión y mantenimiento va creciendo progresivamente.

Gracias a la tecnología de red LAN y los diferentes protocolos de VoIP es posible converger estas dos infraestructuras en una. De esta forma no sólo se consigue reducir costes, si no que se puede ofrecer nuevos servicios de valor añadido y funcionar independientemente del dispositivo de acceso utilizado (teléfono convencional, teléfono IP, softphone, etc.)

También cabe mencionar que el registro y control de los servicios telefónicos que ahora se ofrecen es más sencillo, además de que existen sistemas tarificadores que realizan estas actividades que antes se hacían de forma manual. El nivel y la estructura de las tarifas de telecomunicaciones tienen una importante función que cumplir, en la creación de fondos de generación interna, además para sufragar los gastos recurrentes de las entidades de telecomunicaciones. Las tarifas pueden promover también la utilización eficaz de la red y de los servicios, mejorar la prestación del servicio universal y tener efectos multiplicadores en el resto de la economía, puesto que las telecomunicaciones son a la vez un servicio comercial y un medio para facilitar el desarrollo económico.

Es por ello que este trabajo es realizado para complementar el servicio VoIP implementado recientemente en la facultad, el cual ayudará a mantener un registro de este. Además de proporcionar cierta información útil para la Administración y los usuarios de la misma.

Este trabajo está dividido en diferentes capítulos y se da una breve descripción de ellos.

Capítulo I: Se presenta una introducción del problema, así como los objetivos que se quieren lograr, además de una breve descripción de la solución propuesta.

Capítulo II: Se da una descripción acerca de Voz sobre IP, el sistema Asterisk como algunas soluciones diferentes de Asterisk, algunos conceptos con respecto a la facturación y facturación electrónica, además de dar a conocer algunos lenguajes que se usaron en el proyecto.

Capítulo III: Se presenta el planteamiento del problema, los requerimientos necesarios para su uso, los objetivos y el análisis de sus principales funciones, además del diseño del sistema, los diferentes diagramas de las funciones principales y tablas del sistema.

Capítulo IV: En este capítulo se explica la implementación, las ventanas del sistema y las herramientas empleadas para desarrollar el mismo.

Capítulo V: Conclusiones: limitaciones y perspectivas del sistema.

Cuenta con dos apéndices, Apéndice I “Instalación y Migración”, y Apéndice II “Manual de Usuario”.

# CAPITULO 1

## 1.1 Planteamiento del Problema

Ante los primeros esfuerzos para la implementación de un servicio de Voz sobre IP en la Universidad, específicamente en la FCC, y la reducción de costos que esta conlleva, es necesario tener un registro de este servicio. Este tipo de registro podría proporcionar información acerca de su uso, frecuencia con la que se llama, número de llamadas que se realizan, así como otros datos que, por ejemplo, podrían usarse para una auditoría si este se ameritara.

Por lo que un tarificador es un buen ejemplo para llevar este control, ya que además de proporcionar información de una venta o prestación de servicio, mantendría la información acerca del uso del servicio; esto implica llevar un control de las llamadas que se realizan sobre el sistema Asterisk.

En una forma simbólica, el sistema enviaría facturas, como si fuese en una empresa Telefónica, con el respectivo desglose del uso para cada usuario.

## 1.2 Objetivos.

### 1.2.1 Objetivo general

El Objetivo principal del proyecto es tener un control de llamadas, las cuales para cada usuario se le enviará su respectivo historial con el número de llamadas realizadas en el mes, además de que a la parte administrativa de la Facultad podrá consultar el número de llamadas realizadas en ese mismo mes o en meses anteriores.

### 1.2.2 Objetivos específicos:

- Llevar un control de llamadas en un sistema Asterisk
- Usar software libre en el registro y consulta de llamadas, como es MYSQL y PHP.

- Usar el servicio de VoIP implementado en la Facultad para la realización de este proyecto.
- Indagar sobre formas correctas de proporcionar el registro de llamadas y sus tarifas.
- De forma periódica realizar procesos de facturación y actualización de información de forma automatizada.
- Vía Web realizar consultas de las llamadas realizadas, tanto por los usuarios como por la administración.

### 1.3 Solución propuesta.

El sistema de VoIP Asterisk implementado ofrecía varias posibilidades para registrar las llamadas que se hacían, una de ellas que por defecto tiene es guardar esta información en un archivo de texto plano. Otra opción era por medio de una Base de Datos externa, la cual se tiene que configurar en Asterisk, para que almacene la información de las llamadas.

La opción de guardar la información en una base de datos externa, daba más posibilidades de usarse en distintas aplicaciones, como es la de tarificar, ya que Asterisk tiene como característica proporcionar información de billing (cobro), esto facilita el proceso de la información.

Además se optó usar los lenguajes XML, XSLT y XSL-FO, para realizar los documentos que proporcionarán la información de cobro (factura, véase figura 1.1). Y también los lenguajes PHP, SQL y AJAX para realizar las consultas a esta información a través de la Web (véase figura 1.2).

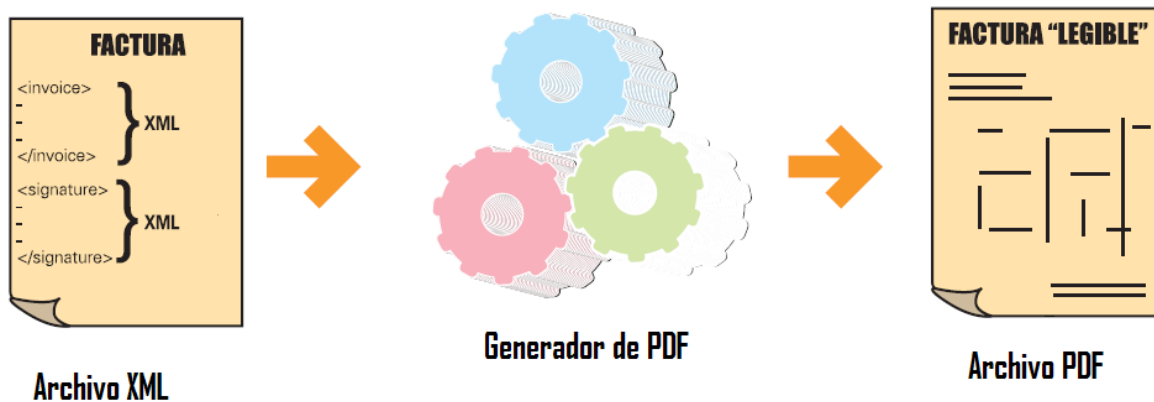


Figura 1.1 Ejemplo de esquema en la generación de una factura.

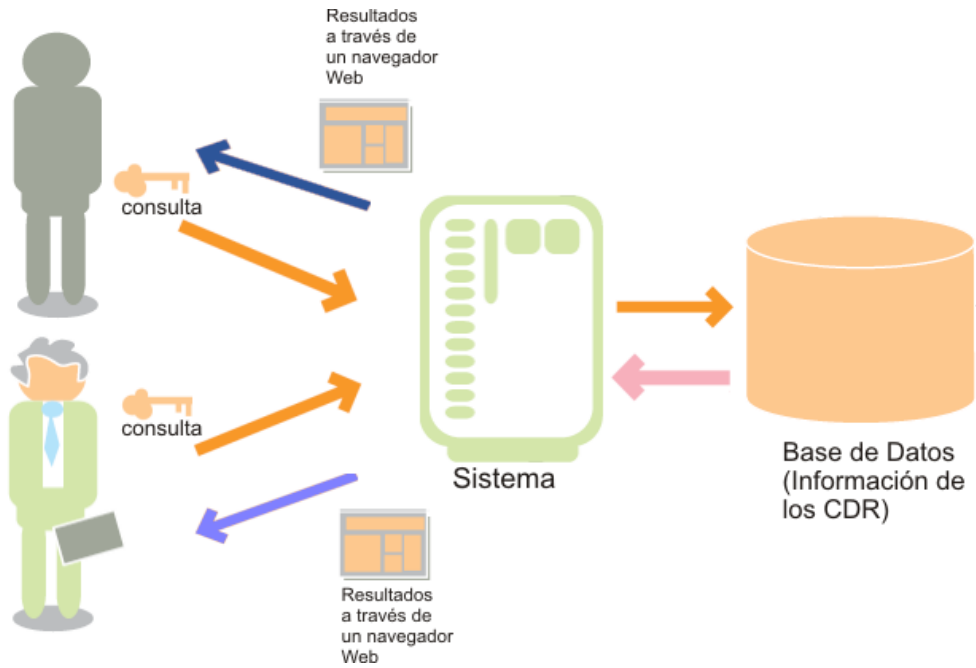


Figura 1.2 Ejemplo de esquema en la consulta vía Web del Sistema.

Y por último para automatizar el proceso, se utilizó Shell-Script y el uso del servicio *cron-tab*, que en la mayoría de distribuciones Linux esta presente.

## Conceptos de VoIP, Asterisk y Facturación

### 2.1 Introducción

Antes de que las redes digitales llegaran, cualquiera podría usar un plan de servicio telefónico, el cual funciona sobre la Red Telefónica Básica (RTB). Esos servicios de telefonía usan el tan conocido método circuit-switched, en la que los terminales telefónicos (teléfonos) se comunican con una central de conmutación a través de un solo canal compartido por la señal del micrófono y del auricular.

Cuando las redes digitales fueron introducidas, las compañías telefónicas buscaron la forma de usarlas por que estas proveían una forma más eficiente de la señal de transmisión. Como era de esperarse las compañías encontraron un problema ¿Cómo es que podían hacer que los teléfonos análogos trabajaran con una red digital? La respuesta fue el uso de un *códec*[1] que convertía la señal análoga a digital. ¿Pero dónde podría tomar lugar la conversión? En los últimos años de la revolución digital, esta conversión tomó lugar en las instalaciones de la compañía de teléfonos, la cual le permitía usar el existente cableado entre las instalaciones y el lugar del usuario.

La mayor parte de Redes Telefónicas mantienen una combinación de analógico y digital. Los usuarios toman un teléfono, el cual convierte el audio a señales analógicas. Esas señales son llevadas sobre la red telefónica hacia las instalaciones del *carrier*[2], donde ellos la convierten a una señal digital. Las señales son reenviadas al destino que lo recibirá. Antes que las señales sean recibidas por el último equipo del *carrier*, son convertidas a análogas.

Hoy en día los usuarios pueden tener entre un teléfono análogo o digital, dependiendo de la locación y el equipo del usuario.

---

1 **Códec** es una abreviatura de Compresor-Decompresor  
2 **Carrier** se refiere al proveedor del servicio telefónico.

En este capítulo enfocaremos los diferentes aspectos de VoIP, Asterisk y la Facturación, como se describen en las siguientes secciones. Cada una pretende un cambio con respecto al concepto de Telefonía y los formatos de Facturación.

## 2.2 VoIP

*VoIP* es el nombre de una tecnología de comunicaciones que está cambiando el significado de la frase *llamada telefónica*. *VoIP* se entiende por voz sobre el protocolo de internet, esto significa *voz transmitida sobre la red de computadoras*.

El protocolo de Internet (IP) es soportado por todas las redes: corporativas, privadas, públicas y también redes inalámbricas. *VoIP* funciona sobre cualquier tipo de red. VoIP es referida a la Telefonía IP (*IPT* por sus siglas en inglés) por que este usa protocolos de internet que hacen posible la comunicación de voz. Técnicamente, *IPT* se refiere a las llamadas telefónicas realizadas sobre las organizaciones de redes locales (LAN) como un edificio, una red de un campus o incluso una red local en una casa. Cuando la *IPT* atraviesa desde una LAN a una WAN u otro tipo de red externa, incluyendo otras LANs operando por el mismo *carrier* en distintas locaciones o el internet, este se vuelve *VoIP*.

*VoIP* también hace posible otros servicios que los viejos sistemas telefónicos no pueden proveer. El servicio telefónico *VoIP* es interoperable, lo que significa que trabaja sobre cualquier tipo de redes. Estos son altamente portables, lo cual significa que trabaja con cualquier dispositivo IP como es un teléfono IP, una computadora, o también un asistente digital Personal (PDA).

VoIP no es solo acerca de hacer y recibir llamadas; es una forma totalmente nueva de comunicarse. Con *VoIP*, una compañía puede disfrutar del incremento de productividad y satisfacción de consumo. Esas mejoras son típicamente realizadas a través de la flexibilidad que ofrece, por mayores características de una llamada. Una de las mejoras como es los mensajes de voz y la transferencia de llamadas, han estado en las telefonías comunes alrededor del mundo. Por otro lado, integrar aplicaciones de datos, voz y video para funcionar sobre una simple red y trabajar con teléfonos inalámbricos es una de las más recientes innovaciones hechas posible por la telefonía IP.

### 2.2.1 Elementos de la Voz sobre IP

El modelo de Voz sobre IP (véase fig. 2.1) está formado por tres principales elementos:

- **El cliente.** Este elemento establece y termina las llamadas de voz. Codifica, empaqueta y transmite la información de salida generada por el micrófono del usuario. Asimismo, recibe, decodifica y reproduce la información de voz de entrada a través de los altavoces o audífonos del usuario. Cabe destacar que el elemento cliente se presenta en dos formas básicas:
  - **Hardware:** un teléfono IP es un terminal que tiene soporte VoIP nativo y puede conectarse directamente a una red IP.
  - **Software:** un softphone es una aplicación audio ejecutable desde PC que se comunica con las *PABX*[3] a través de la LAN. Para interactuar con el usuario se basa en la utilización de un micrófono y altavoz o mediante un teléfono USB.
- **Servidores.** El segundo elemento de la Voz sobre IP está basado en servidores, los cuales manejan un amplio rango de operaciones complejas de bases de datos, tanto en tiempo real como fuera de él. Estas operaciones incluyen validación de usuarios, tasación, contabilidad, tarificación, recolección, distribución de utilidades, enrutamiento, administración general del servicio, carga de clientes, control del servicio, registro de usuarios y servicios de directorio entre otros. Este puede adoptar diferentes nombres dependiendo del protocolo de señalización utilizado. Así en un sistema basado en el protocolo H.323, el servidor es conocido como **Gatekeeper**; en un sistema SIP, **servidor SIP**; y en un sistema basado en MGCP o MEGACO, **Call Agent** (Agente de llamadas).
- **Gateways.** El tercer elemento lo conforman los gateways de Voz sobre IP, los cuales proporcionan un puente de comunicación entre los usuarios. La función principal de un gateway es proveer las interfaces con la telefonía tradicional apropiada, funcionando como una plataforma para los clientes virtuales.
- Estos equipos también juegan un papel importante en la seguridad de acceso, la contabilidad, el control de calidad del servicio (QoS; Quality of Service) y en el mejoramiento del mismo.
- **Red IP:** provee conectividad entre todos los terminales. La red IP puede ser una red IP privada, una Intranet o Internet.

---

3 **PBX** o **PABX** (siglas en inglés de *Private Branch Exchange* y *Private Automatic Branch Exchange* para PABX) cuya traducción al español sería *Central secundaria privada automática*

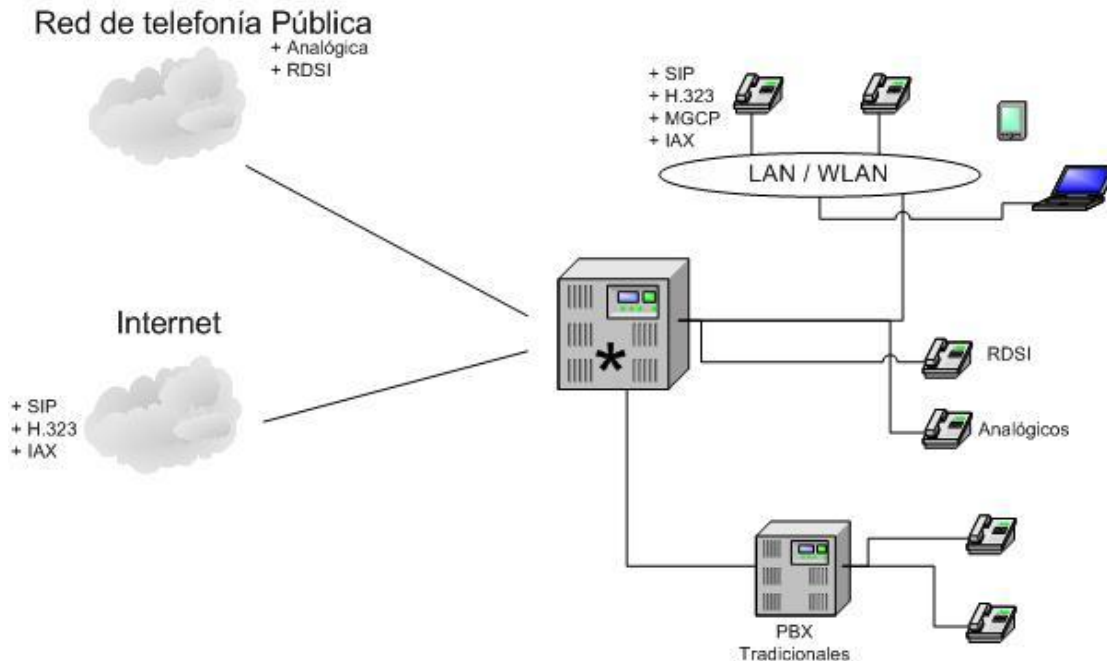


Figura 2.1 Estructura de una red VoIP

### 2.2.2 Codificadores de audio.

La señal de audio ha de ser digitalizada, comprimida y codificada antes de ser transmitida por la red IP. Para ello se utilizan algoritmos matemáticos implementados en software llamados códecs (acrónimo de codificador-descodificador aunque actualmente se le atribuye también las funciones de compresor-descompresor). Existen diferentes modelos de códecs de audio utilizados en VoIP, y dependiendo del algoritmo escogido en la transmisión, variará la calidad de la voz, el ancho de banda necesario y la carga computacional. El objetivo principal de esta tecnología es encontrar un equilibrio entre eficiencia y calidad de la voz (Véase tabla 2.1). Aunque el sistema auditivo humano es capaz de captar las frecuencias comprendidas entre 20 Hz y 20 kHz, la gran mayoría de códecs procesan aquella información dentro de la banda de 400 Hz – 3,5 kHz ya que a la hora de reconstruir la señal, esta sigue siendo inteligible. A continuación se enumeran y describen, entre otros, los códecs más utilizados en VoIP:

- **G.711:** principal códec de la PSTN<sup>4</sup> estandarizado por la ITU (*International Telecommunication Union*) en 1972. Este estándar muestrea a una frecuencia de 8

<sup>4</sup> Siglas para **Public Switched Telephone Network (PSTN)**, que en español se refiere a la Red Telefónica Básica.

kHz y utiliza PCM (*Pulse Code Modulation*) para comprimir, descomprimir, codificar y decodificar. Existen dos modelos:

- **μ-law:** codifica cada 14 muestras en palabras de 8 bits. Usado en EE.UU. y Japón.
- **A-law:** codifica cada 13 muestras en palabras de 8 bits. Usado en el resto del mundo, incluido Europa.

Al entregar ambas palabras de 8 bits se requiere un ancho de banda de 64 kbps. G.711 es un algoritmo más simple y de menor carga computacional y es la base del resto de estándares.

- **G.726:** este estándar de la ITU, también conocido como ADPCM (*Adaptive Differential Pulse Code Modulation*), sustituyó al obsoleto estándar G.721 en 1990. Permite trabajar a velocidades de 16 kbps, 24 kbps y 32 kbps. La gran ventaja de este códec es la disminución de ancho de banda requerido sin aumentar en gran medida la carga computacional.
- **G.723.1:** este algoritmo, estandarizado en 1995 por la ITU, puede operar a 6.3 kbps o 5.3 kbps. Si es utilizado en una aplicación comercial es necesario pagar una licencia.
- **G.729A:** códec desarrollado por *France Telecom, Mitsubishi Electric Corporation, Nippon Telegraph and Telephone Corporation (NTT)* y la Universidad de Sherbrooke. Requiere 8 kbps de ancho de banda. La carga computacional de este algoritmo es elevada y también es necesaria una licencia para su uso comercial. Actualmente la propiedad intelectual es de la empresa SIPRO.
- **GSM (*Global System Mobile*):** estándar que opera a 13 kbps con una carga de CPU aceptable. No requiere el pago de una licencia.
- **iLBC (*Internet Low Bit rate Codec*):** complejo algoritmo libre desarrollado por *Global IP Sound (GIPS)* que ofrece una buena relación ancho de banda – calidad de voz, a cambio de una mayor carga computacional. Es definido en los RFCs 3951 y 3952 de la IETF (*Internet Engineering Task Force*). Todo y ser el códec usado por *Skype* o *Google Talk* su uso no ha sido extendido en teléfonos IP. iLBC opera a 13.3 kbps y 15.2 kbps.
- **Speex:** software libre creado por *Xiph.Org Foundation* que implementa un algoritmo capaz de variar la velocidad de transmisión dependiendo de las condiciones actuales de la red (VBR: *Variable Bit Rate*). El ancho de banda puede variar desde 2.15 a 22.4 kbps.
- **MP3 (*Moving Picture Experts Group Audio Layer 3 Encoding Standard*):** es un códec de audio y está optimizado para música y no para telefonía. Este popular

códec de la ISO (*International Standard Organization*) es utilizado por los teléfonos IP principalmente para ofrecer servicios de música en espera.

Códec	Estandarizado Por	Ancho de banda [kbps]	Retardo [ms]	Patente
<b>G.711</b>	ITU-T	64	20-30	No
<b>G.726</b>	ITU-T	16, 24, 32	20-30	No
<b>G.723.1</b>	ITU-T	6.3, 5.3	37.5	Si
<b>G.729A</b>	SIPRO	8	15	Si
<b>GSM</b>	GSM group	13	20	No
<b>iLBC</b>	IETF	13.3, 15.2	20-30	No
<b>Speex</b>	<i>Xiph.Org Found.</i>	2.15-22.4	30	No

**Tabla 2.1 Resumen comparativo de los distintos códecs utilizados en VoIP**

### 2.2.3 Protocolos de Voz sobre IP

Hoy en día, existen varios protocolos para transmitir voz sobre IP, estos definen la manera en que los dispositivos de este tipo deben establecer comunicación entre sí, además de incluir especificaciones para códecs (codificador-decodificador) de audio para convertir una señal auditiva a una digitalizada compresada y viceversa.

#### 2.2.3.1 H.323

H.323 es el estándar creado por la Unión Internacional de Telecomunicaciones (ITU) que se compone por un protocolo sumamente complejo y extenso, el cual además de incluir la voz sobre IP, ofrece especificaciones para vídeo-conferencias y aplicaciones en tiempo real, entre otras variantes.

#### 2.2.3.2 Session Initiation Protocol (SIP)

Session Initiation Protocol (SIP) fue desarrollado por la IETF (Internet Engineering Task Force) específicamente para telefonía IP, que a su vez toma ventaja de otros protocolos existentes para manejar parte del proceso de conversión, situación que no se aplica en H.323 ya que define sus propios protocolos bases.

### 2.2.3.3 MGCP-MEGACO

*Media Gateway Control Protocol* (MGCP) es otro estándar de señalización para VoIP desarrollado por la IETF. MGCP está basado en un modelo maestro/esclavo donde el *Call Agent* (servidor) es el encargado de controlar al gateway. De esta forma se consigue separar la señalización de la transmisión de la información, simplificando la integración con el protocolo SS7.

Esta importante ventaja propició la colaboración conjunta entre el IETF y la ITU para el desarrollo de una nueva especificación basada en MGCP que fuera complementaria a SIP y H.323. El resultado fue MEGACO aunque la ITU se refiere a este protocolo como H.248. En definitiva, SIP y H.323 se utiliza para la señalización en los extremos, mientras que MEGACO es óptimo para los grandes operadores de telefonía

### 2.2.3.4 IAX

*Inter-Asterisk eXchange protocol* (IAX) fue desarrollado por *Digium* para la comunicación entre centrales telefónicas basadas en *Asterisk* aunque actualmente se ha implementado clientes que también soportan este protocolo.

El principal objetivo de IAX es minimizar el ancho de banda utilizado en la transmisión de voz y vídeo a través de la red IP y proveer un soporte nativo para ser transparente a los NATs (*Network Address Translation*). La estructura básica de IAX se fundamenta en la multiplexación de la señalización y del flujo de datos sobre un simple puerto UDP, generalmente el 4569.

El protocolo original ha quedado obsoleto en favor de su segunda versión conocida como IAX2. Se caracteriza por ser robusto y simple en comparación con otros protocolos. Permite manejar una gran cantidad de *códecs* y transportar cualquier tipo de datos.

## 2.3 ASTERISK

*Asterisk* es una aplicación con licencia GPL (código abierto) capaz de simular las funciones de una IPBX (*Internet Protocol/ Private Branch Exchange*). Este proyecto fue desarrollado por el ingeniero *Mark Spencer*, miembro fundador de la compañía *Digium*, principal desarrolladora de *Asterisk*.

