



Universidad Carlos III de Madrid

Escuela Politécnica Superior

Departamento de Informática

Proyecto Final de Carrera de
Ingeniería Técnica en Informática de Gestión

OBELISK

“Portal Web de gestión de usuarios de una centralita de VoIP”

Autor: Ana Isabel Gómez-Pimpoyo Fernández

Tutores: Javier Fernández Muñoz

Alejandro Calderón Mateos

Leganés, Octubre de 2010

A mi familia, a los que están en cuerpo y alma, y a los que sólo están en alma, por hacer posibles mis sueños.

“Por más difícil que se nos presente una situación, nunca dejemos de buscar la salida, ni de luchar hasta el último momento. En momentos de crisis, sólo la imaginación es más importante que el conocimiento” (Albert Einstein)

Agradecimientos

Agradezco a mis tutores toda la atención y paciencia que me han prestado, los conocimientos y los nuevos caminos que han abierto en mi vida gracias a sus propuestas.

A mi familia que ha estado apoyándome en todo momento. En especial a mi padre, que desde donde esté verá otro de sus sueños cumplidos.

A mi chico, por su ayuda, paciencia y por el tiempo que no hemos podido compartir.

También agradezco el apoyo incondicional de mis compañeros que siempre estuvieron ahí para cuando me atascaba darme un empujoncillo.

Por lo nombrado y por muchas cosas más que cada uno sabe, GRACIAS.

Resumen

OBELISK es una interface gráfica, desarrollada como página *Web*, que tiene como función principal simplificar el alta de usuarios en una centralita de voz IP, en este caso en Asterisk. Otras funciones destacables son:

- Poder programar llamadas a una hora determinada a un conjunto de contactos del usuario.
- Realizar llamadas desde la aplicación *Web* sin necesidad de tener un software adicional instalado.
- Permite al usuario gestionar su agenda de contactos.

Para poder ofrecer estas funcionalidades se hace uso de la centralita de Voz IP Asterisk y del programa TTS (*Text To Speech*, texto a voz) Festival entre otros.

Abstract

OBELISK is a graphic interface developed as a Web page whose main function is to simplify users' register in an IP voice central, in this case Asterisk. Other important points are:

- The ability to make schedule calls in a specific moment to contacts' group.
- To make calls from a *Web* application without additional software.
- For users to manage their contact lists.

These functionalities are offered by Asterisk (Voice IP central) and by the Festival Text To Speech program, among others.

Índice general

| | |
|--|----|
| Agradecimientos | 5 |
| Resumen | 7 |
| Abstract | 8 |
| Índice de ilustraciones..... | 13 |
| Índice de tablas..... | 17 |
| 1. Introducción..... | 19 |
| 1.1. Origen y motivación | 19 |
| 1.2. Objetivos..... | 19 |
| 1.3. Estructura del documento..... | 20 |
| 2. Estado de la cuestión | 21 |
| 2.1. Voz sobre IP | 21 |
| 2.1.1. Asterisk..... | 24 |
| 2.1.2. XLite | 30 |
| 2.2. Plataforma | 30 |
| 2.2.1. PHP | 31 |
| 2.2.2. MySQL..... | 32 |
| 2.2.3. Apache..... | 32 |
| 2.2.4. VMWare | 33 |
| 2.2.5. Ubuntu..... | 34 |
| 2.2.6. Festival Speech Synthesis System..... | 35 |
| 3. Análisis | 37 |