*Asterisk* pese a ser una aplicación software ofrece las mismas características y servicios que los caros sistemas propietarios PBX como puede ser el buzón de voz, salas de conferencia o música en espera entre otros. Son muchas las ventajas que ofrece *Asterisk* respecto a las centrales telefónicas hardware:

- Reducción de costes y no sólo por el hecho de integrar voz y datos bajo una misma infraestructura, sino el hecho de que *Asterisk* sea una aplicación de código abierto evitando tener que pagar grandes cantidades por licencias.
- Facilita la integración y desarrollo de nuevos servicios de valor añadido.
- Compatibilidad con un gran número de protocolos VoIP y códecs.
- Es posible conectar *Asterisk* con otras centrales telefónicas, lo que le convierte en una solución flexible para futuros redimensionamientos.
- Existe un gran número de empresas y comunidades interesadas en el desarrollo de *Asterisk* que generan nuevas actualizaciones periódicamente. Debido a este gran interés existe una gran cantidad de información disponible.

*Asterisk* está formado por un núcleo principal encargado de gestionar todo el sistema PBX. Sus funcionales principales son:

- Interconectar de forma automática cada llamada ente los usuarios participantes teniendo en cuenta el tipo de protocolo utilizado por cada terminal.
- Lanzar los servicios de valor añadido cuando sean requeridos.
- Traducir y adaptar los códecs a cada terminal involucrado en la comunicación.
- Gestionar el sistema para que funcione de forma óptima en diversas condiciones de carga.

En la tabla siguiente se muestran algunas funcionalidades básicas PBX y características mas avanzadas de *Asterisk*.

<b>Asterisk</b>	
<b>Características de Llamadas</b>	<b>Integración de La Telefonía con la Computadora</b>
<ul style="list-style-type: none"> <li>• ADSI en el menú de pantalla</li> <li>• receptor de alarma</li> <li>• Añade mensajes</li> <li>• Asistente automatizado</li> <li>• Autenticación</li> <li>• listas negras</li> </ul>	<ul style="list-style-type: none"> <li>• AGI (PORTAL DEL INTERFAZ ASTERISK)</li> <li>• Encargado de llamadas graficas</li> <li>• Canalizado de llamadas salientes</li> <li>• TCP/IP manejo de interfaz</li> <li>• Marcado Predictivo</li> </ul>
	<b>Scalability</b>

<ul style="list-style-type: none"> <li>• Grabado de llamadas detallado</li> <li>• Llamada en ocupado</li> <li>• Llamada entrante en no responder</li> <li>• Llamada entrante variable</li> <li>• Monitoreo de llamadas</li> <li>• Call Parking</li> <li>• Llamadas en espera</li> <li>• Grabación de llamadas</li> </ul>	<ul style="list-style-type: none"> <li>• TDMoE (DIVISION DE TIEMPO MULTIPLE SOBRE INTERNET)</li> <li>• Permite la conexión directa con el PBX del Asterisk</li> <li>• Estado latente cero</li> <li>• Utiliza un hardware de Internet material</li> <li>• VoIP</li> <li>• Permite la integración de instalaciones físicamente separadas</li> <li>• Utiliza conexiones de dato comúnmente desplegados</li> <li>• Permite un plan unificado a través de múltiples oficinas.</li> </ul>
<ul style="list-style-type: none"> <li>• ☑ Recuperación de llamadas</li> <li>• ☑ ☑ Guía de llamadas (DID y ANI)</li> <li>• Call snooping</li> <li>• Transferencia de llamadas</li> <li>• Llamadas en espera</li> <li>• Identificación de usuarios</li> <li>• Bloque de identificaron de usuarios</li> <li>• Identificación de usuarios en llamadas de espera</li> <li>• Tarjetas de llamadas</li> <li>• Conferencias</li> <li>• Recuperación de base de datos almacenados</li> <li>• Integración de base de datos</li> <li>• Dial por nombre</li> <li>• Acceso directo al sistema interno</li> </ul>	<p style="text-align: center;"><b>CODECS</b></p> <ul style="list-style-type: none"> <li>• ADPCM</li> <li>• G.711</li> <li>• G.723.1 (pasar a través)</li> <li>• G.726</li> <li>• G.729</li> <li>• Gsm</li> <li>• Ilbc</li> <li>• Linear</li> <li>• lpc-10</li> <li>• Speex</li> </ul>
<ul style="list-style-type: none"> <li>• Anillo distintivo</li> <li>• No molestar</li> <li>• e911</li> <li>• ENUM</li> <li>• Fax transmitidos y recibidos</li> <li>• Lógica flexible de la extensión</li> <li>• Lista de directorio interactivo</li> <li>• Respuesta de voz interactivo</li> <li>• Agentes de llamada local y lejana</li> </ul>	<p style="text-align: center;"><b>Protocolos :</b></p> <ul style="list-style-type: none"> <li>• IAX (intercambio del Asterisk)</li> <li>• H.323</li> <li>• SIP (sesión de inicio del protocolo)</li> <li>• MGCP (Media Gateway Control Protocol)</li> <li>• SCCP (Cisco Skinny)</li> </ul>
	<p style="text-align: center;"><b>Interoperabilidad de Telefonía Tradicional</b></p> <ul style="list-style-type: none"> <li>• E y M</li> <li>• E y M wink</li> <li>• Feature group D</li> <li>• FXS</li> <li>• FXO</li> </ul>

<ul style="list-style-type: none"> <li>• Macros</li> <li>• Creación de música</li> <li>• Transferencia de música</li> <li>• Sistema básico de mp3</li> <li>• control de volumen</li> </ul>	<ul style="list-style-type: none"> <li>• GR – 303</li> <li>• Loop star</li> <li>• Ground star</li> </ul>
<ul style="list-style-type: none"> <li>• Privacidad</li> <li>• Establecimiento de protocolo abierto (OSP)</li> <li>• Paginación arriba</li> <li>• Conversión de protocolo</li> <li>• Recepción de llamadas lejanas</li> <li>• Apoyo a oficinas de lejos</li> <li>• Extensiones roaming</li> <li>• Mensajes SMS</li> <li>• Acceso a los medios afluyentes</li> <li>• Detección de conversaciones</li> <li>• Hora y fecha</li> <li>• Transcodificación</li> <li>• Trunking</li> <li>• entradas al VoIP</li> <li>• Voicemail</li> <li>• indicador visual para los mensajes en espera</li> <li>• voicemails a emails</li> <li>• grupos de voicemail</li> <li>• interfaces de web voicemail</li> <li>• zapateller</li> </ul>	<ul style="list-style-type: none"> <li>• MF y DTMF support</li> <li>• RBS</li> </ul>
	<b>Protocolos PRI</b>
	<ul style="list-style-type: none"> <li>• 4ESS</li> <li>• BRI (ISDN4LINUX)</li> <li>• DSM100</li> <li>• euroISDN</li> <li>• Lucent 5E</li> <li>• NATIONAL ISDN2</li> <li>• NFAS</li> </ul>

**Tabla 2.2 Características de Asterisk**

En la figura 2.2 se muestra un esquema de un Sistema basado en Asterisk, así como la manera en que interactúan los protocolos, las interfaces de hardware y Asterisk.

### 2.3.1 Arquitectura de Asterisk

Asterisk está cuidadosamente desarrollado para máxima flexibilidad. APIs específicos son definidos en un sistema central PBX. Este centro avanzado maneja interconexión interna

del PBX, abstraídos limpiamente por protocolos específicos, Códecs, e interfaces de hardware de aplicaciones de telefonía (véase fig. 2.3). Esto le permite al Asterisk utilizar cualquier hardware conveniente y tecnología disponible, ahora ó en el futuro para realizar sus funciones esenciales, conectando hardware y aplicaciones.

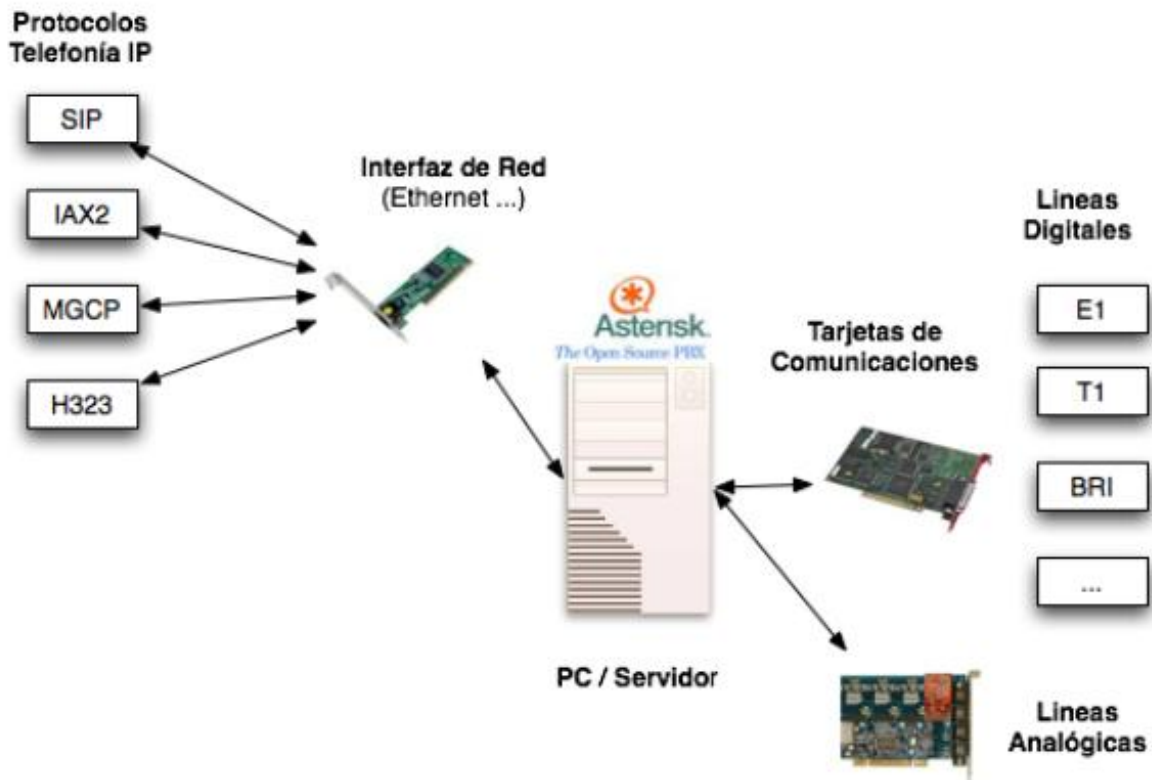


Figura 2.2 Esquema conceptual de Asterisk

**Asterisk maneja lo siguiente internamente:**

- **PBX SWITCHING:** La esencia del Asterisk, por supuesto es un sistema de conmutación de intercambio de rama privada (PBX), conectando llamadas entre varios usuarios y tareas automatizadas. La base de conmutación conecta a los usuarios usando software y hardware como interface.
- **LANZADOR DE APLICACIONES:** Lanza aplicaciones que mejoran servicios para usos tales como, voicemail, file playback y lista de directorio.
- **TRADUCTOR DE CODECS:** usa módulos de Códecs para codificar y decodificar varios formatos de comprensión de audio usado en la industria de la telefonía. Un

gran número de Códecs están disponibles para satisfacer necesidades y llegar al mejor balance entre la calidad del audio.

- **ORGANIZADOR Y MANEJADOR:** Maneja la organización de tareas de bajo nivel y sistemas de manejo para un óptimo performance bajo cualquier condición de carga.

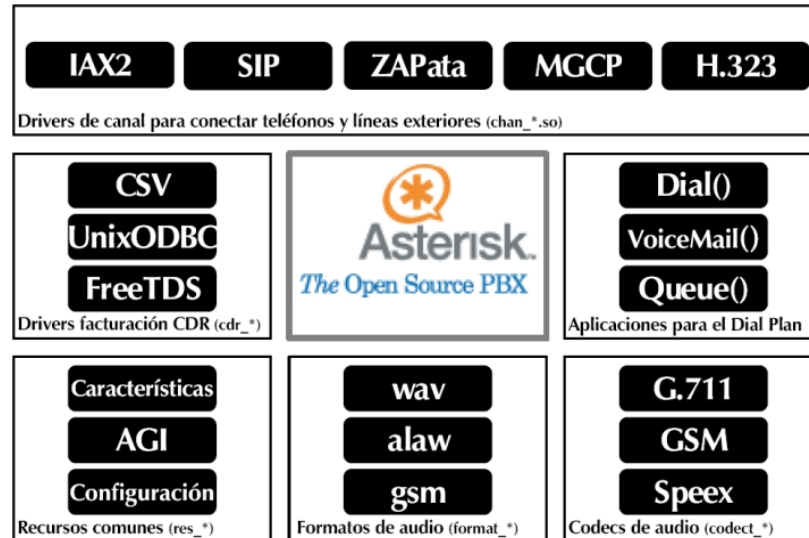


Figura 2.3 Arquitectura base de Asterisk.

### 2.3.1.1 Aplicaciones

Para conectar las llamadas de entrada con las llamadas de salida u otros usuarios de *Asterisk* son usadas diversas aplicaciones como es Dial, por ejemplo. La mayor parte de las funcionalidades de *Asterisk* son creadas en forma de aplicaciones como son el VoiceMail (buzón de voz), Meetme (conferencia), entre otras.

*Asterisk* incorpora una consola CLI (*Command Line Interface*) que permite controlar y comunicarse con el demonio y un par de interfaces (AGI: *Asterisk Gateway Interface* y AMI: *Asterisk Management Interface*) que permiten añadir aplicaciones externas programadas en C, C++, Java, Perl o en cualquier otro lenguaje de programación soportado por *Linux*.

Se pueden ver las aplicaciones disponibles en *Asterisk* usando el comando "core show applications" en la interface de línea de comando del *Asterisk*. Más allá de las aplicaciones en la versión central existen aplicaciones que pueden ser adicionadas a partir de archivos asterisk-addons y de terceros.

Asterisk puede actuar de servidor y Gateway (véase figura 2.4). Para conectar esta IPBX a la red PSTN, a la RDSI[5] o bien conectarle teléfonos analógicos; no basta con un simple módem sino que son necesarias unas tarjetas telefónicas con puertos FXO[6], BRI/PRI o FXS[7] respectivamente. La comunicación entre el servidor y las tarjetas se efectúa a través de otro módulo externo, llamado *zaptel*, que actúa como controlador de la tarjeta. Además si el acceso al exterior se realiza mediante puertos primarios de una red RDSI se necesita un segundo módulo complementario llamado *libpri*.

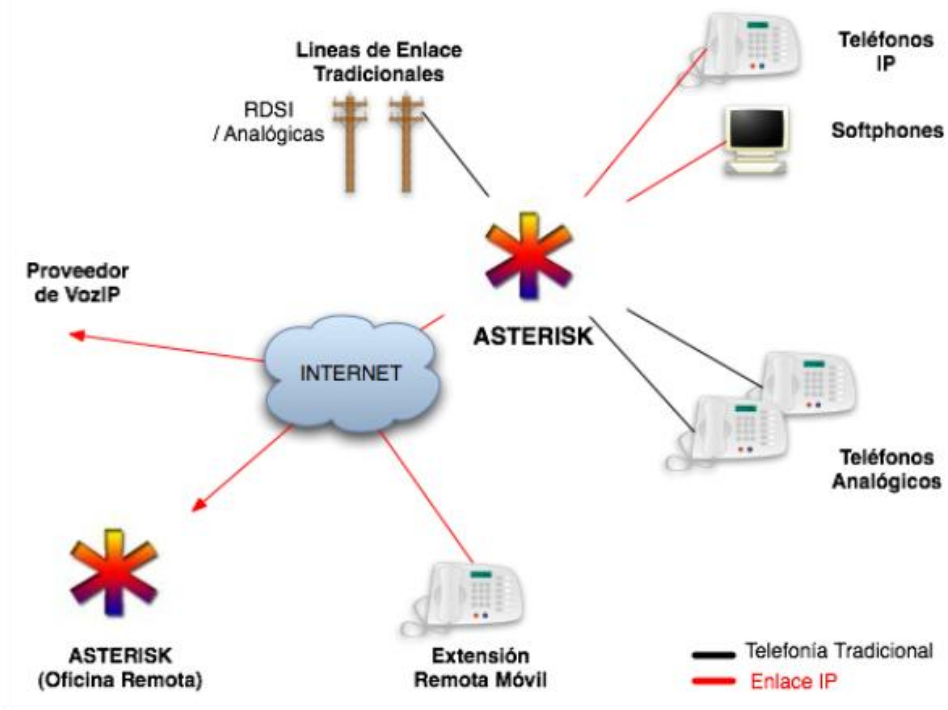


Figura 2.4 Asterisk como una IPBX

### 2.3.2 Alternativas

Asterisk es una de tantas opciones para una solución VoIP, por lo que a continuación se dará una breve explicación de algunas características de IPBX y mostraremos una tabla de alternativas para VoIP.

---

5 Red Digital de Servicios Integrados (RDSI o ISDN en inglés).

6 FXO (*Foreign Exchange Office*, en inglés) es un dispositivo de computador que permite conectar éste a la Red Telefónica Básica.

7 FXS (sigla de **Foreign Exchange Station**) es el conector en una central telefónica o en la pared de nuestro hogar, que permite conectar un teléfono analógico estándar.

### 2.3.2.1 Introducción

Los beneficios que puede proporcionar un IPBX sobre un sistema telefónico tradicional o PBX incluyen lo siguiente:

- Una infraestructura clara y simple
- Costos de operación bajos
- Simplicidad en equipamiento y manutención
- Comunicaciones unificadas
- Mejora de escalabilidad y crecimiento
- Mejora de características para operaciones de negocio
- Mejora de características y capacidades tecnológicas.

Beneficios Específicos.

	Telefonía Tradicional	VoIP
Llamadas en Conferencia	Equipamiento especial es requerido para más de tres personas	Fácilmente un gran número de conferencias.
Movilidad	Muy difícil dar para usuarios remotos si los sistemas son locales	Fácilmente agrega usuarios remotos de cualquier tipo.
Teléfono	Puede proveer servicios de telefonía tradicional, aunque complejas.	Puede ser programado para proveer aplicaciones internas y externas de todo tipo.
Eficiencia	Ofrecer líneas de voz dedicadas con niveles de calidad pero sin flexibilidad.	Mayor eficiencia en el uso de la Red.

**Tabla 2.3 Beneficios de la Telefonía VoIP vs Tradicional.**

### 2.3.2.2 Características básicas a considerar

Las características básicas que comúnmente una pequeña organización busca en un sistema IPBX se puede encontrar en los siguientes:

### 2.3.2.2.1 Características para Enrutamiento de Llamadas

- ✓ *Asistente automatizado.* Un sistema automatizado para responder llamadas con la habilidad para construir sistemas de menú telefónico, agregar menús de llamada, transferir correos de voz y crear reglas flexibles y programables para manejar todas estas características.
- ✓ *Menús de Llamada.* Flexible gestión de llamadas con menús de opciones seleccionables por el usuario. Un mejor sistema de calidad que le permitirá tener múltiples conjuntos de menús e incluso cambiarlos basados en el tiempo o en la información obtenida desde el identificador del que llama.
- ✓ *Gestión de Extensiones.* Características para ayudar al administrados del sistema telefónico, como es la habilidad de agregar nuevas extensiones, quitar extensiones innecesarias, cambiar de lugar las extensiones y mucho mas desde un panel de control basado en Web.
- ✓ *Desvío de Llamadas.* Desvío de una llamada automáticamente, programada o de forma manual de cualquier número.
- ✓ *Transferencia de Llamada.* La habilidad de transferir llamadas entre extensiones sin tener que regresar a la centralita.
- ✓ *Call Parking.* Esencialmente poner una o varias llamadas en un área de espera en la que cualquier usuario del sistema telefónico pueda tomar la llamada.

### 2.3.2.2.2 Características de Mensajería y de Gestión

- ✓ *Correo de Voz y Buzones de Voz.* Cualquier IP PBX puede alojar un número infinito con mayor flexibilidad que un sistema de telefonía normal – más características avanzadas que podrían incluir la habilidad para grabar todas las conversaciones de entrada y salida automáticamente.
- ✓ *Llamada en espera.* Colocar correctamente las llamadas en espera, dejarlas en cola con música de espera y opciones programables acerca de los tiempos de espera.
- ✓ *Llamada en Conferencia.* Manejar llamadas de conferencia multipartidista, internamente y externamente.
- ✓ *Administración y Gestión basado en Web.* Hacer de manera rápida y fácil el manejo del sistema telefónico directamente desde un navegador Web – este puede incluir la habilidad de agregar configuraciones y manejar funciones también.
- ✓ *Directorio de Usuario.* Alguna forma de directorio de usuario y libro de direcciones que es parte del sistema telefónico y esta centralmente actualizado.

### 2.3.2.3 Características Avanzadas.

Otras características de los IPBX pueden ser extremadamente específicas, y la mezcla precisa de éstas puede hacer una gran diferencia entre las operaciones del negocio. La mayoría de esas características avanzadas vienen bajo áreas como ayuda con oficinas remotas y usuarios remotos, o programabilidad y flexibilidad del sistema.

#### 2.3.2.3.1 Gestión

- ✓ *Escalabilidad.* La Habilidad para crecer rápidamente o reducir el sistema bajo su control. Mejores sistemas pueden escalarse para manejar cientos de usuarios.
- ✓ *Gestión de Permisos.* Permitir tener diferentes grupos de empleados con diferentes permisos dentro del sistema telefónico para gestión, administración, uso y mucho más.
- ✓ *Gestión de Grupos.* Gestión de grupos de las personas que llaman y beneficiarios.
- ✓ *Gestión de petición de llamadas.* Facilitar la visibilidad y la capacidad de gestión de colas de llamada, manejo de reglas y diferentes cargas del sistema.

#### 2.3.2.3.2 Programabilidad en el enrutamiento y Scheduling

Programar la creación de reglas para dirigir las llamadas sobre criterios flexibles como identificador de llamadas o tiempo, o incluso en la próxima extensión disponible en el grupo designado.

- ✓ *Enrutamiento de llamadas programadas (Scheduling).* Manejo de diferentes llamadas entrantes basado en los tiempos de recepción - e incluso la creación de varios sistemas de gestión de llamadas que son todos diferentes, dependiendo de la hora del día o día de la semana.
- ✓ *Características de regreso de tono o llamada automática.* La característica del regreso automático de llamadas está basado sobre criterios programables.
- ✓ *Monitoreo de llamadas.* La habilidad para monitorear silenciosamente las llamadas como su progreso para propósitos como entrenamiento de ventas y soporte al cliente.
- ✓ *Barge in.* La habilidad para cortar una llamada entre dos personas – generalmente relacionadas con el monitoreo de llamadas.

### 2.3.2.3.3 Características para Usuarios Remotos.

- ✓ *Oficina de Apoyo.* La capacidad para gestionar y administrar de forma remota extensiones en otras oficinas tan fácilmente como si fueran locales.
- ✓ *Soporte de usuarios remotos como si fueran locales.* Uno de las características más poderosas de la telefonía IP es la habilidad para tener trabajadores remotos y parecer que están localmente.
- ✓ *Hoteling.* Permitir a los usuarios usar cualquier teléfono físico en el sistema actuando como si este fuera otro número, por lo que cualquier usuario puede usar cualquier teléfono dentro del sistema como si este fuera su propio teléfono en cualquier momento.

### 2.3.2.3.4 Unificar / Integración de Sistemas de Oficina

- ✓ *Integración total de Email/Outlook.* Las llamadas entrantes pueden ser combinados con los registros de gestión de contactos, y las llamadas salientes pueden ser iniciadas desde Outlook de modo que los usuarios pueden hacer click sobre el contacto del manejador de registros del sistema y marcar desde dentro de Outlook o de otras aplicaciones.
- ✓ *Correo de Voz por Email.* Enviando todos los correos de voz desde un buzón a una cuenta de email donde los mensajes pueden ser abiertos y escuchados por una PC – así como almacenar y manejar.
- ✓ *Integración de la Red de Datos.* Alguna forma de integración en su red de datos para que "haga clic para llamar" como funcionalidad, la integración con documentos de oficina, sistemas de correo electrónico o incluso los sistemas de CRM pueden añadirse.
- ✓ *Click-to-Dial.* Alguna forma de dar click para llamar, es decir, para ir de un número en pantalla de un PC a una llamada en el teléfono sin tener que marcar el número usted mismo.

### 2.3.2.4 Características Únicas.

- ✓ *Respuesta de Voz Integrado.* El sistema incluye la habilidad para el que llama navegar a través de menús usando el teclado del teléfono o por respuestas de voz

- ✓ *Manejo de Análogo e IP.* Muchos sistemas IPBX pueden manejar ambos, telefonía VoIP y la telefonía regular al mismo tiempo - aunque no toda la funcionalidad está disponible para los usuarios de teléfonos normales.
- ✓ *Fax IP.* Un sistema de fax integrado dentro del sistema telefónico IP. No puedes usar directamente las máquinas de fax en la red telefónica IP sin algún tipo de interface.
- ✓ *Función de Presencia e integración de Mensajes instantáneos.* La función de presencia indica el estado de un usuario dentro del sistema telefónico a otros usuarios, incluso aquellos usuarios externos, si esta función está disponible. Estos indicadores pueden ser tan amplios como para decir la localización, tipo de dispositivo por el cual la comunicación toma lugar, y la transferencia entre los métodos de enrutamiento.

**2.3.2.5 Algunas Alternativas.**

Teniendo en cuenta los requerimientos que podemos encontrar, mencionaremos algunas soluciones comerciales en la siguiente tabla:

<b>Mano factura</b>	3Com	Altigen	Avaya	Cisco
<b>Producto</b>	3Com NBX V3000 BRI-ST	AltiServ	IP Office	Unified Call Manager/Media Convergence Server 7800 Series
<b>Integración de Outlook</b>	Included	Included	Optional via VoiceMail Pro	Included
<b>Integración de Respuesta de Voz</b>	Yes, with optional add-on	Multiple options with call-center software	Included	Included
<b>Soporte de Call Center</b>	Basics included, advanced capabilities available with add-on	With call center software	Yes, with optional add-on	Included
<b>Plataforma de Software</b>	Security-hardened Linux	Windows-based	IP Office release 4	Either Windows or Linux
<b>Plataforma de Hardware</b>	Proprietary V3000 NBX platform	Proprietary	Avaya Office IP 500	Cisco CallManager and Media Convergence servers
<b>Protocolo(s) Soporta-</b>	SIP (Session Initiation	SIP, H.323, 802.x	SIP, H.323, 802.x	SIP, H.323 and

<b>dos</b>	Protocol), H.323, 802.x and many more protocols supported	and many more protocols supported	and many more protocols supported	numerous other standards
<b>Gestión y Administración</b>	Browser-based interface supports provisioning, regular administration and management features and integrates with 3Com's enterprise network- management suites	User- and system- management software options	Windows-based graphical system- management tools and end user/client tools	Suite of management and administration tools that support functions across groups of phones as well as single phones
<b>Soporte de teléfonos</b>	Multiple third-party SIP devices	Two proprietary, IP hardphones, one softphone and analog phone support		Numerous hard SIP phones
<b>Soporte de teléfonos análogos</b>	Yes, built-in	Yes	Yes	Yes
<b>Seguridad</b>	Encrypted VoIP, strong phone authentication and strong management securityhybrid	No VoIP security- hybrid system	Encrypted VoIP, strong phone authentication, strong management security	Excellent encrypted VoIP, phone authentication and management security
<b>Capacidad de estación IP</b>	Expandable up to 1,500 lines	48 users per box with multiple boxes-overall system supports up to several hundred users	Starts at 32 lines, expandable to 270 lines	Up to 7,500 phones per server and 30,000 phones per cluster
<b>Modularidad/Escalabilidad</b>	Highly modular/expandable. Good integration with 3Com network products	Modular and expandable, with units suitable for main offices and branch offices	Modular and expandable with options for main and branch offices	Modular and expandable with cards/additional units; lots of additional modules for extra features
<b>Redirigir a PSTN (red telefónica pública conmutada) en caso de fallo</b>	Full failover	No-but does explicitly include backup and disaster recovery options	Yes	Yes
<b>Documentación/Soporte</b>	Enterprise level documentation and support levels	Enterprise-class support and do-	Enterprise-class support and do-	Enterprise level documentation

		cumentation	cumentation	and support levels
<b>Precio</b>	Starts at approx \$20,000 for 50-user system	Up to \$18,000 depending on reseller and precise features and configurations selected.	\$16,900	Starts at approximately \$20,000
<b>Fácil Instalación</b>	Flexible solution from 3Com that can connect directly to PSTN as well as IP networks. But it is also a complex system that requires professional help to install.	Flexible system with lots of configurable options; is managed by AltAdmin software	Installation straightforward; configuration cumbersome at some points	Complex system that requires extensive analysis and testing before deployment
<b>Tamaño de Negocio al que va dirigido.</b>	Medium to large enterprise	SMB (small- to medium-sized business)	SMB	Large enterprise
<b>Comentarios</b>	A scalable and powerful proprietary solution that integrates well with other 3Com products. Most advanced features require add-ons, and it is recommended that installation and configuration services be used to make best use of the product.	Older company with very robust system that continues to be extended; feature-rich	Award-winning. Good choice for reliability and performance	A highly scalable, powerful and extremely extendable solution.

<b>Mano factura</b>	Digium	Fonality	Intuitive Voice	Mitel
<b>Producto</b>	<b>Asterisk Business Edition</b>	<b>PBXtra</b>	<b>Evolution PBX V2.3</b>	<b>SX-200 IP Communications Platform</b>
<b>Integración de Outlook</b>	Click-to-dial available as plug-in	Included		Included
<b>Integración de Respuesta de Voz</b>	Included	Included	Included	Included
<b>Soporte de Call Center</b>	Requires significant configuration	Included with Call Center Edition	No	Included
<b>Plataforma de Software</b>	Open-source, Red Hat Linux based, with extensive customization	Based on Asterisk, true Web-based GUI; extremely intuitive for administrators and	Open Source - Asterisk	Proprietary – Integrates with common network systems

		users alike		
<b>Plataforma de Hardware</b>	Pretty much any Linux server	Pretty much any Linux server	Preinstalled on customized Linux servers	Proprietary
<b>Protocolo(s) Soportados</b>	SIP, H.323, 802.x and many more protocols supported	SIP, H.323, 802.x and many more protocols supported	SIP, H.323, 802.x and many more protocols supported	SIP, H.323, 802.x and many more protocols supported
<b>Gestión y Administración</b>		Good encrypted VoIP, phone authentication and management security	Standard Asterisk administration	Ships with two management tools: MyAdministrator & IP Phone Analyzer; MACs are simple to execute
<b>Soporte de teléfonos</b>	40+ hardphones supported; numerous soft-phone clients	Six IP hardphones supported; one IP softphone (bundled); also supports analog phones	Polycom, Linksys, Cisco and Counterpath phones	Supports very wide range of SIP phones and IP devices
<b>Soporte de teléfonos análogos</b>	Yes	Yes	Yes	Yes
<b>Seguridad</b>	Encrypted VoIP, moderate phone, management and authentication security. Add-ons can extend security.	Encrypted VoIP, moderate phone, management and authentication security. Add-ons can extend security.	Encrypted VoIP, moderate phone, management and authentication security.	Strong encrypted VoIP, phone authentication and management security
<b>Capacidad de estación IP</b>	Expandable and modular – low cost makes it easy to upgrade for branch office	Expandable and modular – low cost makes it easy to upgrade for branch office	Expandable and modular – low cost makes it easy to upgrade for branch office	Allows expansion up to more than 600 users
<b>Modularidad/Escalabilidad</b>	Highly modular/expandable. Good integration with 3Com network products	Modular and expandable, with units suitable for main offices and branch offices	Modular and expandable with options for main and branch offices	Modular and expandable with cards/additional units; lots of additional modules for extra features
<b>Redirigir a PSTN (red telefónica pública conmutada) en caso</b>	No	No	No	Yes

<b>de fallo</b>				
<b>Documentación/Soporte</b>	Includes Asterisk user manual and quick-start guide; free support for one year	Responsive support with fast response times; 'open-source' style documentation	Offers support and services ongoing. Documentation is basic and straightforward.	Enterprise class support and documentation
<b>Precio</b>	Starts at \$995	Starts at \$995	Starts at \$995	\$12,000
<b>Fácil Instalación</b>	Integrated installation now much improved. At heart, a simpler system than the complex proprietary solutions but still not plug and play.	Intuitive GUI streamlines installation; delivered as appliance; up and running in 30 minutes	Asterisk-based turnkey system that is easy to install and operate.	Straightforward installation via MyAdministrator application
<b>Tamaño de Negocio al que va dirigido.</b>	Medium to large enterprise	SMB (small- to medium-sized business)	SMB	Large enterprise
<b>Comentarios</b>	Digium adds valuable commercial license, support and documents to already robust Asterisk	Low-cost, all-inclusive system. Good for hybrid IP/analog setup requiring easy maintenance	Low-cost Asterisk system – clear pricing and support options.	High-end communications system with very many unified communications features, including integration of mobile devices

**Tabla 2.4 Alternativas de Asterisk para un Servicio IPBX obtenidas de voip-news.com[8]**

Un sistema IPBX puede costar alrededor de \$1,000 US. dls en adelante sin incluir los teléfonos. Esto puede dar una idea de las alternativas para medianas y grandes empresas u organizaciones, que requieren ciertas funciones o características, además de soporte y facilidad de instalación y uso.

## 2.4 Factura

La factura es un documento que refleja la entrega de un producto o la provisión de un servicio, junto a la fecha de devengo, además de indicar la cantidad a pagar como contraprestación (véase fig. 2.5). En la factura se encuentran los datos del expedidor y del desti-

[8] [www.voip-news.com](http://www.voip-news.com), sitio de internet dedicado a ofrecer información acerca de la tecnología VoIP, la tabla obtenida de *The Definitive Guide to Premise-Based PB*.



meter la entrega de un bien o servicio y obligar a realizar el pago correspondiente, de acuerdo con lo establecido en el propio documento. Es utilizada por el comprador y por el vendedor como comprobante ante las autoridades y en las auditorías internas.

La factura electrónica puede ser enviada, archivada y transmitida por medios electrónicos, pero también es posible imprimirla bajo las especificaciones del SAT, en caso de que así se requiera.

Igual que en el proceso tradicional, la ley obliga a todos los contribuyentes a conservar las facturas electrónicas por un periodo mínimo de cinco años en el formato original (XML). Sin embargo, la factura electrónica no es sólo para las grandes empresas, también está al alcance de las pequeñas y medianas empresas (PYMES) y de personas físicas. Sus beneficios son indudables, por lo que se prevé que poco a poco se extienda su uso. En la figura 2.6 se muestran algunos de los formatos que puede optar una factura electrónica.

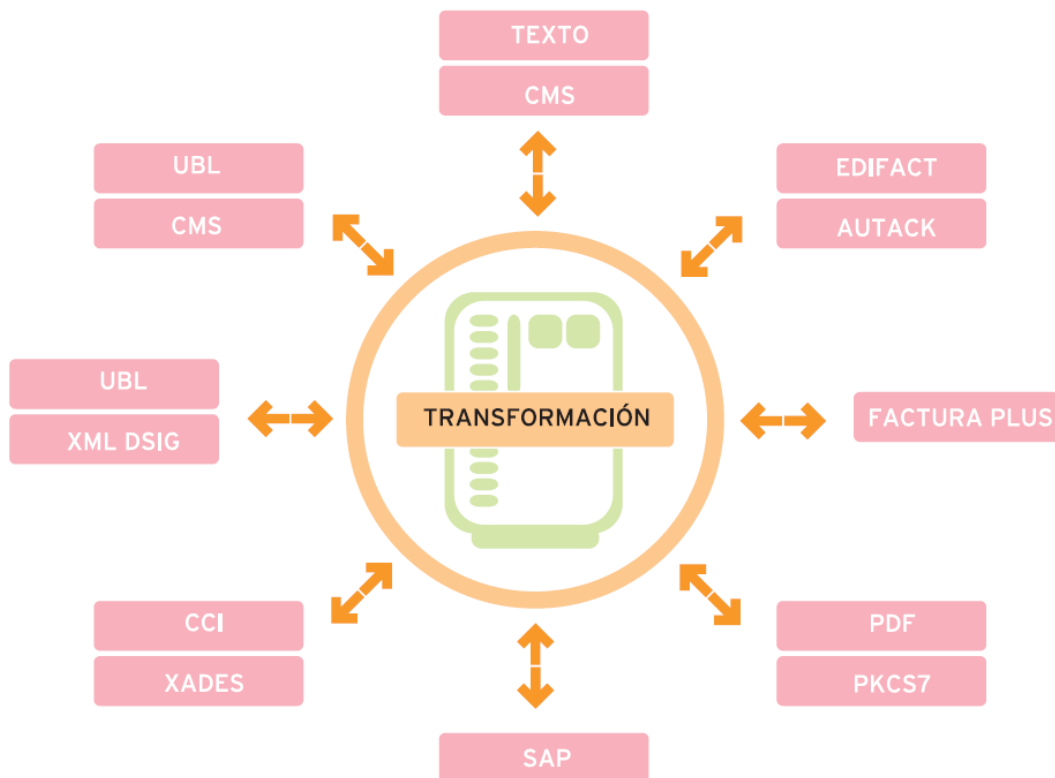


Figura 2.6 Algunos de los formatos para la e-factura.

**2.5.1 Beneficios**

- Reduce tiempos y procesos administrativos, brinda rapidez y seguridad en el intercambio de información y agiliza la recepción de mercancía, lo cual se traduce en ahorros y en un importante incremento de la productividad.
- Logra ahorros mayores a 50% de los costos en el proceso de facturación (en gastos de papelería, bodegas de almacenaje y envíos).
- Aunque implica un cambio de cultura y una reingeniería de los procesos, las obligaciones fiscales se simplifican. Facilita los procesos de auditoría, ya que permite la búsqueda y localización rápida y en un mismo lugar.
- Facilita procesos administrativos, recepción y envío oportunos, así como la posibilidad de explorar la información. No hay riesgo de que se infecte con virus informáticos.
- Asegura confidencialidad en el manejo, resguardo y envío de facturas. Disminuye la posibilidad de falsificación: a una persona le llevaría 10 años descifrar el código de una llave privada asociada a un certificado de sello digital (elementos que sirven para generar el sello digital de cada factura), en comparación con los 30 minutos que le tomaría a un falsificador copiar una firma en papel.

El receptor también se beneficia al tener la oportunidad de acceder a la oferta de servicios para recibir y validar de forma automática si la factura está vigente y no ha sido alterada. En la tabla siguiente se muestra los beneficios de la e-factura con respecto a una factura común.

Descripción	Sin factura electrónica	Con factura electrónica
Costo de distribución	100%	10%
Tiempo de procesamiento	100%	40%
Tiempos de entrega	100%	0.5%

**Tabla 2.5 Beneficios de la E-factura vs Factura.**

## 2.6 Lenguajes usados.

### 2.6.1 XML

XML significa *eXtensible markup language*, o lenguaje de anotación extensible. Ya conocemos el lenguaje HTML (*hypertext markup language*), lenguaje de anotación para página webs que permite navegación tipo hipertexto; sin embargo, XML no es sólo un lenguaje, es una forma de especificar lenguajes, de ahí lo de extensible. Todo lenguaje que se exprese de una forma determinada puede ser XML. Por lo tanto, XML no es un lenguaje para hacer mejores páginas web, sino un lenguaje para información auto-descrita, o al menos, auto-descrita si las etiquetas están bien puestas.

XML usa *tags* para etiquetar, categorizar y organizar información en una forma específica. El *marcado (Markup)* describe un documento o una estructura de datos y la organización. El contenido, como texto, imágenes y datos, es la parte del código que las etiquetas de marcado contienen; es también lo de mayor interés para la mayoría de las personas que leen cada día o interactúan con los datos o documentos. XML no se limita a un conjunto de marcas, crear su propio marcado para satisfacer sus necesidades de datos y documentos. La flexibilidad de XML ha conducido a su amplia utilización para el intercambio de datos en una multitud de formas.

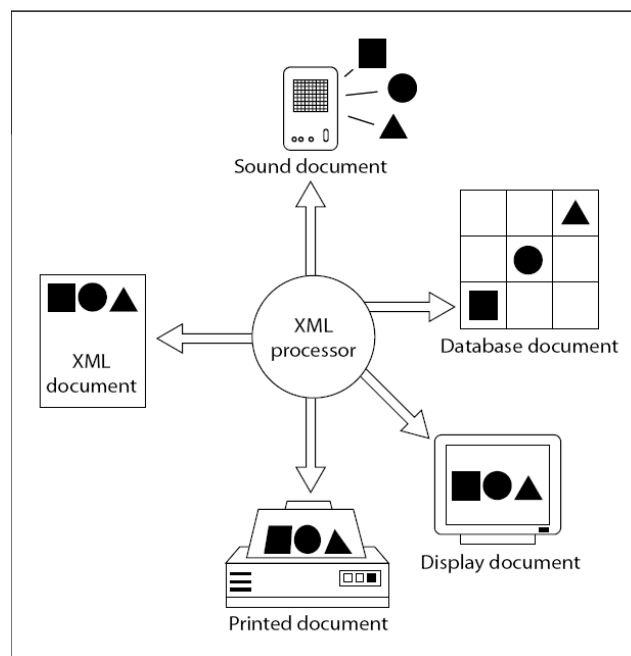


Figura 2.7 Uso de XML para distintas salidas.

XML se puede usar para cambiar totalmente el paradigma de publicación; de un programa que recibe unas entradas y produce unas salidas, se pasa a un documento que genera otro documento, o bien programas que toman documentos y producen otros documentos. Por eso, también, y, en general, salvo en entornos de servicios web, lo normal es que el XML se use en el servidor, y se sirva otro tipo de documentos, HTML, por ejemplo, que se obtienen a base de una serie de transformaciones. Precisamente, esto hace que los documentos XML se usen dentro de entornos de aplicaciones. Este entorno de aplicaciones permite publicar documentos XML, que, antes de ser enviados al cliente, sufrirán una serie de transformaciones para adaptarlo a los requisitos del mismo.

### 2.6.2 XSLT

Al igual que XML, XSLT es un lenguaje de programación. Forma parte de la trilogía transformadora de XML, compuesta por las CSS (*Cascading Style Sheets*, hojas de estilo en cascada), que permite dar una apariencia en el navegador determinada a cada una de las etiquetas XML; XSLT (*XML Stylesheets Language for Transformation*, o lenguaje de transformación basado en hojas de estilo); y XSL-FO, (*Formatting Objects*, objetos de formato), o transformaciones para fotocomposición, o, en general, para cualquier cosa que no sea XML, como por ejemplo HTML “del viejo” o PDF (el formato de Adobe).

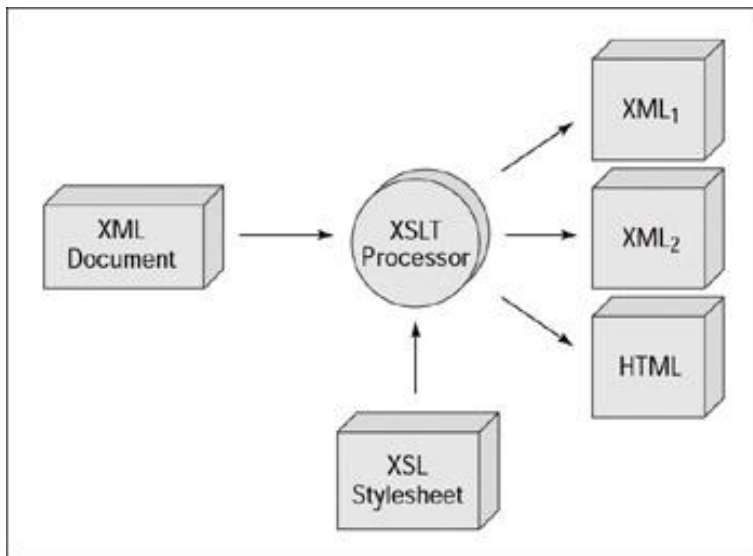


Figura 2.8 XSLT transforma un XML en varias salidas.

XSLT es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. Las hojas de estilo XSLT realizan la transformación del documento utilizando una o varias reglas de plantilla. Estas reglas de plantilla unidas al documento fuente a transformar alimentan un procesador de XSLT, el que realiza las transformaciones deseadas poniendo el resultado en un archivo de salida, o como en el caso de una página web, las hace directamente en un dispositivo de presentación tal como el monitor al usuario.

### 2.6.3 XSL-FO

XSL-FO es un lenguaje XML diseñado para describir todos los aspectos visuales de documentos paginados. El ya tan conocido HTML es otro lenguaje para especificar formatos semánticos, pero es más usado para documentos que son presentados en pantalla, y menos usado para materiales destinados a imprimirse, porque este no soporta elementos paginados como son cabeceras y pie de página, especificación de tamaño de página, notas de pie de página, etc.

La unidad básica de trabajo de un documento XSL-FO es el *“Formatting Object”*, para presentar (formatear) la información. Estos objetos de formato se refieren a páginas, párrafos, tablas, imágenes, etc. Los documentos XSL-FO son archivos XML con información de salida. Ellos contienen información acerca del formato y contenido de salida.

Estos mismos documentos especifican estructura de la página, tamaño de la página, cualquier cabecera o pie de página, márgenes y números de páginas, etc. Por ejemplo, las especificaciones de una página pueden ser para páginas de tamaño A4 de cierto ancho y alto. El título de la página puede ser especificado de forma separada desde el contenido principal. Otras páginas necesitan especificaciones de manera distinta. Los documentos XSL-FO también especifican en detalle como cada pieza del contenido puede ser formateado, por ejemplo, los títulos pueden ser grandes, en negritas y centrados. Hay también algunas tareas suplementarias claves, incluyendo generar tablas de contenido, listas, e indexar párrafos.

El proceso empieza con un documento XML el cual es dado o que tiene que ser creado: el XML de origen. Se toma el documento y se aplica una transformación XSLT (usando una hoja de estilo XSLT ) para seleccionar las partes o todo del contenido del documento, y este produce una salida en documento XML que usa el vocabulario XSL-FO (véase fig. 2.9).

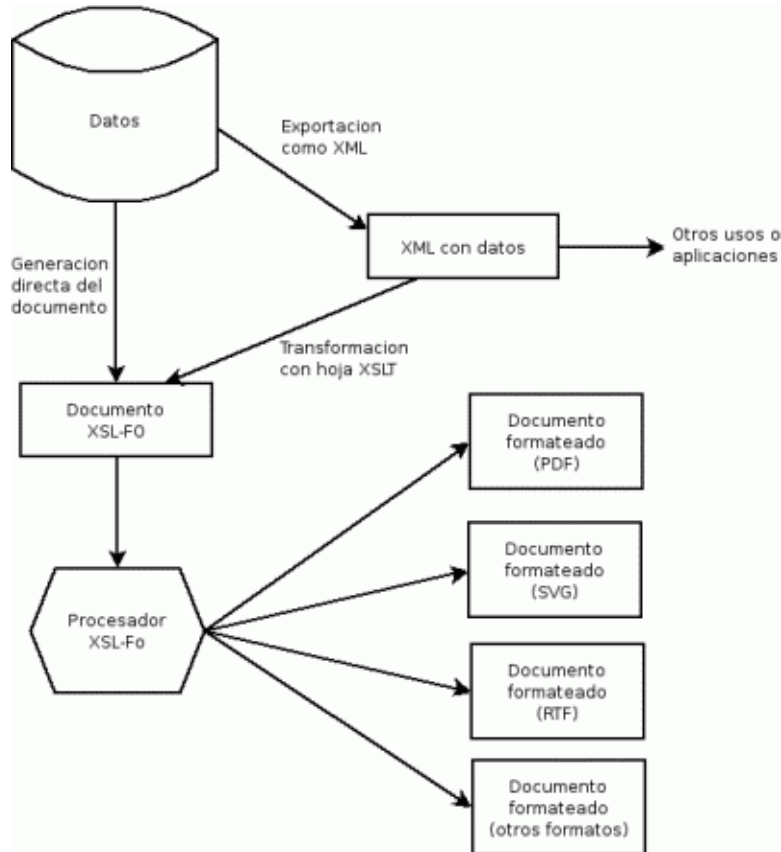


Figura 2.9 Vista del Proceso de XSL-FO

#### 2.6.4 SQL

**SQL (Structured Query Language** traducido al español **Lenguaje de consulta estructurado**) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre la misma.

### 2.6.5 Shell Script

Normalmente los intérpretes de comandos (*Shell*) son interactivos. Esto significa que el *Shell* acepta comandos desde el teclado y los ejecuta. Como siempre, si tu guardas una secuencia de comandos en un archivo de texto y le dices al *Shell* que ejecute este archivo instanciado de varios comandos, este es conocido como un programa de *Shell* o *Shell Script*. También usamos el término *Shell Script* para referirnos a programas escritos para la *Shell* de UNIX/LINUX, mientras que cuando usamos la línea de comandos de MS-DOS, nos referimos como Batch files (archivos por lotes).

El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario, además de realizar tareas repetitivas.

Cada *Shell script* consiste de

- Comandos de *Shell* como son if,else,do,while
- Comandos binarios de Linux tales como w, who, free, etc.
- Herramientas de procesamiento de texto en Linux como grep, awk, cut, etc.

### 2.6.6 PHP

*PHP* es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente PHP Tools, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre. En la figura 2.10 se muestra como el programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.