| | | |
|--------|---|----|
| 3.1. | Introducción | 37 |
| 3.2. | Especificación de Requisitos del <i>Software</i> (ERS)..... | 39 |
| 3.2.1. | Requisitos de usuario..... | 40 |
| 3.2.2. | Requisitos Software..... | 45 |
| 4. | Diseño | 55 |
| 4.1. | Introducción | 55 |
| 4.2. | Diseño de tres capas | 55 |
| | Capa de presentación..... | 56 |
| | Capa de aplicación (Lógica de negocio/Capa lógica/Capa de acceso a los datos/Capa intermedia)..... | 56 |
| | Capa de datos..... | 56 |
| 4.3. | Diagrama de Flujo de Datos (DFD)..... | 57 |
| 4.4. | Modelo entidad/relación extendido (E/R) | 59 |
| 4.5. | Diagramas de navegación | 61 |
| 4.5.1. | Diagrama de navegación no_usuario..... | 61 |
| 4.5.2. | Diagrama de navegación usuario | 62 |
| 4.5.3. | Diagrama de navegación administrador | 62 |
| 4.6. | Diseño de Base de Datos (modelo relacional) | 62 |
| 5. | Implementación e implantación..... | 65 |
| 5.1. | Descripción de la implementación..... | 65 |
| 5.1.1. | Interfaz con Base de Datos | 65 |
| 5.1.2. | Base de Datos con Asterisk | 65 |
| 5.1.3. | Integración de Asterisk y Festival..... | 66 |
| 5.2. | Implantación..... | 67 |
| 5.2.1. | Ubuntu | 67 |
| 5.2.2. | Asterisk | 68 |
| 5.2.3. | Festival | 68 |
| 5.2.4. | Apache, MySQL y PHP (LAMP)..... | 69 |
| 6. | Presupuesto..... | 71 |

| | | |
|--------|--|----|
| 6.1. | Planificación | 71 |
| 6.2. | Presupuesto | 71 |
| 6.2.1. | Recursos materiales | 72 |
| 6.2.2. | Recursos humanos | 73 |
| 6.2.3. | Gastos indirectos | 74 |
| 6.2.4. | Resumen del presupuesto..... | 75 |
| 7. | Conclusiones y trabajos futuros..... | 77 |
| 7.1. | Conclusiones | 77 |
| 7.1.1. | Sobre el proyecto..... | 77 |
| 7.2. | Trabajos futuros..... | 78 |
| 8. | Apéndices..... | 79 |
| 8.1. | Apéndice A. Bibliografía | 79 |
| 8.2. | Apéndice B. Manual de usuario..... | 80 |
| 8.3. | Apéndice C. Manual del administrador | 93 |

Índice de ilustraciones

| | |
|---|----|
| Ilustración 1: Red de conexión de centralitas a <i>routers</i> que disponen de soporte VoIP..... | 23 |
| Ilustración 2: Esquema de funcionamiento de VoIP con Asterisk..... | 24 |
| Ilustración 3: XLite 3.0..... | 30 |
| Ilustración 4: El modelo de análisis como un puente entre la descripción del sistema y el modelo de diseño (Pressman) | 38 |
| Ilustración 5: Arquitectura de tres capas..... | 57 |
| Ilustración 6: Diagrama de contexto | 58 |
| Ilustración 7: Diagrama de Sistema..... | 59 |
| Ilustración 8: Modelo E/R | 60 |
| Ilustración 9: Diagrama de navegación de no_usuario..... | 61 |
| Ilustración 10: Diagrama de navegación de usuario..... | 62 |
| Ilustración 11: Diagrama de navegación de administrador..... | 62 |
| Ilustración 12: Esquema relacional..... | 63 |
| Ilustración 13: Fichero para la configuración de Festival..... | 68 |
| Ilustración 14: Diagrama Gantt de OBELISK..... | 71 |
| Ilustración 15: Inicio (Manual Usuario) | 80 |
| Ilustración 16: Inicio (Manual Usuario) | 80 |
| Ilustración 17: Inicio (Manual usuario)..... | 81 |
| Ilustración 18: Recordar contraseña (Manual usuario) | 82 |
| Ilustración 19: Pantalla principal (Manual usuario) | 83 |
| Ilustración 20: Principal con agenda (Manual de usuario) | 83 |