**Figura 2.10 Diagrama del Comportamiento de PHP**

Entre sus ventajas se encuentran:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación

y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño **Modelo Vista Controlador** (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

### 2.6.7 Ajax

*AJAX*, acrónimo de *Asynchronous JavaScript And XML* (*JavaScript* asíncrono y *XML*), es una técnica de desarrollo web para crear aplicaciones interactivas o *RIA* (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. El término *AJAX* se acuñó por primera vez en el artículo “*Ajax: A New Approach to Web Applications*” publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo

Las tecnologías que forman *AJAX* son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

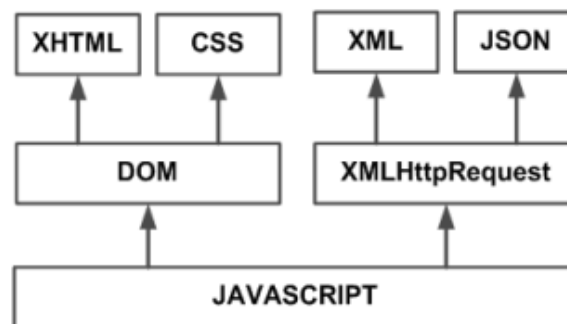
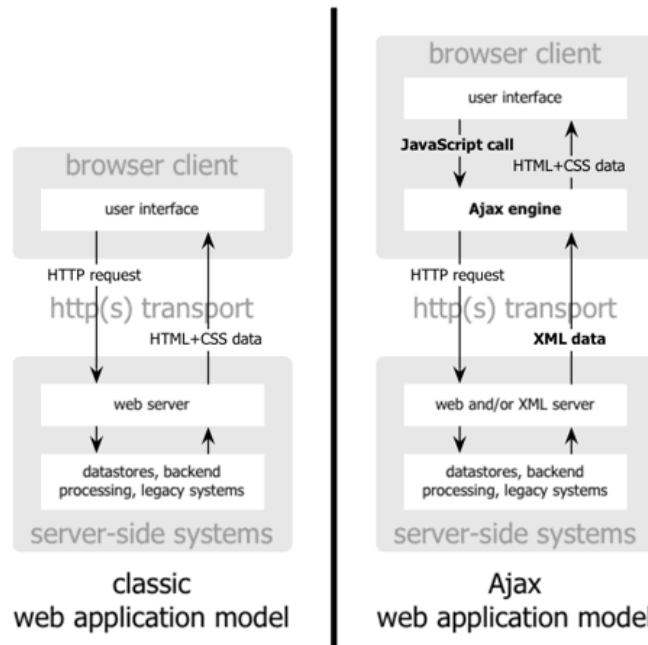


Figura 2.11. Tecnologías agrupadas bajo el concepto de *AJAX*

En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario.



**Figura 2.12.** La imagen de la izquierda muestra el modelo tradicional de las aplicaciones web. La imagen de la derecha muestra el modelo de Ajax.

Esta técnica tradicional para crear aplicaciones web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, la aplicación web se convierte en algo más molesto que útil.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

## CAPÍTULO 3

# ANÁLISIS Y DISEÑO

### 3.1 Introducción

En el siguiente capítulo describiremos el análisis y diseño del sistema, nos auxiliaremos con técnicas de modelado de sistemas computacionales que se ubican dentro del ámbito de la ingeniería de software. Esta es la técnica de modelado para el desarrollo de sistemas conocida como Lenguaje de Modelado Unificado *UML* por las siglas en inglés (*Unified Modeling Language*). Este lenguaje permite la descripción, el análisis y diseño de sistemas complejos mediante el uso de diagramas. Entre ellos tenemos a los “Diagramas de Casos de Uso”, “Diagramas de Actividades”, “Diagramas de Secuencia”, “Diagramas de estados”, etc.

*UML* capta la información de la estructura estática y el comportamiento dinámico del sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo, además, pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. *UML* no es un lenguaje de programación. Pero sí una herramienta que ayuda a los desarrolladores a especificar, visualizar, construir y documentar un sistema de software.

Es importante señalar que, en el desarrollo de cualquier sistema es necesario cumplir con un proceso cíclico de depuración y refinamiento. Con el fin de visualizar el concepto al que hacemos referencia nos auxiliaremos de la figura 3.1.

### 3.2 Especificación de Requerimientos

Con la introducción de un Sistema VoIP en la Facultad de Ciencias de la Computación y el incentivo al uso de este nuevo servicio a la facultad, será requerido el control del uso del

mismo, por lo que se necesita desarrollar una aplicación que permita registrar el uso y proporcionar información del mismo.

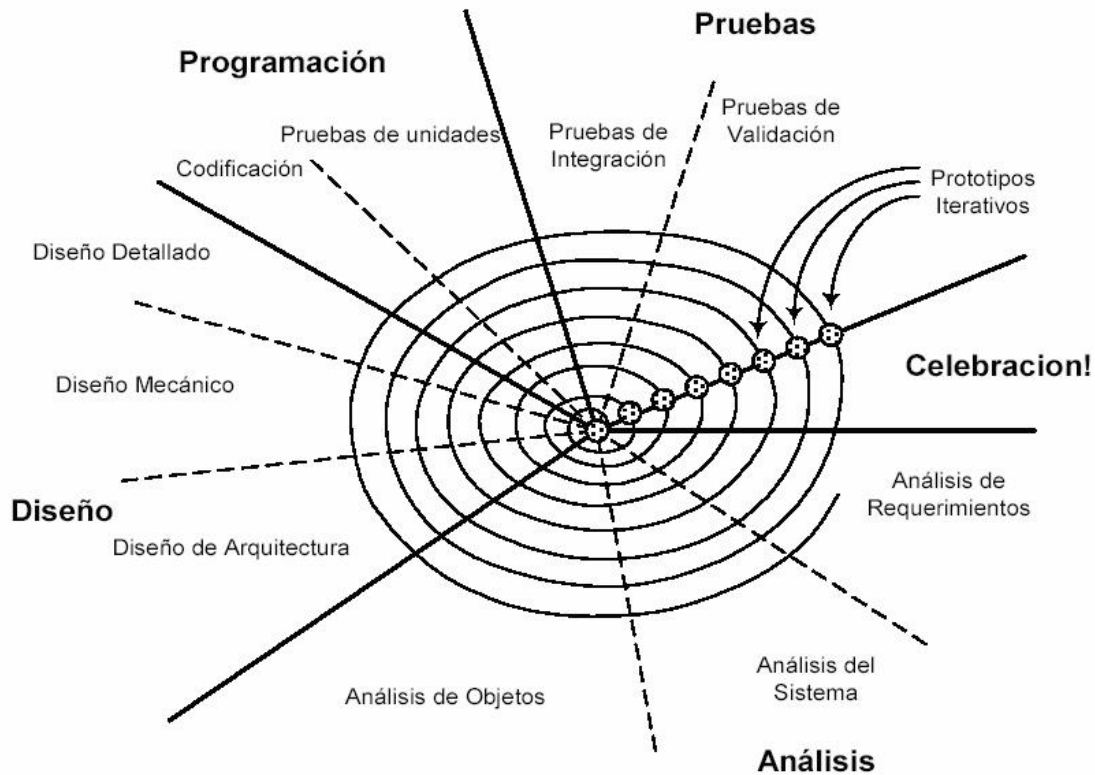


Figura 3.1 Proceso Iterativo de Desarrollo

A continuación se da una descripción del sistema:

El sistema de facturación es un sistema que permite al usuario hacer consultas y recibir información acerca del uso del servicio VoIP de Asterisk. Las consultas se harán vía web, a través de un navegador de internet.

El sistema de consultas presenta un mensaje de bienvenida describiendo las consultas posibles dependiendo del tipo de Usuario. Este acceso se da por medio de la inserción de un login previamente especificado y un password previamente asignado y que debe validarse. Una vez de haberse validado el login y password se pueden seleccionar las siguientes actividades:

- Consulta de Llamadas.
- Consulta por quién llamó más.
- Consulta por llamada de mayor duración.
- Consulta de Información de Usuarios.

En Consulta de llamada muestra al usuario o admin las llamadas que han realizado en el servicio VoIP. En el caso de los usuarios solamente podrán ver la información de sus llamadas. En el caso del Admin, este podrá consultar las llamadas de cada usuario o extensión.

En Consulta por quién llamó más se muestra al usuario quien hizo mas llamadas en una fecha determinada y esta consulta solo la puede realizar el Admin.

En Consulta por llamada de mayor duración, se le muestra al usuario quien hizo la llamada mas larga en una fecha determinada, al igual que la anterior consulta solo puede realizarla el Admin.

En Consulta de Información de Usuarios permite al Admin mostrar datos acerca de los usuarios, como es la extensión, id de teléfono, nombre y email,

Además de forma automática cada mes el usuario recibirá su respectiva factura (reporte) sobre el uso del servicio, el cual será enviado por el sistema a su dirección electrónica de correo. Esta factura será sobre un mes de cobro, el cual será el mes anterior de la fecha actual. También el sistema hará semanalmente una actualización de la información almacenada por Asterisk (*CDR*[9]).

El sistema deberá mantener un control de la información almacenada, así como registrar cada operación que realice, también de enviar cualquier contratiempo o error en las operaciones al administrador del sistema.

---

<sup>9</sup> **Call Detail Record (CDR)**, es un registro de computadora producido por una telefónica, el cual contiene detalles de una llamada.

### 3.3 Objetivos del sistema

En este apartado vamos a definir una lista con los diferentes objetivos que se esperan alcanzar cuando el sistema software a desarrollar esté en explotación. Serán especificados mediante una plantilla para objetivos.

<b>OBJ-01</b>	<b>Gestionar las Base de Datos</b>
<b>Descripción</b>	El sistema deberá gestionar los registros guardados en las Base de Datos A y B, además de hacer los procesos entre ellas.
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

<b>OBJ-02</b>	<b>Crear y Enviar las Facturas a los Usuarios.</b>
<b>Descripción</b>	El sistema deberá crear las respectivas facturas para cada usuario, además de enviárselos a su respectivo correo electrónico.
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

<b>OBJ-03</b>	<b>Gestión de Información vía Web</b>
<b>Descripción</b>	El sistema deberá permitirle al usuario consultar su información del uso del Servicio.
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

### 3.4 Requisitos de almacenamiento de información

Esta sección contiene la lista de requisitos de almacenamiento de información que se han identificado. Especificaremos toda la información que debemos almacenar en nuestro sistema.

<b>RI-01 Información sobre los Registros CDR de Asterisk</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestionar las Bases de Datos
<b>Requisitos asociados</b>	<ul style="list-style-type: none"> <li>• RF-01 Registro de Llamadas en Base de Datos A</li> <li>• RF-02 Paso de Llamadas de BD(A) a la BD(B).</li> </ul>
<b>Descripción</b>	El sistema deberá almacenar y procesar la información correspondiente a los registros CDR de Asterisk. En concreto:
<b>Datos específicos</b>	<ol style="list-style-type: none"> <li>I. Origen de la llamada</li> <li>II. Destino de la llamada</li> <li>III. Duración de la llamada en segundos</li> <li>IV. Fecha en que se realizó la llamada, en año, mes y día</li> <li>V. Hora de la llamada, en horas, minutos y segundos</li> <li>VI. Identificación única asignada por Asterisk</li> <li>VII. Nombre del usuario registrado por la aplicación de telefonía.</li> </ol>
<b>Intervalo temporal</b>	pasado y presente
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

<b>RI-02 Información de Facturación.</b>	
<b>Objetivos asociados</b>	<ul style="list-style-type: none"> <li>• OBJ-02 Crear y Enviar las Facturas a los Usuarios</li> </ul>
<b>Requisitos asociados</b>	<ul style="list-style-type: none"> <li>• RF-03 Facturación</li> </ul>
<b>Descripción</b>	El sistema deberá procesar la información correspondiente a la Base de Datos B. En concreto:
<b>Datos específicos</b>	<ol style="list-style-type: none"> <li>VIII. Origen de la Llamada</li> <li>IX. Destino de la llamada</li> <li>X. Duración de la llamada en segundos</li> <li>XI. Fecha en que se realizó la llamada, en año, mes y día</li> <li>XII. Hora de la llamada, en horas y minutos</li> <li>XIII. Nombre del Usuario</li> <li>XIV. Email del Usuario</li> <li>XV. El tipo de llamada obtenida por la forma de marcación</li> </ol>

<b>Intervalo temporal</b>	sólo presente
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

<b>RI-03</b>	<b>Información sobre el uso de los usuarios</b>
<b>Objetivos asociados</b>	OBJ-03 Gestión de Información vía Web
<b>Requisitos asociados</b>	<ul style="list-style-type: none"> <li>• RF-04 Llamadas y No. De Llamadas</li> <li>• RF-05 Por quién llamó más</li> <li>• RF-06 Por llamada de mayor duración</li> <li>• RF-07 Información de Usuarios.</li> </ul>
<b>Descripción</b>	El sistema deberá procesar la información correspondiente a la Base de Datos B. En concreto:
<b>Datos específicos</b>	XVI. Origen de la Llamada XVII. Destino de la llamada XVIII. Duración de la llamada en segundos XIX. Fecha en que se realizó la llamada, en año, mes y día XX. Hora de la llamada, en horas, minutos y segundos XXI. Nombre(s) del(os) Usuario(s) XXII. email(s) del(os) Usuario(s) XXIII. El tipo de llamada obtenida por la forma de marcación
<b>Intervalo temporal</b>	sólo presente
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Un usuario solamente puede consultar su propia información, el admin puede consultar todas las llamadas que se registren.

### 3.5 Diagramas de casos de uso

En esta sección hemos incluido los diagramas de casos de uso de nuestro sistema, desarrollados con la herramienta StarUML.

Usando la información planteada anteriormente del problema se expondrá cada caso de uso, definiendo sus actores y la relación que tienen con el uso del sistema, las cuales se describen en los siguientes apartados.

### 3.5.1 Definición de actores

Este apartado contiene los diferentes actores que se han identificado, especificados mediante la plantilla para actores de casos de uso.

<b>ACT-01</b>	<b>Asterisk</b>
<b>Descripción</b>	Este actor representa al sistema IPBX Asterisk
<b>Comentarios</b>	Ninguno

<b>ACT-02</b>	<b>Usuario</b>
<b>Descripción</b>	Este actor representa a los usuarios registrados en el sistema Asterisk
<b>Comentarios</b>	Ninguno

<b>ACT-03</b>	<b>Admin.</b>
<b>Descripción</b>	Este actor representa a la Administración o al Administrador.
<b>Comentarios</b>	Ninguno

<b>ACT-04</b>	<b>Cronograma de Facturación</b>
<b>Descripción</b>	Este actor representa el día de proceso de facturación
<b>Comentarios</b>	Ninguno

<b>ACT-05</b>	<b>Cronograma Semanal</b>
<b>Descripción</b>	Este actor representa el día de proceso para trasladar las llamadas de la Base de Datos A a la Base de Datos B
<b>Comentarios</b>	Ninguno

### 3.5.2 Casos de uso del Sistema

El sistema contiene tres partes principales, las cuales son:

- xxiv. Gestionar las Bases de Datos
- xxv. Crear y Enviar las Facturas a los Usuarios (Facturación).
- xxvi. Gestión de Información vía Web

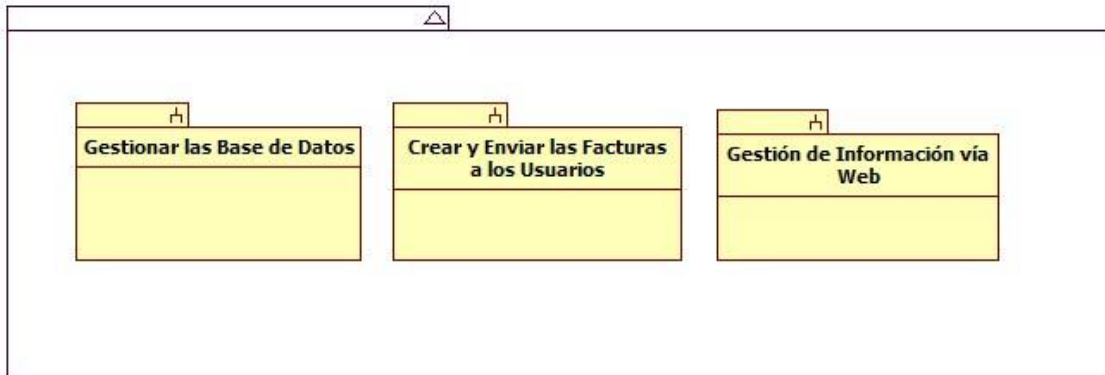


Figura 3.2 Diagrama de subsistemas

A continuación se muestran de forma visual cada caso de uso del Sistema de Facturación.



Figura 3.3 Diagrama de casos de uso del subsistema Gestionar las Bases de Datos

En la figura 3.3 se muestra la interacción de los actores que inician los procesos de gestión de los datos guardados por el sistema Asterisk.

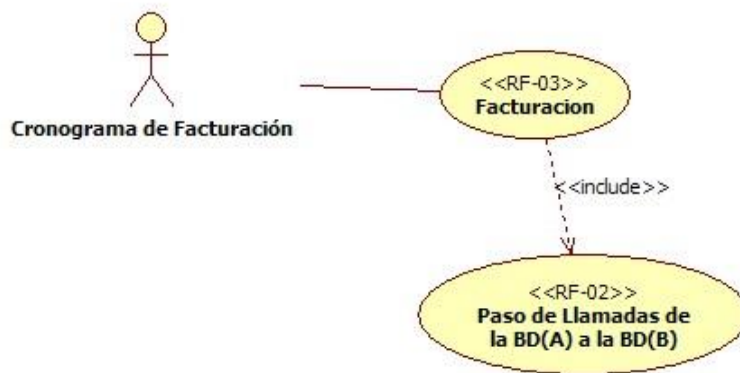


Figura 3.4 Diagrama de casos de uso del subsistema Crear y Enviar las Facturas a los usuarios

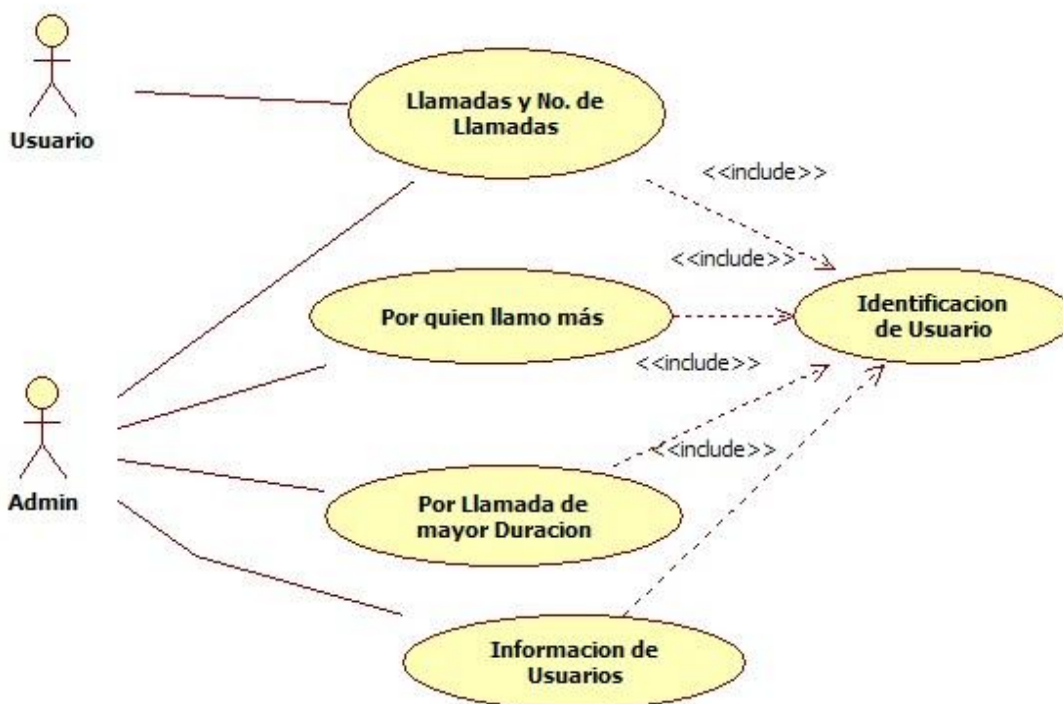


Figura 3.5 Diagrama de casos de uso del subsistema Gestión de Información vía Web

En la figura 3.4 se muestra el diagrama para la facturación el cual es descrito en la sección 3.5.3, al igual que el diagrama de la Gestión de Información vía Web figura 3.5.

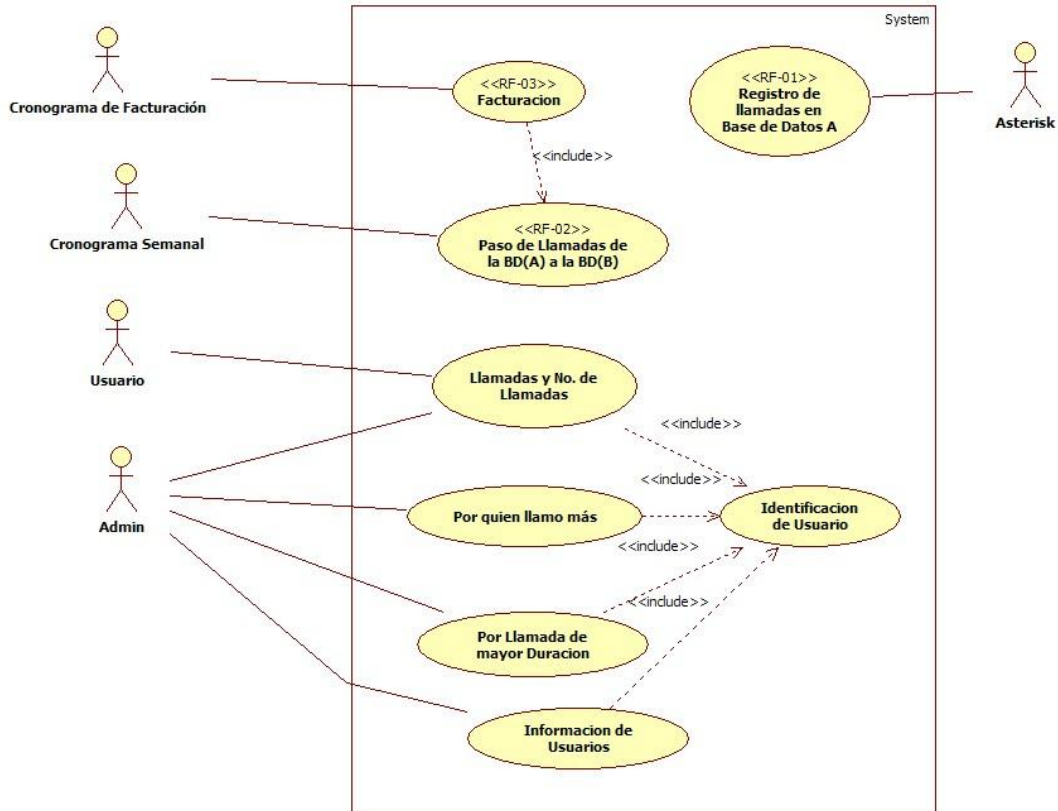


Figura 3.6 Diagrama general de casos de uso del Sistema de Facturación.

### 3.5.3 Especificación de casos de uso

A continuación describe cada uno de los casos mostrados en los diagramas anteriores:

RF- 01		Registro de Llamadas en Base de Datos A
Objetivos asociados	OBJ-01 Gestionar las Bases de Datos	
Requisitos asociados	RI-01 Información sobre los CDR de Asterisk	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando Asterisk registre llamadas.	
Precondición	Ninguna	
Secuencia	<b>Paso</b>	<b>Acción</b>
Normal	1	Asterisk abre comunicación con la Base de Datos A, para guardar los registros.

	2	Se guarda la siguiente información en la Base de Datos A: calldate, callans, callend, clid, src, dst, dcontext, channel, dstchannel, lastapp, lastdata, duration, billsec, disposition, amaflags, accountcode, Userfield, uniqueid.
	3	Cierra comunicación.
<b>Postcondición</b>	Se almacenan los registros de llamadas por Asterisk	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Si no consigue comunicación Asterisk con la base de datos, se registrara en un log.
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia dependerá del número de usuarios que estén ocupando el servicio VoIP.	

RF- 02		Paso de Llamadas de la BD(A) a la BD(B)
<b>Objetivos asociados</b>	OBJ-01 Gestionar las Bases de Datos	
<b>Requisitos asociados</b>	RI-01 Información sobre los CDR de Asterisk	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando sea solicitado por el Cronograma Semanal o por el Cronograma de Facturación.	
<b>Precondición</b>	Ninguna	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Cronograma ejecuta el proceso de copiado entre base de datos.
	2	Se verifica el estado de las Bases de Datos
	3	Se respaldan las Base de Datos.
	4	Se procesa los registros de la Base de Datos A, con la condición que sean tipo ANSWERED y que el origen de la llamada no sea la operadora, a la Base de Datos B a los siguientes campos: Fecha, hora, nombre, teléfono, id, destino, duración.

	5	Se limpia todos los registros de la Base de Datos A
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si existió un error al verificar el estado de las BDs, se envía un correo al administrador y este caso de uso termina.
<b>Frecuencia esperada</b>	1 vez por semana o 1 vez por mes.	
<b>Comentarios</b>	Ninguno.	

RF- 03		Facturación
<b>Objetivos asociados</b>	OBJ-02 Crear y Enviar las Facturas a los Usuarios	
<b>Requisitos asociados</b>	RI-02 Información sobre socios	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Cronograma de Facturación lo solicite.	
<b>Precondición</b>	Ninguna	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Cronograma de Facturación inicia el proceso de Facturación
	2	Se realiza el caso de uso RF-02 (Paso de Llamadas de BD(A) a la BD(B)
	3	El sistema extrae los datos de la BD (B) con formato de salida XML para cada usuario.
	4	El sistema procesa cada archivo XML generado, el cual genera otro archivo XML.
	5	El sistema procesa los nuevos XML para generar cada archivo PDF (La Factura de cada Usuario).
	6	El sistema manda cada archivo PDF a los e-mails de los Usuarios
	7	Borra los archivos generados.
	8	Envía un correo al administrador con el archivo de los Usuarios que no tienen correo electrónico.
<b>Postcondición</b>	Ninguna	

<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si Falla el Caso de Uso RF-02, se envía un correo al administrador con el log de fallo y este caso de uso termina.
	5	Si el sistema detecta que los archivos generados no contienen llamadas, este las rellena con ceros.
	6	Si el Usuario no tiene correo electrónico, el sistema almacena el PDF en un archivo comprimido.
<b>Frecuencia esperada</b>	1 vez al Mes	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno.	

<b>RF- 04</b>		<b>Llamadas y No. de Llamadas</b>
<b>Objetivos asociados</b>	OBJ-03 Gestión de Información vía Web	
<b>Requisitos asociados</b>	RI-03 Información sobre el uso de los usuarios	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un Usuario o Admin solicite saber las llamadas que ha realizado.	
<b>Precondición</b>	El Usuario esta registrado en el sistema.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Usuario solicita al sistema comenzar el proceso de consulta de los datos de una llamada o llamadas.
	2	Se realiza el caso de uso RF-08 (Identificación de Usuario)
	3	El sistema solicita que ingrese la siguiente información para identificar la llamada o llamadas a mostrar: nº de Extensión, Id de Teléfono, Destino, Por Tipo, Por Mes, Por Año y opción de Cantidad de Llamadas.
	4	El sistema analiza los datos proporcionados por el Usuario.

	5	<p>El sistema muestra los siguientes datos correspondientes a la llamada:</p> <p>Para el Usuario: Fecha, Hora, Destino y duración.</p> <p>Para el Admin: Extensión, Nombre, email, Fecha, Hora, Id_telefono, destino y duración.</p> <p>Esta información esta dada por páginas de 20 elementos, además de navegación entre cada página.</p>
<b>Postcondición</b>	El sistema espera petición de navegación entre páginas o nueva consulta.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el caso de uso RF-08(identificación de Usuario) falla, se muestra una página de error.
	3	<p>Si el usuario no ingresa ninguna información, el sistema por default hará una consulta por el número de llamadas en el año activo.</p> <p>Si el usuario quitó el año por defecto (dejar la selección en la opción “por año”), el sistema hará una consulta de todas las llamadas registradas en Asterisk del usuario.</p>
<b>Frecuencia esperada</b>	1 vez/mes	
<b>Comentarios</b>	La frecuencia varia para el Administrador(es), ya que esto supone una carga de 1 o varias veces a la semana.	

<b>RF- 05</b>	<b>Por Quien llamo Más.</b>	
<b>Objetivos asociados</b>	OBJ-03 Gestión de Información vía Web	
<b>Requisitos asociados</b>	RI-03 Información sobre el uso de los usuarios	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un Admin solicite saber quién llamo más en una fecha.	
<b>Precondición</b>	Ninguna	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Admin solicita al sistema comenzar el proceso de consulta de los datos quién realizó mas llamadas en una fecha.

	2	Se realiza el caso de uso RF-08 (Identificación de Usuario)
	3	El sistema solicita que ingrese la siguiente información para identificar la llamada o llamadas a mostrar: Mes, Año.
	4	El sistema analiza los datos proporcionados por el Usuario.
	5	El sistema muestra los siguientes datos correspondientes a la llamada: Id_Telefono, extensión, Nombre, email, Año, Mes y Total de Llamadas.
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el caso de uso RF-08(identificación de Usuario), se muestra una pagina de error.
	3	Si el Admin deja los valores de selección “Por mes” y “Por año”, regresará el sistema quien hizo más llamadas en todos los años que ha registrado Asterisk.
<b>Frecuencia esperada</b>	1 vez/semana	
<b>Comentarios</b>	Ninguno.	

<b>RF- 06</b>		<b>Por Llamada de mayor duración</b>
<b>Objetivos asociados</b>	OBJ-03 Gestión de Información vía Web	
<b>Requisitos asociados</b>	RI-03 Información sobre el uso de los usuarios	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un Admin solicite saber la llamada de mayor duración.	
<b>Precondición</b>	Ninguna	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
	<b>Normal</b>	1

	2	Se realiza el caso de uso RF-08 (Identificación de Usuario)
	3	El sistema solicita que ingrese la siguiente información para identificar la llamada: nº de Extensión, Id de Teléfono, Destino, Por Tipo, Mes, día y Año.
	4	El sistema analiza los datos proporcionados por el Usuario.
	5	El sistema muestra los siguientes datos correspondientes a la llamada: Id_Telefono, Extensión, Nombre, email, destino, fecha, hora y duración.
<b>Postcondición</b>	El sistema espera petición de navegación entre páginas o nueva consulta.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el caso de uso RF-08(identificación de Usuario), se muestra una pagina de error.
	3	Si el Admin no ingresa ninguna información, el sistema por default hará una consulta por la llamada de mayor duración en el año activo.  Si el Admin quitó el año por defecto (deja la selección en la opción “por año”), el sistema hará una consulta de la llamada de mayor duración registrada en Asterisk.
<b>Frecuencia esperada</b>	1 vez/semana	
<b>Comentarios</b>	Ninguno.	

<b>RF- 07</b>		<b>Información de Usuarios</b>
<b>Objetivos asociados</b>	OBJ-03 Gestión de Información vía Web	
<b>Requisitos asociados</b>	RI-03 Información sobre el uso de los usuarios	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un Admin solicite saber información de los usuarios.	
<b>Precondición</b>	Ninguna	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>

<b>Normal</b>	1	El Admin solicita al sistema comenzar el proceso de consulta de los datos de los usuarios.
	2	Se realiza el caso de uso RF-08 (Identificación de Usuario)
	3	El sistema muestra los siguientes datos correspondientes a la consulta: Extensión, Teléfono, Nombre y email. El sistema despliega esta información en forma de páginas con 20 elementos navegables.
<b>Postcondición</b>	El sistema espera petición de navegación entre páginas o nueva consulta.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	En caso de que no halla información, el sistema no muestra ningún dato.
<b>Frecuencia esperada</b>	1 vez/semana	
<b>Comentarios</b>	Ninguno.	

<b>RF-08</b>		<b>Identificación de usuario.</b>
<b>Objetivos asociados</b>	· OBJ-03 Gestionar información de los usuarios vía Web	
<b>Requisitos asociados</b>	· RI-03 Información sobre el uso de los usuarios	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un socio solicite hacer un ingreso al sistema.	
<b>Precondición</b>	El usuario debe estar registrado.	
<b>Secuencia</b> <b>Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El Usuario solicita al sistema comenzar el proceso de ingreso a su cuenta.
	2	El sistema solicita que se identifique al Usuario y se indique su Extensión y password.
	3	El Usuario proporciona al sistema la Extensión y el password.
	4	El sistema registra el ingreso.

<b>Postcondición</b>	Ninguna.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Si el usuario ya esta identificado en el sistema, el caso de uso termina.
<b>Frecuencia esperada</b>	Frecuente.	
<b>Comentarios</b>	Cualquier operación el sistema siempre verifica que el usuario este identificado en el sistema.	

### 3.6 Diagrama de Clases

En el siguiente diagrama de clases se muestra como estas interactuarán en el sistema, las clases fueron detectadas de acuerdo a los casos de uso (ver el apartado 3.5), y se diseñó nuestro diagrama de clases (ver Figura 3.7).

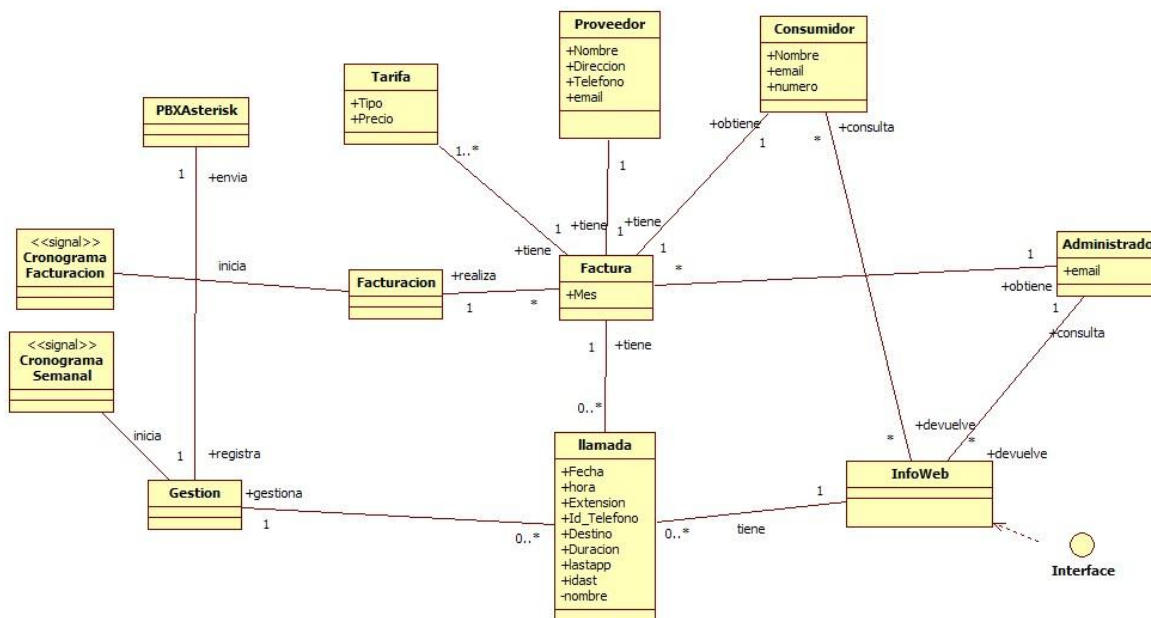


Figura 3.7 Diagrama de clases

#### 3.6.1 Diccionario de Datos

IPBX Asterisk: Este es el sistema de VoIP el cual manda los datos de llamadas al sistema (Gestión) para ser procesados. IPBX Asterisk envía la petición para el registro de las llamadas.

**Gestión:** La gestión sirve para procesar los CDR de Asterisk, además de trabajar con los datos que almacena el sistema. Gestión registra las llamadas del IPBX Asterisk. Además gestiona las llamadas registradas.

**Factura:** Una factura se denomina por medio del mes de cobro. La factura tiene como datos principales el proveedor del servicio y los datos del consumidor, además de tener la información de consumo, en este caso las llamadas realizadas.

**Proveedor:** El proveedor es la que ofrece el servicio VoIP. Este se identifica por el Nombre, la dirección, teléfono y email.

**Consumidor:** El consumidor es el usuario que utiliza el servicio VoIP, además de ser el que usa el servicio de consultas vía web.

**Administrador:** El administrador es el que mantiene un control de las llamadas que realiza cada usuario, así como de recibir las facturas de los usuarios sin email.

**Tarifa:** Las diferentes facturas tienen múltiples tarifas, variando según el tipo de llamada.

**InfoWeb:** La infoWeb es una interfaz que le permite al consumidor y al administrador obtener información acerca de las llamadas realizadas en Asterisk. Esta devuelve vía Web la información requerida por el usuario y el admin.

**Llamada:** La llamada es un registro del servicio de VoIP de Asterisk.

**Facturación:** Es el proceso de facturación el cual realiza las facturas (reporte) para cada Consumidor (Usuario).

En las tablas que se describirán a continuación, muestran la información que requiere Asterisk para guardar los CDR, además de las tablas propuestas para el sistema de Facturación:

<b>Tabla requerida por Asterisk para registrar los CDR (Call Data Record) (BDA)</b>				
Campo	Tipo	Longitud	Requerido	Descripción
calldate	Datetime		Si	Inicio de la llamada
callans	Datetime		No	Respuesta de la llamada
callend	Datetime		No	Fin de la llamada
clid	Varchar	80	Si	Caller ID con texto
src	Varchar	80	Si	Número del Caller ID

dst	Varchar	80	Si	Extensión del destinatario
dcontext	Varchar	80	Si	Contexto del Destinatario
channel	Varchar	80	Si	Canal usado
dstchannel	Varchar	80	Si	Canal del destinatario
lastapp	Varchar	80	Si	Última aplicación
lastdata	Varchar	80	Si	Última aplicación de datos
duration	Int	11	Si	Tiempo total de la llamada en segundos
billsec	Int	11	Si	Tiempo total desde que la llamada fue respondida. (en segundos)
disposition	Varchar	45	Si	Condición de la llamada: ANSWERED,NO ANSWER, BUSY, FAILED
amaflags	Int	11	Si	Banderas de uso de Asterisk,_DOCUMENTATION, BILLING, IGNORE etc, especificadas para los canales.
accountcode	Varchar	20	Si	Que número de cuenta usa
Userfield	Varchar	255	No	Un campo para el usuario si este esta definido.
uniqueid	Varchar	32	Si	Identificador único

Nota: el src, es el cid\_number del archivo de configuraciones de Asterisk (user.cfg).

**Tabla propuesta para el registro de llamadas del sistema de Facturación (BDB)**

Campo	Tipo	Longitud	Requerido	Descripción
fecha	date		Si	La fecha de la llamada
hora	time		Si	La hora de la llamada
nombre	varchar	80	Si	Usando el clid de la

### 3.6 Diagramas de Clases

				tabla de Asterisk,
teléfono	varchar	80	Si	Identificación del usuario (Caller ID)
id	varchar	80	Si	Extensión, obtenida de channel de la tabla de Asterisk
destino	varchar	80	Si	Extensión destino.
duración	int	11	Si	Duración de billing, obtenida del campo billsec de Asterisk
idast	varchar	32	Si	Identificación única obtenida de uniqueid de la tabla de Asterisk
lastapp	varchar	80	Si	Última aplicación de la llamada, en este caso se consideró Voicemail y Dial

#### Tabla propuesta para el control de usuarios del sistema de Facturación (BDB)

Campo	Tipo	Longitud	Requerido	Descripción
id	varchar	80	Si	Extensión a la que esta ligado el usuario
cid_number	varchar	11	Si	Número de identificación del usuario
fullname	varchar	80	Si	Nombre Completo
secret	varchar	30	Si	Contraseña del usuario
email	varchar	80	Si	Correo electrónico del usuario

Nota: esta tabla no esta ligada al archivo de configuración de usuarios de Asterisk, por lo que si es agregado un usuario nuevo, se tendrá que agregar manualmente en esta tabla.

### 3.7 Escenarios del Sistema

En este apartado se desglosarán algunos escenarios del sistema. Estos ayudan a desarrollar una mejor descripción del problema para especificar información del modelo, para asegurar que los pasos importantes no sean omitidos, y que el flujo general del sistema sea correcto.

#### I. Escenario para proceso de Gestión.

1. El proceso de Gestión espera a ser iniciado por el cronograma (Cronograma Semanal) o por el proceso de Facturación.
2. El proceso de Gestión al ser iniciado verifica los directorios de trabajo.
3. El proceso de Gestión verifica el estado de las Base de Datos A y B.
4. El proceso de Gestión realiza respaldos de las Bases de Datos A y B.
5. El proceso de Gestión borra los archivos temporales generados en el proceso de respaldo.
6. El proceso de Gestión copia y procesa los registros de llamadas de la Base de Datos A a la Base de Datos B.
7. El proceso de Gestión vacía los registros de llamadas de la Base de Datos A.
8. El proceso de Gestión limpia archivos temporales generados en los procesos anteriores.

#### II. Escenario para proceso de Facturación

1. El proceso de Facturación espera a ser iniciado por el cronograma (Cronograma Facturación).
2. El proceso de Facturación inicia el proceso de Gestión.
3. El proceso de Facturación inicia el proceso de Creación y Envío de Facturas PDF.
4. El proceso de Facturación envía las facturas de usuarios sin email al administrador en un archivo comprimido si este existe.
5. El proceso de Facturación limpia archivos temporales generados en los procesos anteriores.

#### III. Escenario para el proceso de Creación y Envío de Facturas

1. Se sitúa en el directorio de trabajo.
2. Registra la fecha en que se realiza la operación en un archivo.
3. Obtener la información de los usuarios de la Base de Datos B.
4. Obtener la cantidad de llamadas para cada tipo de marcación.
5. Obtener los datos de las llamadas realizadas para cada uno de los usuarios.
6. Procesar los datos para crear la factura (reporte)
7. Enviar la factura (reporte) a cada usuario.

### 3.8 Diagramas de Secuencia

Los diagramas de secuencia son utilizados para mostrar la interacción que existe dentro del sistema. Provee un mapa secuencial de pasos de mensaje entre objetos a través del tiempo. A continuación se presentan los diagramas de secuencia para algunos de los casos de uso descritos.

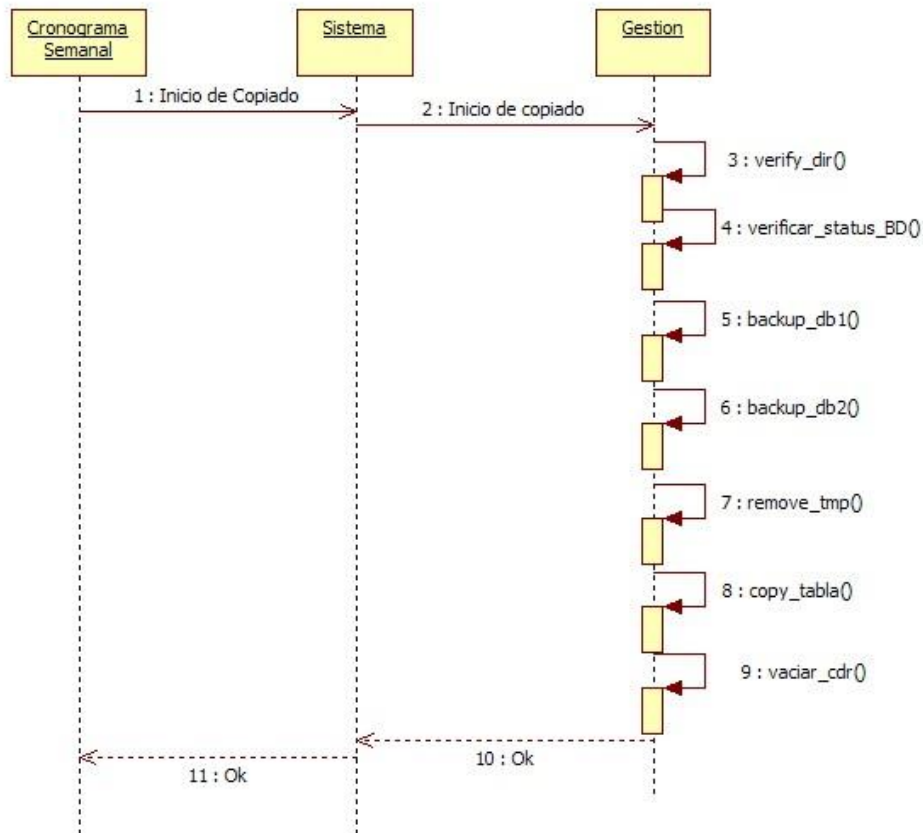


Figura 3.7 Diagrama de Secuencia para el Cronograma Semanal.

En la figura 3.7 se da una vista del escenario para el proceso de Gestión descrito en la sección 3.7.

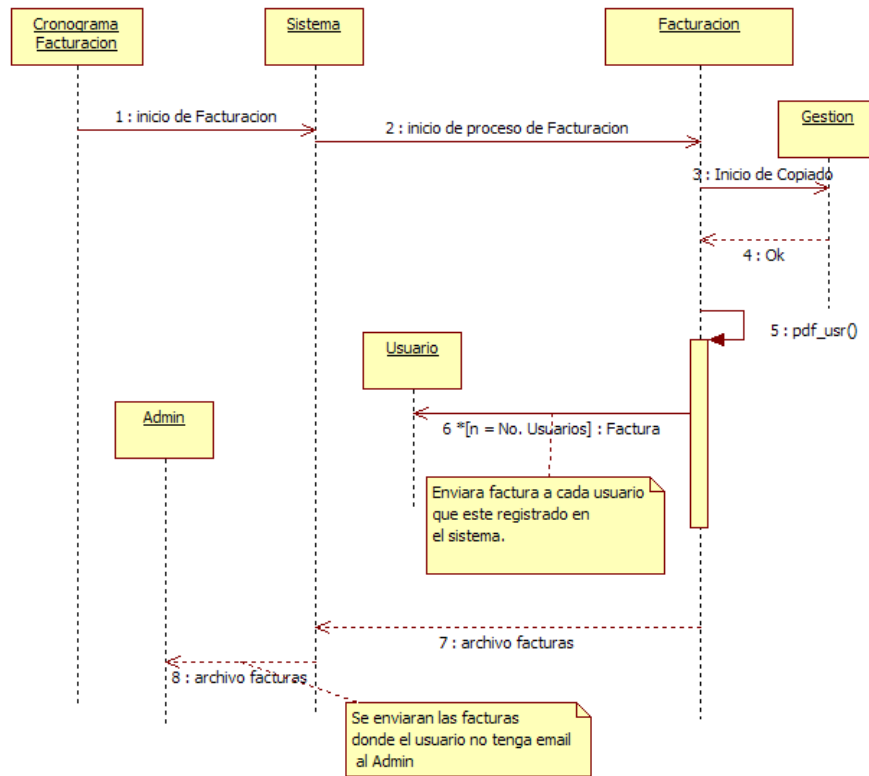


Figura 3.8 Diagrama de Secuencia para el Cronograma de Facturación.

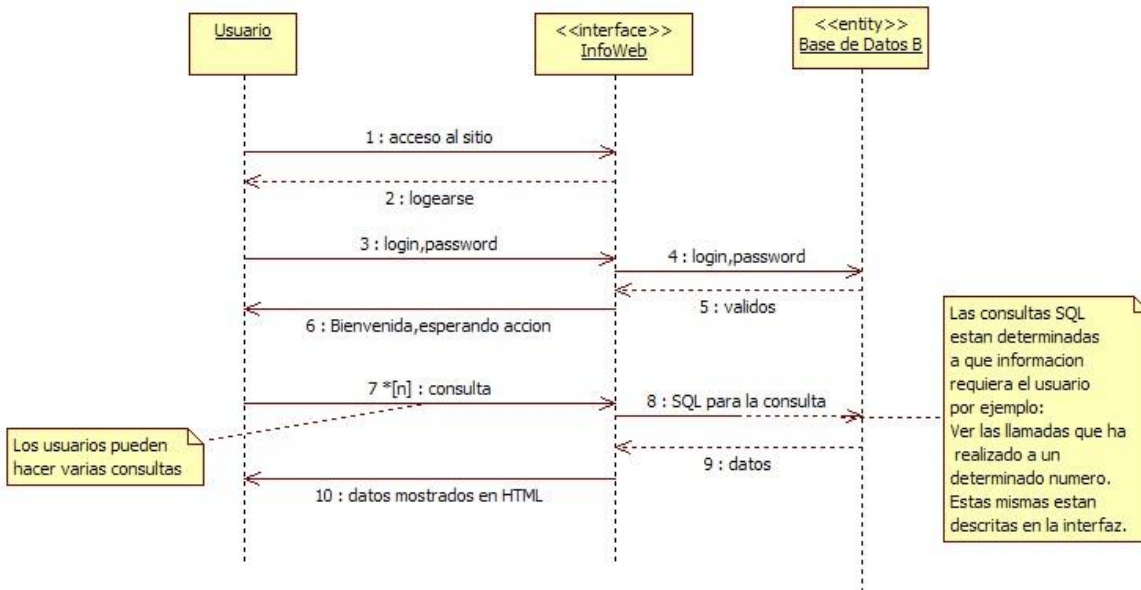


Figura 3.9 Diagrama de secuencia para las consultas Vía Web

En la figura 3.9 se muestra la forma en que un Usuario o Admin realiza una consulta al sistema. Este diagrama es un caso general para los casos de uso RF-04, RF-05, RF-06 y RF-07 descritos anteriormente.