| | |
|---|-----|
| Ilustración 21: Navegación usuario (Manual usuario) | 84 |
| Ilustración 22: Navegación usuario I (Manual usuario) | 85 |
| Ilustración 23: VoiceMemo (Manual usuario)..... | 86 |
| Ilustración 24: Crear mensaje (Manual usuario)..... | 86 |
| Ilustración 25: Modificar mensaje (Manual usuario) | 87 |
| Ilustración 26: Navegación usuario II (Manual usuario)..... | 88 |
| Ilustración 27: Modificar perfil (Manual usuario) | 89 |
| Ilustración 28: Navegación usuario III (Manual usuario) | 90 |
| Ilustración 29: Editar agenda (Manual usuario) | 91 |
| Ilustración 30: Modificar contacto (Manual usuario)..... | 92 |
| Ilustración 31: Inicio (Manual administrador) | 93 |
| Ilustración 32: Inicio (Manual administrador) | 93 |
| Ilustración 33: Inicio (Manual administrador) | 94 |
| Ilustración 34: Recordar contraseña (Manual administrador)..... | 95 |
| Ilustración 35: Pantalla principal (Manual administrador)..... | 96 |
| Ilustración 36: Principal con agenda (Manual administrador)..... | 96 |
| Ilustración 37: Navegación administrador (Manual administrador)..... | 97 |
| Ilustración 38: Navegación administrador I (Manual administrador) | 98 |
| Ilustración 39: VoiceMemo (Manual administrador)..... | 99 |
| Ilustración 40: Crear mensaje (Manual administrador)..... | 100 |
| Ilustración 41: Modificar mensaje (Manual administrador) | 100 |
| Ilustración 42: Navegación administrador II (Manual administrador)..... | 101 |
| Ilustración 43: Modificar perfil (Manual administrador) | 102 |
| Ilustración 44: Navegación administrador III (Manual administrador) | 103 |
| Ilustración 45: Editar agenda (Manual administrador) | 104 |
| Ilustración 46: Modificar contacto (Manual administrador)..... | 105 |
| Ilustración 47: Navegación administrador IV (Manual administrador) | 106 |

| | |
|---|-----|
| Ilustración 48: Usuarios (Manual administrador)..... | 107 |
| Ilustración 49: Crear usuario (Manual administrador) | 107 |
| Ilustración 50: Modificar usuario (Manual administrador)..... | 108 |

Índice de tablas

| | |
|--|----|
| Tabla 1: Formato de los requisitos | 40 |
| Tabla 2: RUC 'Acceso al sistema' | 41 |
| Tabla 3: RUC 'Desconexión del sistema' | 41 |
| Tabla 4: RUC 'Modificar contraseña' | 41 |
| Tabla 5: RUC 'Modificar nombre' | 42 |
| Tabla 6: RUC 'Modificar apellido' | 42 |
| Tabla 7: RUC 'Modificar correo electrónico' | 42 |
| Tabla 8: RUC 'Gestionar usuarios' | 43 |
| Tabla 9: RUC 'Gestionar agenda' | 43 |
| Tabla 10: RUC 'Gestionar mensajes programados' | 43 |
| Tabla 11: RUR 'Entorno amigable' | 44 |
| Tabla 12: RUR 'Entorno seguro' | 44 |
| Tabla 13: RUR 'Entorno operativo' | 44 |
| Tabla 14: RUR 'Idioma' | 45 |
| Tabla 15: RUR 'Arquitectura del sistema' | 45 |
| Tabla 16: RUR 'Sistema operativo' | 45 |
| Tabla 17: RSF 'Identificarse' | 46 |
| Tabla 18: RSF 'Cerrar sesión' | 46 |
| Tabla 19: RSF 'Cambiar contraseña' | 47 |
| Tabla 20: RSF 'Consultar usuario' | 47 |
| Tabla 21: RSF 'Borrar usuario' | 47 |