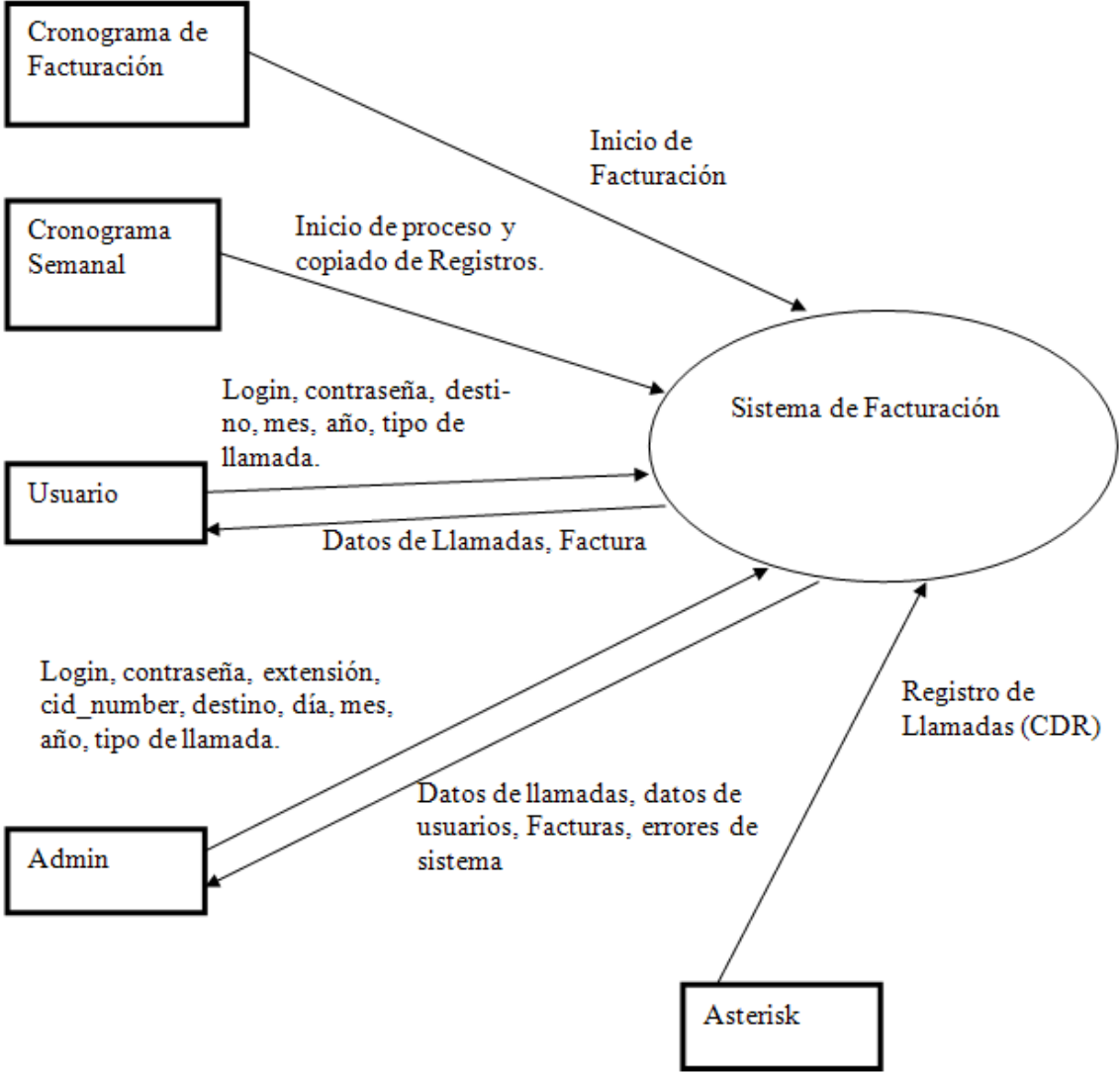


Figura 3.10 Diagrama de flujo para el Sistema de Facturación.

En la figura 3.10 Se muestra el sistema en general, los datos de entrada y salida, así como los componentes que interactúan con el sistema

## CAPÍTULO 4

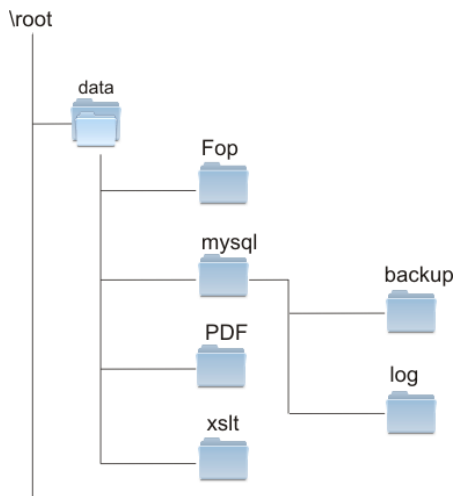
# IMPLEMENTACIÓN

### 4.1 Introducción

En este capítulo describiremos la forma en que se fue implementando el sistema conforme a lo descrito en el capítulo anterior. Además de exponer alguno de los problemas que se observaron al realizar este proyecto y las soluciones que se optaron.

### 4.2 Estructura de los directorios del proyecto.

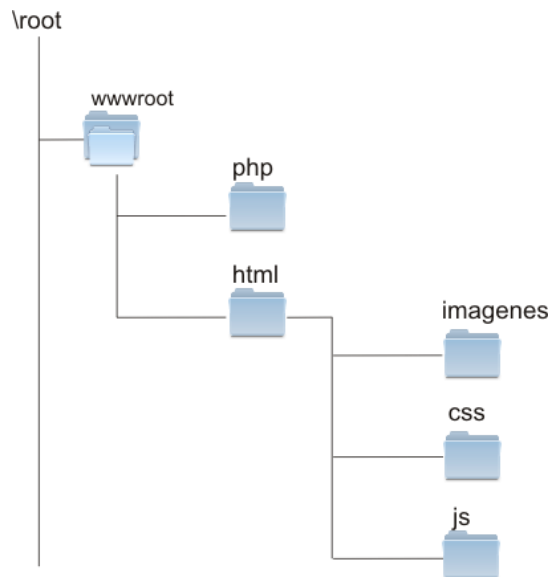
La forma en que se manejaron los directorios de trabajo se muestran en las figuras 4.1 y 4.2 respectivamente.



**Figura 4.1 Directorios de Trabajo para los Shell scripts**

Como se muestra en la figura 4.1 los directorios que van a contener los archivos requeridos para el sistema son Fop, donde se alojará la aplicación para el tratamiento de los archivos XML y XSL-FO, el directorio XSLT, donde se encontrarán las plantillas para procesar los archivos XML. Los demás directorios contendrán los archivos generados por el sistema.

En el caso del sistema de consultas (fig. 4.2), el directorio html, será definido a la hora de instalación, este depende del servidor web y de donde se quiera poner la aplicación. También cabe mencionar que el directorio PHP, será restringido el acceso desde la red, por lo que este directorio debe ser configurado desde el servidor. En este caso en el archivo de configuración de PHP, en la sección de directorios include.



**Figura 4.2 Directorio de Trabajo para las consultas vía Web**

### 4.3 Herramientas usadas.

A continuación se describirán algunas herramientas utilizadas en el proyecto.

- *Fop*. Apache FOP (Formatting Objects Processor) API desarrollada por el proyecto Apache que permite la generación de los PDFs en línea de comandos o mediante llamadas al API.
- *xsltproc*. Es un procesador XSLT de línea de comandos creado por Daniel Veillard. Es parte de la librería GNOME XSLT (libxslt).
- *mailx*. Es un programa que se encuentra en varios sistemas operativos Unix-like para enviar y recibir correo, también conocido como programa MUA. Es una versión mejorada del programa "mail" de Unix.
- *mysql*. MySQL es un sistema de administración de bases de datos. Se uso el CLI (command line interface) para realizar las consultas.
- *mysqldump*. El comando mysqldump permite realizar copias de seguridad (Backup) del gestor de base de datos MySQL.

- *awk*. Es un lenguaje de programación diseñado para procesar datos basados en texto, ya sean ficheros o flujos de datos.
- *sed*. Es un editor de emisiones (stream editor) utilizado para el procesamiento de texto en ficheros. Utiliza un lenguaje de programación para realizar transformaciones en una emisión de datos leyendo línea por línea de estos. Fue desarrollado entre 1973 y 1974 por Lee E. McMahon de Bell Labs. Está incluido en las instalaciones básicas de prácticamente todas las distribuciones de GNU/Linux
- *tar*. Es usado para almacenar archivos y directorios en un solo archivo. Dentro de los entornos Unix tar aparece como una orden que puede ser ejecutada desde la línea de órdenes de una consola de texto o desde un simple terminal.
- *cat*. El comando **cat** (por concatenar) es un programa de Unix usado para concatenar y mostrar archivos.
- *date*. El comando date muestra la hora y fecha del sistema.

### 4.4 Subsistema de Gestión.

En este módulo, el sistema realiza las operaciones de registro de los CDR de Asterisk, además de procesar los datos entre las tablas de las base de datos A y B, además de otros procesos. Este módulo se realizó por medio de Shell script.

A continuación se da una breve explicación del script:

Se definen algunas variables:

- I. Usuario y password de las bases de datos.
- II. Hora y Fecha para registrar en los logs del sistema.
- III. Los directorios de trabajo.
- IV. Nombres de los logs basados en las variables de fecha y hora.
- V. Nombres de los archivos que se le enviaran al administrador.

También se definen las funciones que harán las operaciones entre las bases de datos, las cuales son:

- VI. `copy_tabla( )` : esta copia los registros de la Base de Datos A a la Base de Datos B.
- VII. `vaciar_cdr( )` : elimina los registros de la tabla cdr de la Base de Datos A.
- VIII. `verify_dir( )` : verifica los directorios de trabajo, donde se crearán archivos temporales y los respaldos de las bases de datos, así como los directorios donde se registran las

operaciones del sistema. En dado caso que faltara un directorio, esta función lo crearía.

- IX. verificar\_status\_BD( ): La acción de esta función es verificar que las Bases de Datos estén activas y funcionando.
- X. backup\_db1( ): esta función respalda la base de datos A.
- XI. backup\_db2( ): esta función respalda la base de datos B.
- XII. remove\_tmp( ): esta función elimina archivos temporales.

El flujo principal del programa queda en el siguiente código:

```
verify_dir
verificar_status_BD
bstatus_bd=$?;
if [ $bstatus_bd -eq 0 ]
then
echo "Respaldo Base de datos $NAME_DB1" >> $LOG;
backup_db1
backup_db2
remove_tmp
echo "{+}Realizando copia de las llamadas Dial y Vozmail" >> $LOG;
copy_tabla
echo "{+}Vacando los registros de Call Data Record de asterisk" >> $LOG;
vaciar_cdr
echo "{+}Aca ira la parte para crear pdfs" >> $LOG;
else
#echo "Error al consultar las bases de datos";
echo "{+}Enviando correo al administrador" >> $LOG;
mailx -s VoIP admin ${EMAIL_ADMIN} -a $LOG < ${DIR_MAIL}mailerr.txt
fi
```

#### 4.5 Subsistema de Facturación

En este módulo, el sistema creará las facturas (reporte) para cada usuario para enviárselos a sus correos electrónicos. A continuación se da una breve descripción de las funciones requeridas así como algunas herramientas que se usaron para la elaboración de estas facturas en archivos PDF.

Se definen algunas variables:

- I. Usuario y password de las base de datos.
- II. Hora y Fecha para registrar en los logs del sistema y para seleccionar el mes de cobro.
- III. Los directorios de trabajo.
- IV. Nombres de los logs basados en las variables de fecha y hora.

v. Nombres de los archivos que se le enviarán al administrador.

También se definen las funciones que harán las operaciones para generar las facturas (reporte):

- I. pdf\_usr( ): Función principal del proceso de facturación, la cual llama a otras funciones que realizan operaciones en las bases de datos y programas para procesar los datos obtenidos. En la siguiente lista se muestran las acciones que realiza este proceso:
  - 1) Se ubica en el directorio de trabajo "PDF"
  - 2) Inicia una variable en la que registra la hora del proceso
  - 3) Crea dos arreglos en los cuales se obtiene la extensión y el id de los usuarios, además de obtener el número de usuarios.
  - 4) En un ciclo manda a llamar las funciones que harán las operaciones de obtención de datos y procesamiento de estos para realizar la factura, además del envío respectivo a su email de cada usuario. El número de iteraciones es igual al número de usuarios.
  - 5) Inicia una variable en la que registra la hora en que finalizo el proceso
  - 6) Calcula el tiempo y lo anexa al log del sistema.
- II. get\_llamadas( ): Esta función realiza una extracción de datos a la Base de Datos B, específicamente a la tabla *llamadas*, estos datos son guardados en formato XML, para posteriormente ser procesados.
- III. get\_datos( ): Esta función realiza una extracción de datos de cada usuario, y la guarda en un archivo XML.
- IV. procesar\_xml( ): Esta función procesa todos los archivos XML, obtenidos de las funciones antes descritas, para obtener los archivos PDF que serán enviados después.
- v. enviar\_pdf( ) : Esta función tiene la acción de enviar los archivos PDF obtenidos. Antes de enviar los archivos recupera los emails de cada usuario, si este tiene email, envía el archivo PDF correspondiente a este usuario, en caso contrario, el PDF es empaquetado para después enviarlo al administrador del sistema.

El flujo principal del proceso queda en el siguiente código:

```

cd ${DIR_PDF}
tiempo_inicio=`date +%s`;
echo "Inicio : `date +%H:%M:%S`" >> $LOG;
echo "<mes>${MONTH_BILL_NAME}</mes>" > mes.xml;
varcid=`mysql -u ${USR_DB1} -p${PWD_DB1} ${NAME_DB2} --batch -e "SELECT cid_number
FROM usuarios" -s`;
varid=`mysql -u ${USR_DB1} -p${PWD_DB1} ${NAME_DB2} --batch -e "SELECT id FROM usuarios"
-s`;
len=`mysql -u ${USR_DB1} -p${PWD_DB1} ${NAME_DB2} --batch -e "SELECT
COUNT(cid_number) FROM usuarios" -s`;
for(( i = 1; i <= len; i++ ))
do
#Obtenemos los datos que vamos a generar para cada usuario.
cid_number[$i]=`echo $varcid | awk '{print substr($i,1,length($i))}'`;
id[$i]=`echo $varid | awk '{print substr($i,1,length($i))}'`;
#echo ${cid_number[$i]},${id[$i]} Opción para verificar que si lo estuviera haciendo correctamente.
get_llamadas ${cid_number[$i]} ${id[$i]};
get_datos ${cid_number[$i]} ${id[$i]};
procesar_xml ${cid_number[$i]} ${id[$i]};
enviar_pdf ${cid_number[$i]} ${id[$i]};
done
tiempo_fin=`date +%s`;
let tiempo_total=tiempo_fin-tiempo_inicio;
rm -f mes.xml;
echo "{+}Tiempo tomado para crear los pdfs y enviarlos= $tiempo_total segundos" >> $LOG;
echo "{+}Fin : `date +%H:%M:%S`">>$LOG;

```

En el proceso de facturación se tuvieron los siguientes problemas:

- 1) MSQL tiene la opción de que puede generar los archivos XML a partir de una consulta, el inconveniente de esta forma de obtener los datos es que son difíciles de tratar. Por lo que la solución a este problema fue:
  - a) Con ayuda de una plantilla XSLT, se procesaron los datos para poderlos tratar mas sencillamente.

```

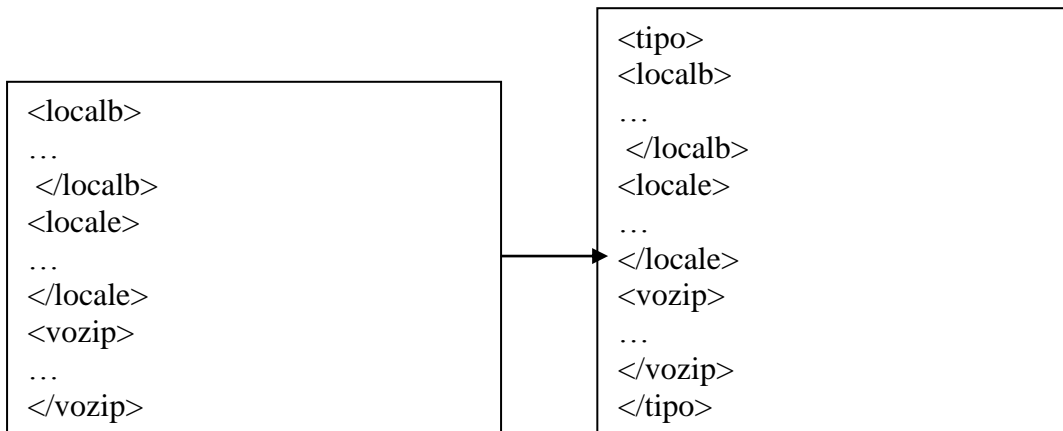
<row>
  <field name="telefono">7205</field>
  <field name="src">600</field>
  <field name="fecha">May-28</field>
  <field name="hora">05:35</field>
  <field name="duracion">25</field>
</row>

```

```
<localb>
  <telefono>7205</telefono>
  <src>600</src>
  <fecha>May-28</fecha>
  <hora>05:35</hora>
  <duracion>25</duracion>
</localb>
```

b) A cada archivo generado por MYSQL, se usó una plantilla diferente.

- 2) Como en el anterior procedimiento se generaban varios archivos XML los cuales, tenían algunas coincidencias como `<?xml version="1.0" encoding="UTF-8"?>`, `<resultset>`, `</resultset>`, se optó por usar la herramienta "sed", el cual nos permitía eliminar estas coincidencias, además de insertar texto al principio y al final de los archivos XML. Por ejemplo:



- 3) Como había varios documentos XML procesados, se tenían que unir para poderlos tratar con una plantilla del tipo XSL-FO y crear con esto el documento PDF, por lo que se usó la herramienta "cat" para unir los archivos.
- 4) Los números que están guardados en la tabla *llamadas* traen junto al número los dígitos de marcación, esto es por que pueden haber distintos tipos de marcación y el tamaño de números puede ser igual a cada tipo de marcación entonces es necesario que el número tenga los dígitos de marcación, por lo que en una factura no deben verse esos dígitos, la solución a esto se optó por las funciones internas de MYSQL. Por ejemplo:

Numero	Función	Resultado
97205	SUBSTRING(destino,2,5)	7205
998884059	SUBSTRING(destino,3,7)	8884059
882345678	SUBSTRING(destino,3,7)	2345678

En el ejemplo podemos ver los dígitos 88 al principio del número, este tipo de marcación actualmente en el sistema no existe, pero da un claro ejemplo de la importancia de mantener los dígitos de marcación en los registros, para poder diferenciar los números telefónicos a la hora de realizar las facturas y la información que se consulta.

- 5) Cuando el proceso de facturación se llamaba desde el crontab, este no respetaba la ruta de JAVA para que el procesador de archivos XSL-FO funcionara correctamente, por lo que se tuvo que definir la ruta en el script, usando la siguiente sentencia:

```
export JAVA_HOME=/usr/lib/jvm/sun-java-5.0u15/jre
```

**4.5.1 La factura (reporte)**

En la siguiente figura se mostrará la factura que recibirá cada usuario y la explicación de la misma:



**Benemérita Universidad Autónoma de Puebla**

**Proveedor:**

Facultad de Ciencias de la Computación, Servicio de VoIP, Av. San Claudio y 14 Sur Ciudad Universitaria C.P. 72570 Puebla, Puebla. México.

**Usuario:**

Manuel Martin I Ortiz  
132  
martin@cs.buap.mx

**Estado de Cuenta:**

**Mes: November**

Datos	Informacion
Locales BUAP:	16
Locales Casa:	2
Voip:	2

**Figura 4.3 Página de Datos de la Factura**

En la figura 4.3 se muestra la primera página del PDF enviado a los usuarios, en la cual se observa:

- I. Información del Proveedor
- II. Información del Usuario
- III. Mes de Factura
- IV. El Estado de Cuenta
  - a) Llamadas Locales BUAP y el total de llamadas
  - b) Llamadas Locales a Casa y el total de llamadas
  - c) Llamadas VoIP y el total de llamadas

En la figura 4.4 Se muestran los detalles de cada tipo de llamada, en las que se observamos la fecha, hora, el destino y la duración de cada una mostrada en segundos, divididas por el tipo de marcación:

- a) VoIP ( 3 dígitos)
- b) Locales BUAP ( 4 dígitos)
- c) Locales a teléfono fijo ( 7 dígitos)

También se puede ver el encabezado y pie de página. En el encabezado se opto por poner el nombre de la Universidad u organización. En el pie de página se pone información para contactar al proveedor del servicio, en este caso la Facultad de Computación. Además de poner un link a la página donde se puede consultar las llamadas realizadas.

Cabe mencionar algunos detalles en las facturas, ¿Qué pasa cuando el usuario no usa el servicio?, ¿Por qué en algunas facturas aparecen con 0 segundos? o que la duración exceda mas de 2 horas (mas de 7200 segundos).

Cuando un usuario no usa el servicio de VoIP simplemente el sistema pone a 0 la factura, aunque esto debería desaparecer mas adelante, con la incentiva de usar este servicio para el beneficio de la Facultad.



## Benemérita Universidad Autónoma de Puebla

Version 1.0

Vozip:			
fecha	hora	telefono	duracion(seg)
November-25	12:24	911	28756
November-25	14:48	132	53
Locales(9):			
fecha	hora	telefono	duracion(seg)
November-6	11:09	5564	183
November-11	11:59	7203	48
November-12	10:05	5119	1142
November-13	12:53	7207	336
November-13	13:56	5939	195
November-14	12:54	7300	147
November-20	13:59	5240	66
November-20	14:12	2452	2
November-20	14:14	2452	2
November-20	14:16	5573	150
November-25	14:46	5389	2214
November-25	14:48	5389	43
November-25	14:50	5389	20
November-25	14:51	5242	26
November-25	14:52	5389	49
November-25	16:09	5723	15
Locales Externas (99):			
fecha	hora	telefono	duracion(seg)
November-4	13:15	2663100	133
November-4	13:18	2663100	285

FCC VoIP - [phone.cs.buap.mx](mailto:phone.cs.buap.mx) - email to: [phonevoip@phone.cs.buap.mx](mailto:phonevoip@phone.cs.buap.mx) - Teléfono :  
 +52 (222) 2 229 55 00 Recepción Extensión 7200 Fax : +52 (222) 2 229 56 72

**Figura 4.4 Detalles de las llamadas de la factura**

En la figura 4.5 se muestra un claro ejemplo cuando una llamada excede mas de 2 horas, en este caso se muestra una llamada de casi 8 horas (28756 segundos), esto da a pensar dos cosas:

- 1) El usuario no colgó correctamente
- 2) El usuario realmente llama demasiado

En costos de la llamada esta sería excesiva para una organización, ya que como se sabe las empresas telefónicas por lo general cobran por minuto, por lo que 8 horas sería una cantidad considerable.

<b>Vozip:</b>			
fecha	hora	telefono	duracion(seg)
November-25	12:24	911	28756
November-25	14:48	132	53

**Figura 4.5 Caso en la que excede la llamada mas de 2 horas.**

En la figura 4.6 se muestra el caso en la que el sistema marca como duración 0 segundos. Comúnmente esto no tendría sentido, la cual pasaría a reclamos de parte del usuario, pero tomando en cuenta las configuraciones y los tiempos de respuesta de los sistemas, pongamos un ejemplo:

- Supongamos que un usuario marca a un número, espera 3 tonos para que le contesten, al momento en que el usuario que llama va a colgar, en ese preciso instante la persona a la que llamó contesta, por lo que el instante en que se abrió la comunicación se registró y es por eso que esa fracción de segundo se registra.

Esto no se puede evitar ya que este tipo de registros se tienen que configurar en el sistema Asterisk y ser restringidos, para que estos registros sean considerados como una llamada completada. Podría decirse que mínimo sea una llamada de 5 segundos.

May-13	15:19	7208	10
May-14	08:12	7270	30
May-14	08:12	7270	0
May-14	09:38	7270	20

**Figura 4.6 Caso en que aparece duración de 0 segundos.**

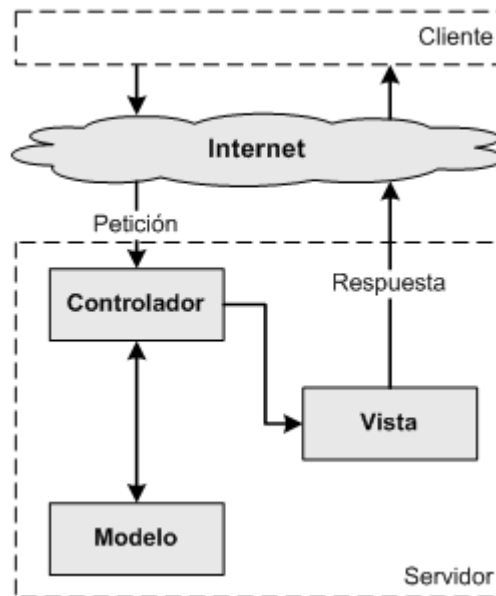
También este tipo de situación pasa por la forma en que administra Asterisk la comunicación entre usuarios.

#### **4.6 Subsistema de Información Vía Web**

El sistema se realizó basado en la arquitectura *Modelo Vista Controlador* (MVC), que esta formado por tres niveles:

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La Figura 4.7 ilustra el funcionamiento del patrón MVC.



**Figura 4.7 El patrón MVC**

Ahora se describirán algunos de los archivos utilizados en el sistema:

Archivo	Descripción
index.php	Este es la página principal y se encarga de incluir los procesos de interacción, acceso y funciones requeridas para las consultas, así como el de la vista para el usuario.
login.php	Página para el acceso al sitio, el cual requiere login y password, además de iniciar sesiones dentro del sistema, para el control del usuario.
paginador.php	Este se encarga de controlar la paginación de resultados mayores de 20 registros.
vista.php	Se encarga de manejar la vista hacia el usuario, dependiendo si es Administrador o usuario normal.
salir.php	Se encarga de cerrar las sesiones con el servidor

accesscontrol.php	Se encarga de verificar si el usuario esta logeado en el sistema.
func_db.php	Funciones para la interacción con el administrador de bases de datos (MySQL)
func_v.php	Funciones para registrar variables en el servidor web, así como para generar algunas consultas (SQL)
vardef.php	Definición de algunas variables para la administración y uso del sistema. Como por ejemplo: el día en que se instalo el sistema, o los passwords para la base de datos o el admin.
divConsultaadmin.php divConsultausr.php	Son complementos para vista.php, los cuales muestran las opciones de consulta para cada usuario.
consulta2.php consulta3.php consultainfoursr.php	Son las distintas consultas que están disponibles en el sistema, además que manejan los datos obtenidos de estas.

A continuación mencionare algunas observaciones en el transcurso de la implementación de las consultas:

- 1) Cuando el admin realizaba una consulta en la que hubiera resultados mayores a 50 datos, había que dividir esos datos para mostrarlos en varias páginas, por lo que se optó a realizar un modulo de paginación, en un principio este se había realizado con paso de variables. Esta forma de enviar variables, no era muy práctica, por lo que se usaron variables de sesión, las cuales permitían suprimir el envío de tantas variables entre el usuario y el servidor, y nada mas enviar una sola variable la cual indica la página que quiere ver. En este caso se hizo un paso de páginas de uno en uno, es decir, simplemente se pusieron dos botones de avance y retroceso, además para ir a la primera página y a la última página de datos.
- 2) Para la realización del código SQL para las consultas, se hicieron dos tipos:

- a) Por medio de una función en PHP. En este tipo de función se evaluaban las variables de entrada (extensión, tipo, año, etc.) y se iba construyendo el código SQL para realizar la consulta.
  - b) Por medio de una función en MYSQL En este tipo de función se evalúan las variables desde MYSQL, por medio de funciones de usuario.
- 3) Un detalle por el cual se define el año de instalación, es por que, cada vez que pasa un año, la interfaz debe proporcionar la opción de este nuevo año, por lo que se uso esta variable junto con una función en PHP para generar los años entre el año de instalación y el año actual.
- 4) Las consultas disponibles a continuación se despliegan:
- I. Consulta por CALL ID y/o Extensión al que se marco. En este tipo de consulta se obtiene las llamadas realizadas por el Usuario, ingresando su CALL ID o extensión ligada a este.
  - II. Consulta por Destino. En esta consulta se obtiene las llamadas realizadas al número ingresado.
  - III. Consulta por año y mes. En esta consulta se obtienen las llamadas realizadas en una fecha en específico.
  - IV. Consultas por tipo de llamada (Marcación: VoIP, Interna BUAP, a teléfono fijo).
  - V. Consulta para la información de los Usuarios
  - VI. Consulta del Usuario que hizo la llamada mas larga a partir del año y/o mes.
  - VII. Consulta del Usuario que realizó mas llamadas a partir del año y/o mes.
  - VIII. Combinación entre las primeras 4 consultas, por ejemplo: Obtener las llamadas del usuario con CALL ID: 132 en el mes de Septiembre del año 2008. También podría ser: Obtener las llamadas realizadas a números particulares, en el mes de Abril.

En las figuras siguientes se muestran los diseños de las interfaces mostradas al Usuario y al Admin.

Nombre: Hansel Alva  
Número: 600  
Id\_usr: 600  
Email: hansel\_98@hotmail.com

Salir

Seleccione el tipo de consulta:  
 Consultas Por:  
 Por Número   
 Por Tipo   
 Por mes  2009

Fecha	Hora	destino	duración (seg)
2008-06-02	13:21:58	850	93
2008-06-03	13:56:10	7205	4
2008-06-04	15:07:47	132	26
2008-06-04	15:08:54	132	5
2008-06-04	15:29:42	2456061	24
2008-06-04	15:29:52	2456061	13
2008-06-04	15:30:33	2456061	32
2008-06-06	15:32:54	4664067	9
2008-06-12	12:23:45	7200	6
2008-06-12	12:28:02	602	18
2008-06-12	12:30:37	911	5
2008-06-12	12:57:43	4664067	12
2008-06-25	17:53:29	911	5
2008-06-29	10:46:36	2456061	5
2008-06-29	10:47:10	2456061	3

Copyright Facultad de Ciencias de la Computación VoIP  
 FCC VoIP - phone.cs.buap.mx  
 - email to: phonevoip@phone.cs.buap.mx  
 - Telefono : +52 (222) 2 229 55 00 Recepción Extensión 7200 Fax : +52 (222) 2 229 56 72

Figura 4.8 Interfaz para el Usuario.

Nombre: administrador  
Email: admin@phone.cs.buap.mx

Salir

Seleccione el tipo de consulta:  
 Consultas Por:  
 Consultas Quien mas llamo:  
 Consultas Duración:  
 Información de Usuarios:

Fecha	Hora	CALL ID	destino	duración (seg)
2008-09-01	09:16:51	911	306	84
2008-09-01	09:25:39	306	7270	169
2008-09-01	10:03:53	306	6218	132
2008-09-01	10:07:23	306	7404	20
2008-09-01	10:08:11	306	7406	113
2008-09-01	10:51:24	911	306	9
2008-09-01	10:51:33	911	306	8
2008-09-01	10:53:16	911	306	72
2008-09-01	11:07:19	306	911	91
2008-09-01	12:36:34	153	5915052	28
2008-09-01	12:37:27	153	5915052	31
2008-09-01	12:38:38	153	5915052	22
2008-09-01	13:43:56	142	7204	26
2008-09-01	16:32:02	306	7907	86
2008-09-01	17:33:13	160	2335272	57
2008-09-01	17:41:02	160	2944284	68
2008-09-01	18:15:00	306	5353	193
2008-09-01	20:28:09	306	7349	86
2008-09-02	09:33:02	302	2663100	73
2008-09-02	11:14:25	302	2663100	85

Primero Pagina 1/13 Siguiente Ultimo

Copyright Facultad de Ciencias de la Computación VoIP  
 FCC VoIP - phone.cs.buap.mx  
 - email to: phonevoip@phone.cs.buap.mx  
 - Telefono : +52 (222) 2 229 55 00 Recepción Extensión 7200 Fax : +52 (222) 2 229 56 72

Figura. 4.9 Interfaz para el Admin.

En las figuras 4.8 y 4.9 se muestran las interfaces para el Usuario y el Admin, como se puede apreciar las consultas se encuentran del lado izquierdo, los datos obtenidos a través de estas consultas están en el lado derecho desplegadas como una tabla de datos. Además se puede observar en la cabecera se indica los datos del Usuario. Y en el pie de pagina los datos de la Facultad de Computación (responsable del servicio VoIP). La figura 4.10 desglosa uno de los menús de consulta donde se aprecian varios campos de texto y de selección, además de peculiares iconos con signo de interrogación (?), que al poner el cursor encima de este, se despliega una nube de texto, la cual da una explicación de la opción.

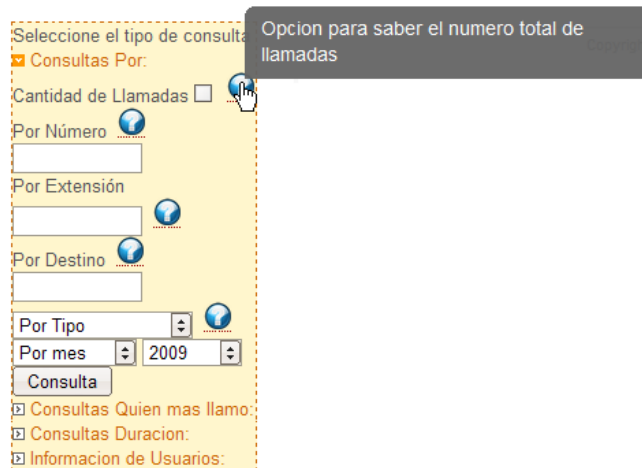


Figura 4.10 Opciones de Consulta con un ejemplo de icono de ayuda



Figura 4.11 Algunas Opciones de consulta para el Admin

**4.7 Pruebas y Resultados**

En la siguiente sección se describen algunos resultados obtenidos de la aplicación, en particular del subsistema de información vía Web. Estos resultados son obtenidos a partir de las opciones que nos proporciona la aplicación. En la tabla 4.1 se escogieron 8 números para realizar las pruebas, usando para estos primeros resultados las opciones de “Consulta por Duración” y “Consulta para ver llamadas”, así como “Información de Usuarios”, para obtener los datos del número.

<b>CALL ID</b>	<b>Nombre</b>	<b>Mes-Año</b>	<b>Llamada de mas duración (número y duración)</b>	<b>No. de llamadas del mes</b>
911	Gustavo Cossio	Ene-09	"5327", 1878 segundos	24
306	Biblioteca	Ene-09	"5361", 1273 segundos	34
302	Bárbara Sánchez	Ene-09	"2873421", 605 segundos	25
132	Manuel Ortiz	Ene-09	"7205", 561 segundos	28
144	Olivia Romero	Ene-09	"1531646", 581 segundos	2
161	Centro de Calidad FCC	Ene-09	"2335272", 343 segundos	10
142	Eduardo Pinto	Ene-09	"5728", 170 segundos	5
128	Jesús Lavalle	Ene-09	"4548187", 315 segundos	27

**Tabla 4.1 Resultados para consultas por llamadas y duración**

Como se ve en la tabla 4.1 para cada usuario se obtuvo la llamada que hizo de mayor duración, además del número de llamadas que realizo en ese mes. En la tabla 4.2 la información que se muestra es el número de llamadas registradas en cada mes, además de dar el número de extensión con mayores llamadas realizadas en esa misma fecha.

<b>Mes-año</b>	<b>Total de llamadas registradas</b>	<b>Usuario con mayor número de llamadas</b>	<b>No. De llamadas</b>
ago-08	285	604	79
sep-08	259	302	81
oct-08	252	306	83
nov-08	236	302	111
dic-08	91	306	34
ene-09	202	306	34
feb-09	240	306	76
mar-09	394	302	95
abr-09	133	306	42
may-09	207	306	69

**Tabla 4.2 Resultados para consultas por quién llamo más y por # de llamadas**

Comparando las tablas 4.1 y 4.2 se puede observar que el usuario con Call\_ID 306 fue respectivamente el que tuvo mayor número de llamadas en el mes de enero del 2009.

Con los datos obtenidos podemos verificar que el registro de llamadas de Asterisk sobre las bases de datos del sistema se hacen correctamente, así también, que las tareas programadas que forman parte de la gestión del sistema funcionan de manera correcta ya que se muestra información actual del uso del servicio.

## CAPÍTULO 5

# CONCLUSIONES Y PERSPECTIVAS

### 5.1 Conclusiones.

El objetivo principal de este trabajo de tesis fue cubierto al 100%, es decir, se hizo el análisis, diseño e implementación de un sistema el cual permita almacenar y consultar la información del Servicio VoIP de la Facultad de Computación, además de actualizar esta información cada cierto periodo. También de enviarles a los usuarios un reporte mensual el cual indica el uso de este servicio para cada usuario.

Dado lo anterior se cumplió con las expectativas propuestas inicialmente, además de haber concluido satisfactoriamente el propósito de este sistema, como complemento del servicio de VoIP de la Facultad.

Además como caso personal el aumento de experiencia en la integración de distintos lenguajes para crear un solo sistema.

### 5.2 Limitaciones.

Las consultas que se pueden realizar son muy extensas, por lo que solo se tomaron las que suponen un uso frecuente.

Como cualquier sistema que almacena información, la capacidad de almacenamiento dependerá del hardware del sistema.

### 5.3 Perspectivas.

A futuro se podría integrar los datos de los usuarios con la configuración de usuarios de Asterisk, por medio de una base de datos, con lo que ayudaría a evitar hacer manualmente la inserción de usuarios a la base de datos del sistema.

Modificar el módulo de Asterisk (`cdr_mysql.c`) para que registre mas detalles de la llamada, como es, fecha y hora en que se contestó, fecha y hora en que se colgó.

Mejorar el sistema para el soporte de nuevas extensiones y tipos de marcado que se vayan agregando.

Aumentar los tipos de consultas, como vayan requiriéndose.

Aumentar el soporte de administradores de base de datos, para que no solamente se limite a un solo Manejador de Base de Datos.

Agregar un módulo para que almacene, consulte y modifique los tipos de tarifas basados en los dialplan.

En general, ir mejorando el sistema como vaya requiriéndose, además de ir optimizando la integración de este con el sistema Asterisk.

**BIBLIOGRAFIA**

- [1]. Building Telephony Systems With Asterisk by David Gomillion, Barrie Dempster, Editorial PACKT Publishing, Release date September 2005
- [2]. Jim Van Maggelen, Leif Madsen & Jared Smith, "Asterisk: The Future of Telephony", 2<sup>nd</sup> Edition, Editorial: O'REILLY (Safari),USA, Agosto 2007.
- [3]. Douskalis, B. (2000). Ip Telephony: The Integration Of Robust Voip Services. New Jersey: Prentice Hall Ptr
- [4]. VoIP for DUMMIES, por Timothy Kelly, Editorial Wiley Publishing, Inc, 2005, Indianapolis, Indiana.
- [5]. Manual Imprescindible de PHP5, Luis Miguel Cabezas Granado Prólogo de Zeev Suraski y Andi Gutmans, EDICIONES ANA YA MULTIMEDIA (GRUPO ANAYA, S.A.), 2004 Juan Ignacio Luca de Tena, 15. 28027 Madrid
- [6]. Manuales Plan Avanza, "La factura electrónica", edición electrónica, editado por: ©Red.es, © ASIMELEC, España.
- [7]. XSL-FO por Dave Pawson, Agosto del 2002, editorial O'Reilly.
- [8]. Learning XML, Second Edition, por Erik T. Ray en Septiembre del 2003, Editorial O'Reilly.
- [9]. XSLT, Second Edition, por Doug Tidwell , Junio del 2008, Editorial O'Reilly.
- [10]. Managing & Using MySQL, Second Edition Open Source SQL Databases for Managing Information & Web Sites,By George Reese, Randy Jay Yarger, Tim King, With Hugh E. Williams, Second Edition April 2002
- [11]. Introducción a CSS por Javier Eguíliz Pérez, edición electrónica gratuita desde [www.librosweb.es](http://www.librosweb.es) .
- [12]. Introducción a AJAX por Javier Eguíliz Pérez, edición electrónica gratuita desde [www.librosweb.es](http://www.librosweb.es) .
- [13]. Introducción a Javascript por Javier Eguíliz Pérez, edición electrónica gratuita desde [www.librosweb.es](http://www.librosweb.es) .
- [14]. *The Definitive Guide to Premise-Based PB*, por [www.voip-news.com](http://www.voip-news.com).

## APÉNDICE I

## MIGRACIÓN E INSTALACIÓN

**Instalación.**

Para la instalación del sistema de facturación se necesitan los siguientes requisitos:

**Requisitos Software.**

Tener instalado y configurado un sistema basado en Asterisk.

Tener las siguientes aplicaciones en el sistema Asterisk:

- xsltproc
- mailx
- mysql
- sed
- awk

Descargar los add-ons de Asterisk si es que no están presentes, el cual se puede descargar desde la siguiente página: <http://www.asterisk.org/downloads>.

Descargar la aplicación Fop de Apache (binario), el cual se puede descargar desde la siguiente página: <http://xmlgraphics.apache.org/fop/download.html>

Tener instalado un servidor web con soporte de PHP y MYSQL.

Tener instalado el Administrador de Bases de Datos MYSQL.

Una vez teniendo los requisitos, se prosigue con lo siguiente:

- 1) Descomprimir e instalar los add-ons de Asterisk si estos no existen en el sistema. Para instalar se usa los siguientes comandos :

- I. *./configure*
- II. *make clean*
- III. *make*

IV. *make install*

V. *make samples*

En dado caso que marcara error, se debe configurar los add-ons que se requieran instalar, se usara *make menuselect* y se Deshabilita oh323 y mpg3.

2) Se crea una base de datos llamada *asterisk* con la siguiente tabla:

```
CREATE TABLE `cdr1` (  
  `calldate` datetime NOT NULL default '0000-00-00 00:00:00',  
  `callans` datetime NOT NULL default '0000-00-00 00:00:00',  
  `callend` datetime NOT NULL default '0000-00-00 00:00:00',  
  `clid` varchar(80) NOT NULL default '',  
  `src` varchar(80) NOT NULL default '',  
  `dst` varchar(80) NOT NULL default '',  
  `dcontext` varchar(80) NOT NULL default '',  
  `channel` varchar(80) NOT NULL default '',  
  `dstchannel` varchar(80) NOT NULL default '',  
  `lastapp` varchar(80) NOT NULL default '',  
  `lastdata` varchar(80) NOT NULL default '',  
  `duration` int(11) NOT NULL default '0',  
  `billsec` int(11) NOT NULL default '0',  
  `disposition` varchar(45) NOT NULL default '',  
  `amaflags` int(11) NOT NULL default '0',  
  `accountcode` varchar(20) NOT NULL default '',  
  `userfield` varchar(255) NOT NULL default '',  
  `uniqueid` varchar(32) NOT NULL default '',  
  PRIMARY KEY (`uniqueid`),  
  KEY `calldate` (`calldate`),  
  KEY `dst` (`dst`),  
  KEY `accountcode` (`accountcode`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

3) Se crea un usuario en MYSQL llamado *asterisk* y se le asignan permisos de lectura y escritura a las bases de datos *asterisk* y *adminVoIP*.

4) Accesamos al archivo de configuración  
cd /etc/asterisk  
nano cdr\_mysql.conf lo editamos de la siguiente manera  
[global]

```
hostname=localhost
dbname=asterisk
table=cdr1
password=asterisk (este password puede ser diferente)
user=asterisk
port=3306
sock=/tmp/mysql.sock
;userfield=1
```

- 5) Además configuramos el siguiente archivo.

```
# nano modules.conf
[modules]
autoload=yes
Y AGREGAMOS AL FINAL
load => cdr_addon_mysql.so
```

- 6) Y recargamos los módulos

```
#asterisk -rvvvvdddd
# CLI>reload
```

- 7) Se crea una base de datos llamada *adminVoIP* y se agregan las siguientes tablas:

Para los registros de llamadas:

```
CREATE TABLE IF NOT EXISTS `llamadas` (
  `Fecha` date NOT NULL,
  `hora` time NOT NULL,
  `nombre` varchar(80) NOT NULL,
  `telefono` varchar(80) NOT NULL,
  `id` varchar(80) NOT NULL,
  `destino` varchar(80) NOT NULL,
  `duracion` int(11) NOT NULL,
  `idast` varchar(32) NOT NULL,
  `lastapp` varchar(80) character set utf8 collate utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`idast`),
  KEY `telefono` (`telefono`),
  KEY `Fecha` (`Fecha`),
  KEY `destino` (`destino`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Y la tabla usuarios:

```
CREATE TABLE IF NOT EXISTS `usuarios` (  
  `id` varchar(80) NOT NULL,  
  `cid_number` varchar(11) NOT NULL,  
  `fullname` varchar(80) NOT NULL,  
  `secret` varchar(30) NOT NULL,  
  `email` varchar(80) NOT NULL default 'ninguno',  
  PRIMARY KEY (`id`,`cid_number`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

### 8) Se anexan las funciones usadas en la base de datos adminVoIP:

```
CREATE DEFINER=`asterisk`@`localhost` FUNCTION `ncallym`(idx varchar(10),telefonox varchar(10),anio varchar(4),mes varchar(2)) RETURNS int(11)  
begin  
  declare total int;  
  declare date varchar(10);  
  if NOT (anio = '') THEN  
    IF NOT(mes = '') THEN  
      SELECT CONCAT(anio,'-',mes,'-%') INTO date;  
    ELSE  
      SELECT CONCAT(anio,'%') INTO date;  
    END IF;  
  ELSE  
    SELECT CONCAT('%-',mes,'-%') INTO date;  
  END IF;  
  IF NOT (idx = '') THEN  
    IF NOT (telefonox = '') THEN  
      SELECT COUNT(*) INTO total FROM llamadas WHERE id = idx AND telefono = telefonox AND Fecha LIKE date;  
    ELSE  
      SELECT COUNT(*) INTO total FROM llamadas WHERE id = idx AND Fecha LIKE date;  
    END IF;  
  ELSE  
    IF NOT (telefonox = '') THEN  
      SELECT COUNT(*) INTO total FROM llamadas WHERE telefono = telefonox AND Fecha LIKE date;  
    ELSE  
      SELECT COUNT(*) INTO total FROM llamadas WHERE Fecha LIKE date;  
    END IF;  
end;
```

```

        END IF;

        RETURN total;

    END
-----
CREATE DEFINER=`asterisk`@`localhost` FUNCTION `ncalltym`(idx varchar(10),telefonox varchar(10),tipox int,anio varchar(4),mes varchar(2)) RETURNS int(11)
begin
    declare total int;
    declare date varchar(10);
    if NOT (anio = '') THEN
        IF NOT(mes = '') THEN
            SELECT CONCAT(anio,'-',mes,'-%') INTO date;
        ELSE
            SELECT CONCAT(anio,'%') INTO date;
        END IF;
    ELSE
        SELECT CONCAT('%-',mes,'-%') INTO date;
    END IF;

    IF NOT (idx = '') THEN
        IF NOT (telefonox = '') THEN
            SELECT COUNT(*) INTO total FROM llamadas WHERE id = idx AND telefono = telefonox AND Fecha LIKE date AND LENGTH(destino)=tipox;
        ELSE
            SELECT COUNT(*) INTO total FROM llamadas WHERE id = idx AND Fecha LIKE date AND LENGTH(destino)=tipox;
        END IF;
    ELSE
        IF NOT (telefonox = '') THEN
            SELECT COUNT(*) INTO total FROM llamadas WHERE telefono = telefonox AND Fecha LIKE date AND LENGTH(destino)=tipox;
        ELSE
            SELECT COUNT(*) INTO total FROM llamadas WHERE Fecha LIKE date AND LENGTH(destino)=tipox;
        END IF;
    END IF;

    RETURN total;

END
-----
CREATE DEFINER=`asterisk`@`localhost` FUNCTION `ncalldym`(idx varchar(10),telefonox varchar(10),destx varchar(10),anio varchar(4),mes varchar(2)) RETURNS int(11)

```

```
begin
    declare total int;
    declare date varchar(10);
    declare dlen int;
    SELECT LENGTH(destx) INTO dlen;
    CASE dlen
    WHEN 3 THEN SET dlen=3;
    WHEN 4 THEN SELECT CONCAT('9%',destx) INTO destx; SET dlen=5;
    WHEN 7 THEN SELECT CONCAT('99%',destx,'%') INTO destx; SET dlen=14;
    ELSE RETURN 0;
    END CASE;
    if NOT (anio = '') THEN
        IF NOT(mes = '') THEN
            SELECT CONCAT(anio,'-',mes,'-%') INTO date;
        ELSE
            SELECT CONCAT(anio,'%') INTO date;
        END IF;
    ELSE
        SELECT CONCAT('%-',mes,'-%') INTO date;
    END IF;