| | |
|---|----|
| Tabla 22: RSF 'Crear mensaje' | 48 |
| Tabla 23: RSF 'Listar mensajes' | 48 |
| Tabla 24: RSF 'Modificar mensaje' | 48 |
| Tabla 25: RSF 'Borrar mensaje' | 49 |
| Tabla 26: RSF 'Crear contacto' | 49 |
| Tabla 27: RSF 'Modificar contacto' | 49 |
| Tabla 28: RSF 'Borrar contacto' | 49 |
| Tabla 29: RSI 'Entorno amigable' | 50 |
| Tabla 30: RSI 'Idioma' | 50 |
| Tabla 31: RSI 'Menú' | 50 |
| Tabla 32: RSR 'Software' | 51 |
| Tabla 33: RSR 'Entorno operativo' | 51 |
| Tabla 34: RUR 'Idioma' | 51 |
| Tabla 35: RSR 'Sistema operativo' | 51 |
| Tabla 36: RSC 'Identificador de usuario' | 52 |
| Tabla 37: RSC 'Repetir contraseña' | 52 |
| Tabla 38: RSC 'Comprobar correo' | 52 |
| Tabla 39: RSC 'Comprobar campos' | 53 |
| Tabla 40: RSD 'Manual de usuario' | 53 |
| Tabla 41: RSD 'Manual de instalación' | 53 |
| Tabla 42: RSS 'Entorno seguro' | 54 |
| Tabla 43: Recursos materiales | 73 |
| Tabla 44: Salarios por categoría | 73 |
| Tabla 45: Gastos de personal imputables al proyecto | 74 |
| Tabla 46: Gastos indirectos | 74 |
| Tabla 47: Resumen del presupuesto | 75 |

1. Introducción

En este capítulo se hace una breve introducción del proyecto indicando el origen y los objetivos, además de mostrar la estructura de este documento.

1.1. Origen y motivación

En un principio este proyecto fue motivado por la curiosidad que despertaba el funcionamiento de las tecnologías VoIP. Después de varios meses de investigación con las herramientas más populares en el mercado, tanto para teléfonos móviles como para teléfonos propiamente IP y centralitas de VoIP, se llegó a la conclusión de que son herramientas muy potentes pero que tienen una configuración bastante tediosa en muchas ocasiones, ya que una operación más o menos sencilla conlleva el tener que modificar gran cantidad de ficheros de configuración.

Se comprobó que había diversos problemas a la hora de trabajar con estos sistemas. El presente proyecto fin de carrera buscará abordar alguno de los principales problemas. Para ello se ha realizado un prototipo que soluciona una parte importante de los problemas que serán comentados más adelante, ofreciendo una interesante solución que se ha denominado OBELISK.

1.2. Objetivos

El objetivo de este proyecto es realizar el análisis, diseño, desarrollo e implantación de una herramienta que permite configurar de forma sencilla una centralita de VoIP, aplicación *Web* para tener toda la versatilidad que se tendría

en el propio puesto de trabajo pero desde *Web* y la realización de llamadas programadas a un conjunto de contactos al mismo tiempo.

Para conseguir estos objetivos se va a utilizar tecnología VoIP, para ello se utilizará Asterisk como centralita de VoIP. Para la realización de llamadas se dispondrá de *softphones* y para la lectura de textos se utilizará 'Festival'.

Todos estos programas estarán integrados en una aplicación *Web* que llamaremos OBELISK.

1.3. Estructura del documento

El documento se divide en los siguientes capítulos:

- **Introducción**, donde se exponen el origen y los objetivos del proyecto, además de la estructura del presente documento.
- **Estado de la cuestión**, donde se expone toda la información relativa a las infraestructuras *hardware* y *software* existentes para la solución que se quiere alcanzar. La información se divide en dos apartados: el apartado relativo a voz IP, y el otro se refiere a la plataforma utilizada para dar la solución al problema planteado.
- **Análisis**, donde se muestra el análisis realizado.
- **Diseño**, donde se muestra el diseño realizado.
- **Implementación e implantación**, donde se muestran los detalles destacables de la implementación y las fases que se han llevado a cabo para la implantación del sistema.
- **Conclusiones y trabajos futuros**, donde se exponen las conclusiones extraídas después del trabajo realizado y en base al funcionamiento del sistema obtenido. Además se incluye un presupuesto, y se explican los futuros trabajos de mejora y ampliación del sistema.
- **Apéndices**, donde se encuentran la bibliografía, el manual de usuario y el manual del administrador.

2. Estado de la cuestión

En este capítulo se presenta una introducción de conceptos para entender el funcionamiento de la telefonía IP¹ y de las herramientas que han sido utilizadas para el desarrollo de la aplicación.

2.1. Voz sobre IP

En sus inicios, el sistema telefónico conmutado público se utilizaba principalmente para el tráfico de voz y algo de tráfico de datos. Pero el tráfico de datos creció bastante, y aproximadamente en 1999, la cantidad de bits de datos movidos igualó a la de bits de voz. En el 2002, el volumen del tráfico de datos era mayor que el volumen del tráfico de voz y continúa creciendo de manera exponencial. Como consecuencia de estas cifras, muchos operadores de redes de conmutación de paquetes de repente se interesaron en transportar voz a través de sus redes de datos. La cantidad de ancho de banda adicional requerida para voz es minúscula debido a que las redes de paquetes están dimensionadas para el tráfico de datos. Sin embargo, probablemente el recibo telefónico de la persona promedio sea más grande que su cuenta de Internet, por lo que los operadores de redes de datos vieron la telefonía de Internet como una forma de ganar una gran cantidad de dinero adicional sin tener que colocar una nueva fibra. De esta forma nació la telefonía de Internet (también conocida como voz sobre IP) (Tanenbaum, 2003).

VoIP es un grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo². Esto significa que se envía la señal

¹ IP: *Internet Protocol*

² Protocolo: acuerdo entre las partes en comunicación sobre cómo se debe llevar a cabo la comunicación (Tanenbaum, 2003)

de voz en forma digital, en paquetes, en lugar de enviarla en forma digital o analógica, a través de circuitos utilizables sólo para telefonía como una compañía telefónica convencional o PSTN (Red Telefónica Pública Conmutada).

Los protocolos que se usan para enviar las señales de voz sobre la red IP se conocen como protocolos de Voz sobre IP o protocolos IP.

El tráfico de Voz sobre IP puede circular por cualquier red IP, incluyendo aquellas conectadas a Internet.

Es muy importante diferenciar entre Voz sobre IP (VoIP) y telefonía sobre IP:

- VoIP es el conjunto de normas, dispositivos, protocolos, en definitiva la tecnología que permite comunicar sobre el protocolo IP.
- Telefonía sobre IP es el servicio telefónico disponible al público realizado con tecnología de VoIP.

La ventaja de este tipo de servicios frente a los que se conocen tradicionalmente es el bajo coste, sobre todo a larga distancia. El ahorro es debido a la utilización de una misma red para llevar voz y datos, especialmente cuando los usuarios tienen sin utilizar toda la capacidad de la red, la parte de red sin utilizar se puede utilizar para usar VoIP sin un costo adicional.

El desarrollo de *codecs*³ para VoIP ha permitido que la voz se codifique en paquetes de datos cada vez más pequeños, lo que deriva en que las comunicaciones de voz sobre IP requieran anchos de banda igualmente pequeños. A ello se suma el avance de las conexiones ADSL en el mercado residencial, por lo que la utilización de VoIP se está haciendo muy popular para llamadas internacionales.

Una de las funcionalidades más destacables de la VoIP es la independencia que permite, ya que en cualquier lugar que se tenga conexión a Internet y un teléfono IP, uno puede configurar su teléfono y recibir llamadas sin depender de que el número de teléfono varíe como es el caso actualmente.

Por todo lo comentado anteriormente, tanto la extensión de Internet en el mundo como el gran ahorro económico, hoy en día es un tema muy interesante y estratégico para las empresas.

³ Codecs: es una caja negra física o lógica, que en la entrada recibe una señal de audio y a la salida devuelve una cadena de bits, o viceversa (discretización o cuantificación de la señal). Es la forma de digitalizar la voz humana para ser enviada por las redes de datos. Algunos ejemplos: G.711, G729A, GSM, iLBC, Speex, G.723.