    IF NOT (idx = '') THEN
        IF NOT (telefonox = '') THEN
            SELECT COUNT(*) INTO total FROM llamadas WHERE id = idx AND telefono = telefonox AND
            Fecha LIKE date AND LENGTH(destino)=dlen AND destino LIKE destx;
        ELSE
            SELECT COUNT(*) INTO total FROM llamadas WHERE id = idx AND Fecha LIKE date AND
            LENGTH(destino)=dlen AND destino LIKE destx;
        END IF;
    ELSE
        IF NOT (telefonox = '') THEN
            SELECT COUNT(*) INTO total FROM llamadas WHERE telefono = telefonox AND Fecha LIKE
            date AND LENGTH(destino)=dlen AND destino LIKE destx;
        ELSE
            SELECT COUNT(*) INTO total FROM llamadas WHERE Fecha LIKE date AND
            LENGTH(destino)=dlen AND destino LIKE destx;
        END IF;
    END IF;
    RETURN total;
end
```

END

-----  
 CREATE DEFINER=`asterisk`@`localhost` FUNCTION `wcallmasym`(anio varchar(4),mes varchar(2))  
 RETURNS varchar(10)

begin

    DECLARE done INT DEFAULT 0;

    declare v int;

    declare x varchar(10);

    declare y varchar(10);

    declare total int;

    declare max int;

    declare len int;

    declare date varchar(10);

    declare usr varchar(10);

    DECLARE traza CURSOR FOR

    SELECT id,cid\_number FROM usuarios;

    DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 1;

    SET v = 0;

    SET max = 0;

    SET usr = '0';

    SELECT COUNT(\*) INTO len FROM usuarios;

    if NOT (anio = "") THEN

        SELECT CONCAT(anio,'-',mes,'-%') INTO date;

    ELSE

        SELECT CONCAT('%-',mes,'-%') INTO date;

    END IF;

    OPEN traza;

    REPEAT

    FETCH traza INTO x,y;

    loop\_label : loop

        SELECT ncallym(x,y,anio,mes) INTO total;

        SET v = v + 1;

        IF total >= max then

            SET max = total;

        IF max > 0 then

            SET usr = x;

        END IF;

    end IF;

```
IF v >= len then
leave loop_label;
end IF;
end loop;
UNTIL done END REPEAT;
CLOSE traza;
RETURN usr;
end
```

- 9) Se crea la estructura de directorios como se muestra en la figura 4.1
- 10) Se copian los archivos anexados a este documento, en sus respectivos directorios.
- 11) Se descomprime la aplicación *Fop* en la carpeta */data/Fop*
- 12) Para el servidor web, se crea el directorio *php* y se copian los archivos anexados al documento, además de configurar en *php.ini* para que esta carpeta se considere como *include*.
- 13) Se crea un directorio como por ejemplo *html* (véase figura 4.2) para los archivos de la aplicación web.
- 14) Se agregan tareas al *crontab* de los shell script: *tesis* y *cronsemanal*, para el primero ejecutarse el primer día de cada mes, y para el de *cronsemanal*, ejecutarse cada inicio de semana.

### Observaciones:

En la configuración de Asterisk este ya traía por default un servidor web, y se instaló apache como segundo servidor web, el primero administra el GUI de instalación y configuración vía web y el segundo (apache) es el que usamos para nuestra aplicación.

Para que este funcione se necesita filtrar los puertos para cada uno, en el caso de *lighthttpd* que viene por default en Asterisk se le asignó el puerto 443 (https) y para Apache el puerto 80 (http) respectivamente.

### Migración.

Para restaurar los registros respaldados por el sistema en dado caso de un upgrade o migración de hardware, se realiza lo siguiente:

- 1) Se descomprime el último respaldo hecho por el sistema.

- 2) Se ejecuta el siguiente comando: *mysql -u asterisk -ppassword -h localhost -D adminVoIP < respaldo\_db.sql*.
- 3) Con lo anterior se recupera los registros de llamadas, además de la tabla de los usuarios.
- 4) Si se desea restaurar alguna parte de la tabla cdr1 para verificar cualquier llamada, se descomprime el archivo con el pedazo de tabla a restaurar y se ejecuta el mismo comando que en el paso 2, usando una base de datos temporal para este propósito.

También se puede restaurar el Master.csv que por default trae Asterisk, el cual se puede hacer con los siguientes comandos desde la consola de MYSQL:

```
USE base_datos_temp
```

```
LOAD DATA LOCAL INFILE '/var/log/asterisk/cdr-custom/Master.csv'
```

```
INTO TABLE cdr1
```

```
FIELDS TERMINATED BY ',' ENCLOSED BY ''''
```

```
LINES TERMINATED BY '\n'
```

```
(clid, src, dst, dcontext, channel, dstchannel, lastapp, lastdata, calldate,callans,callend, duration, billsec, disposition, amaflags, accountcode, uniqueid, userfield);
```

## APENDICE II

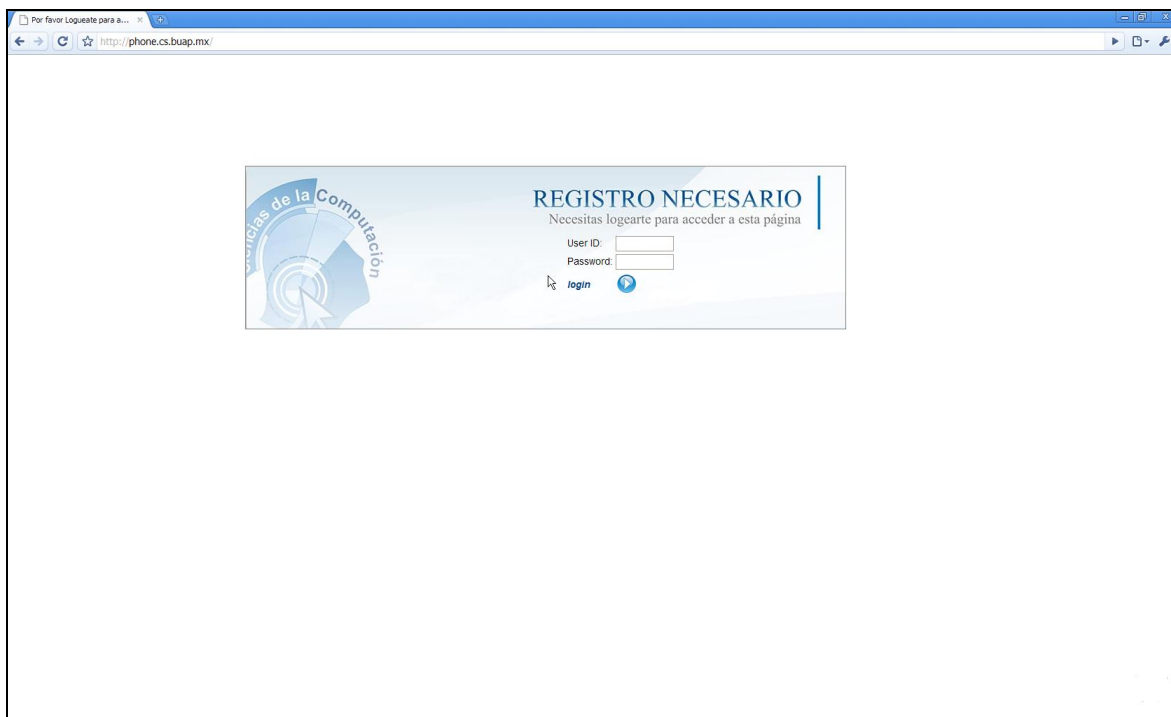
# MANUAL DE USUARIO

### Introducción.

En las siguientes páginas se describirán las ventanas y opciones que se muestran en las interfaces Web de la aplicación. El Usuario debe de conocer de antemano su login y password que se le fue asignado. Este por lo general es el mismo para el uso del servicio VoIP.

### Ingreso al sistema.

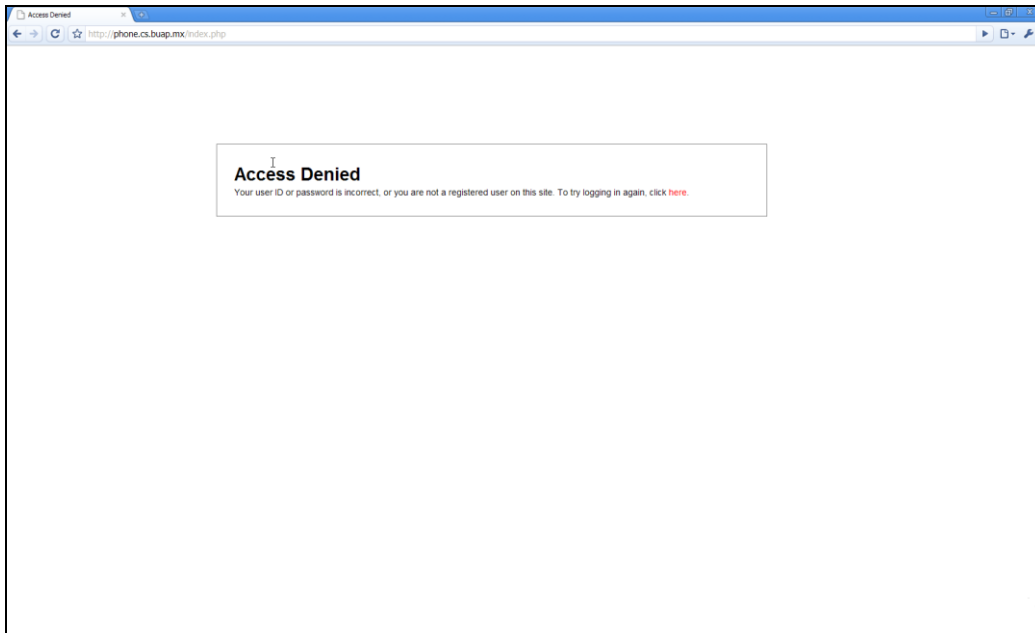
El usuario deberá ingresar la siguiente dirección en su navegador Web:  
<http://phone.cs.buap.mx>, el cual al abrir mostrara la ventana para acceder al sistema (véase Fig. A2.1).



**Figura A2.1** Interfaz para ingresar al Sistema.

A continuación el Usuario proporcionara su login (User ID) y su respectivo password, para después presionar sobre el botón azul para ingresar. En caso que al ingresar los datos

estén incorrectos o no exista dicho usuario, se le mostrará una ventada denegando el acceso y un link para volver a intentar ingresar (véase fig. A2.2).



**Figura A2.2 Acceso denegado al sistema.**

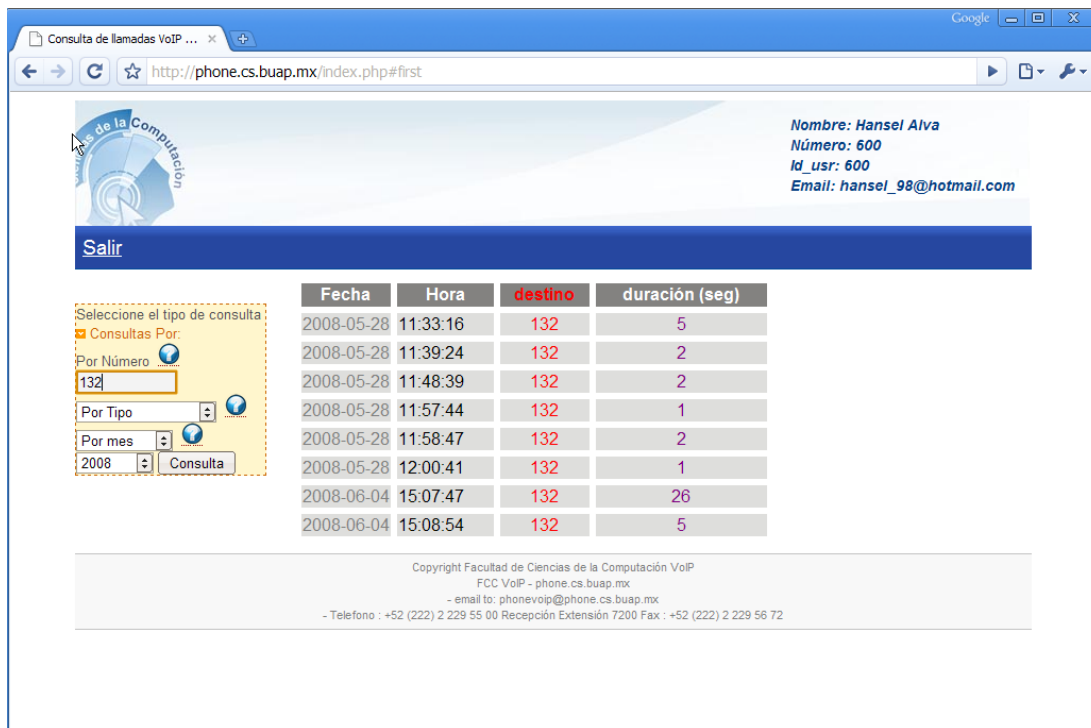
Una vez que se ingreso al sistema, dependiendo el tipo de usuario se mostrara las opciones de consulta. En caso de un usuario normal se mostrarán las siguientes opciones:

a) Consulta Por Llamadas.

Como se muestra en la figura A2.3, la consulta “por Llamadas” muestra varias opciones que a continuación se explican:

- I. Por Número: En este campo de texto el usuario pondrá el número telefónico al cual llamó.
- II. Por Tipo: En este campo de selección el usuario podrá escoger de entre tres tipos de llamada, VoIP, Local BUAP y Teléfono particular, la cual será un filtro al realizar la consulta.
- III. Por mes: En este campo de selección el usuario podrá escoger uno de los 12 meses del año para precisar más la consulta.
- IV. Por Año: Al igual que el campo de Por mes, el usuario podrá seleccionar un año para precisar más la consulta.
- V. Botón Consulta: Este botón es para enviar los datos al sistema.

El resultado al haber hecho una consulta por llamada, es la siguiente:



**Figura A2.3 Consulta por Llamadas**

En la figura A2.3 se ve un ejemplo de consulta por llamada, la cual se introdujo los siguientes datos, el número igual a 132 y el año 2008.

En el caso que el usuario sea el Admin, se mostrarán las siguientes opciones:

- a) Consulta Por Llamadas.
- b) Consulta Quien llamó más.
- c) Consulta por Duración.
- d) Información de Usuarios.

En Consulta por Llamadas como se muestra en la figura A2.3, se dan las siguientes opciones:

- I. Cantidad de Llamadas: Esta opción es de checkbox, lo que hace a la consulta es regresar el número de llamadas, a partir de los datos enviados.
- II. Por Número: Este campo de texto indica que el admin debe ingresar el CALL ID del usuario que quiere consultar.
- III. Por Extensión: Al igual que el campo anterior el admin debe ingresar la extensión que quiere consultar.
- IV. Por Destino: Este campo de texto el admin deberá ingresar el número telefónico al cual el usuario llamo.

- V. Por Tipo: El admin podrá seleccionar entre los tipos de llamada disponibles para realizar la consulta.
- VI. Por mes: El admin tendrá la opción de seleccionar uno de los 12 meses para precisar más la consulta.
- VII. Por Año. Al igual que en el campo anterior, el usuario podrá seleccionar un año de los disponibles para precisar más la consulta.
- VIII. Botón Consulta: Este botón es para enviar los datos al sistema.

El resultado de esta consulta es la siguiente:



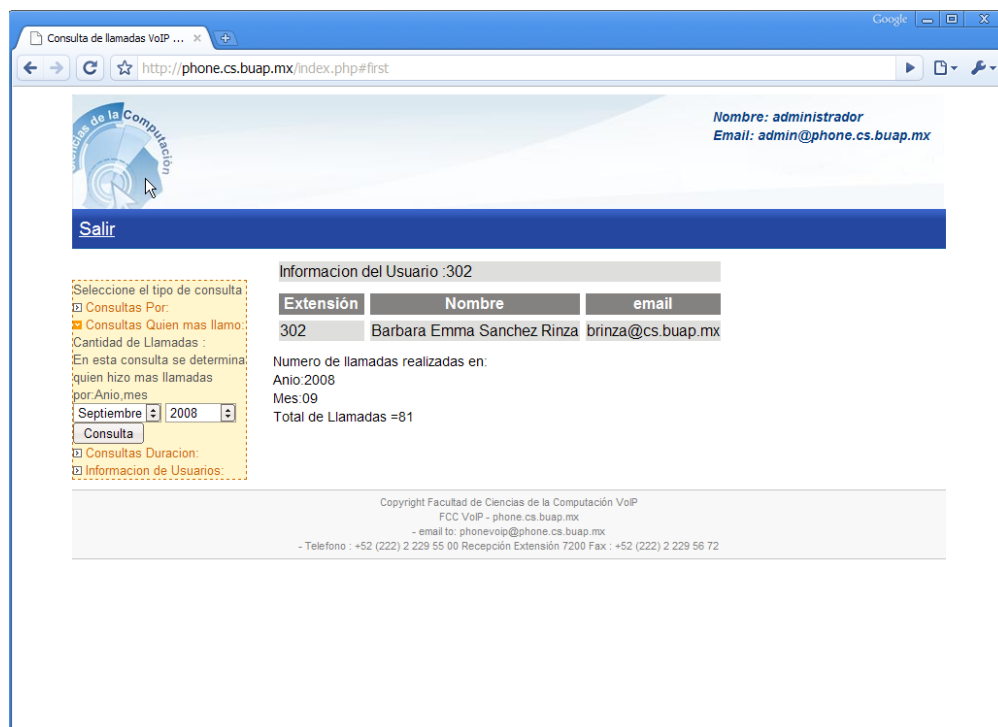
Figura A2.4 Consultas por llamada para el admin.

En el ejemplo anterior se introdujo los siguientes datos Número (Call ID) igual a 132, tipo de llamada VoIP, mes de Agosto y año 2008.

En la Consulta Quien llamó mas, se muestran las siguientes opciones:

- I. Por mes: Se podrá seleccionar uno de los 12 meses para determinar al usuario que llamó más en el mes seleccionado.
- II. Por año. Se podrá seleccionar entre los años disponibles para determinar al usuario que llamó más.
- III. Botón Consulta: Botón para enviar los datos al sistema.

El resultado de esta consulta es la siguiente:



**Figura A2.5 Consulta de quien llamo mas**

En la figura A2.5 se usaron los siguientes valores: mes de Septiembre y año del 2008.

En la Consulta por Duración, se muestran las siguientes opciones:

- I. Por Número: Este campo de texto indica que el admin debe ingresar el CALL ID del usuario que quiere consultar.
- II. Por Extensión: Al igual que el campo anterior el admin debe ingresar la extensión que quiere consultar.
- III. Por Destino: Este campo de texto el admin deberá ingresar el número telefónico al cual el usuario llamó.
- IV. Por Tipo: El admin podrá seleccionar entre los tipos de llamada disponibles para realizar la consulta.
- V. Por mes: El admin tendrá la opción de seleccionar uno de los 12 meses para precisar más la consulta.
- VI. Por día: El admin tendrá la opción de seleccionar uno de los 31 días para precisar la consulta.
- VII. Por año. Al igual que en el campo anterior, el usuario podrá seleccionar un año de los disponibles para precisar mas la consulta.

VIII. Botón Consulta: Este botón es para enviar los datos al sistema.

El resultado de esta consulta es la siguiente:

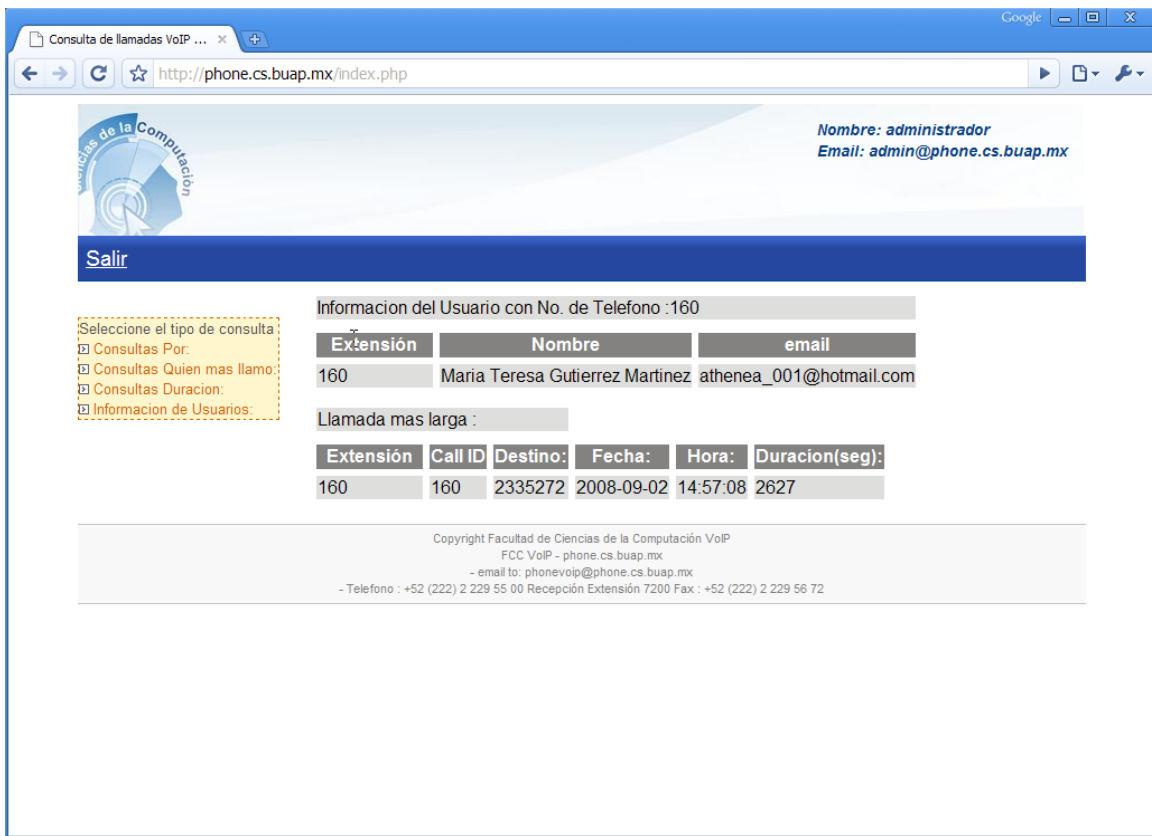


Figura A2.6 Consulta de Llamada por mayor duración

Sobre la consulta A2.6 se usaron los siguientes valores: mes de Septiembre y año del 2008.

En la consulta de Información de Usuarios se dan las siguientes opciones:

- I. Botón de Info de usuario: Manda la petición al sistema para mostrar la información de los usuarios.

El resultado de la consulta se muestra en la figura A2.7.

Si el usuario ha terminado de realizar sus actividades, la interfaz de usuario posee un link para salir del sistema el cual se puede apreciar en la parte izquierda superior, con las palabras “SALIR”, el cual lo retornará a la pantalla para ingresar al sistema.

Consulta de llamadas VoIP ... x +

http://phone.cs.buap.mx/index.php

Nombre: administrador  
Email: admin@phone.cs.buap.mx

Salir

Seleccione el tipo de consulta:  
 Consultas Por:  
 Consultas Quien mas llamo:  
 Consultas Duracion:  
 Informacion de Usuarios:

Extensión	CALL ID	Nombre	e-mail
600	600	Hansel Alva	hansel_98@hotmail.com
601	0003	Guss	ninguno
602	602	Misael Rodriguez	vojamal@gmail.com
603	0004	New User	ninguno
604	0004	Biblio temporal	ninguno
101	101	Ramon Aguirre Vara	ramon@cs.buap.mx
102	102	Almadelia Ambrosio	ambrosio@cs.buap.mx
103	103	Mario Anzures Garcia	anzures@cs.buap.mx
100	100	Etelvina Archundia	etelvina@cs.buap.mx
104	104	Ariza Velazques Eduardo	ninguno
105	105	Apolonio Ata Perez	apolonio@cs.buap.mx
106	106	Cesar Bautista Ramos	bautista@cs.buap.mx
107	107	Pedro Bello	pbello@cs.buap.mx
108	108	Beatriz Beltran Martinez	bbeltran@cs.buap.mx
109	109	Blanca Bermudez Juarez	bbj@cs.buap.mx
110	110	Rocio Boone	rboone@cs.buap.mx
111	111	Maya Carrillo Ruiz	cmaya@cs.buap.mx
112	112	Hilda Castillo Zacatelco	hilda@cs.buap.mx
113	113	Mauricio Castro	mcastro@cs.buap.mx
114	114	Carmen Ceron	mceron@cs.buap.mx

Primero [Pagina 1/5](#) [Siguiente](#) [Ultimo](#)

Copyright Facultad de Ciencias de la Computación VoIP  
 FCC VoIP - phone.cs.buap.mx  
 - email to: phonevoip@phone.cs.buap.mx  
 - Telefono : +52 (222) 2 229 55 00 Recepción Extensión 7200 Fax : +52 (222) 2 229 56 72

Figura A2.7 Consulta de información de Usuarios.