Al aprovechar el ancho de banda desperdiciado puede perderse calidad en la comunicación por lo que se reserva un ancho de banda dentro de la red para garantizar dicha calidad.

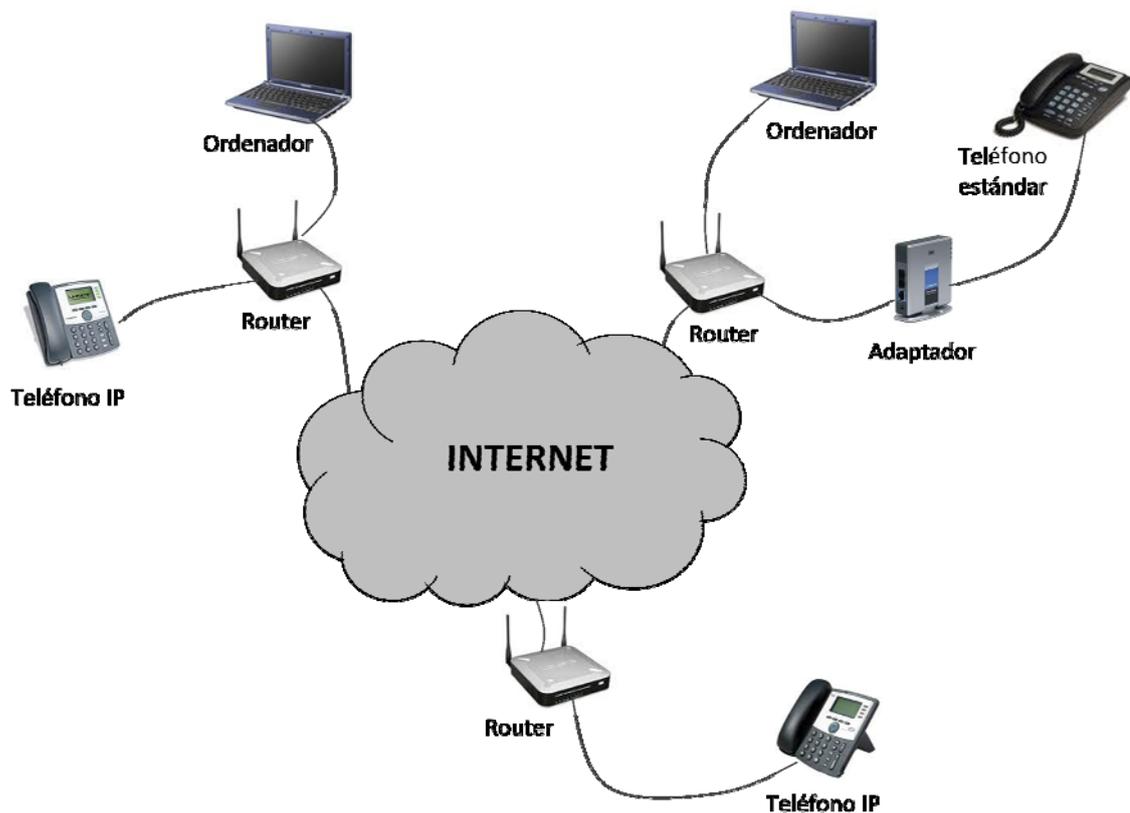


Ilustración 1: Red de conexión de centralitas a routers que disponen de soporte VoIP

El funcionamiento general simplificado de VoIP es el siguiente:

- Paso 1: Los dos comunicantes se registran en el servidor VoIP con sus teléfonos (*Hardphones, Softphones*).
- Paso 2: El equipo del emisor pregunta por el equipo del receptor.
- Paso 3: El servidor VoIP devuelve datos de contacto al emisor (puertos, direcciones IP).
- Paso 4: Se establecen comunicación entre los interlocutores.
- Todo ello utilizando un protocolo determinado: SIP, IAX2, H323 ...

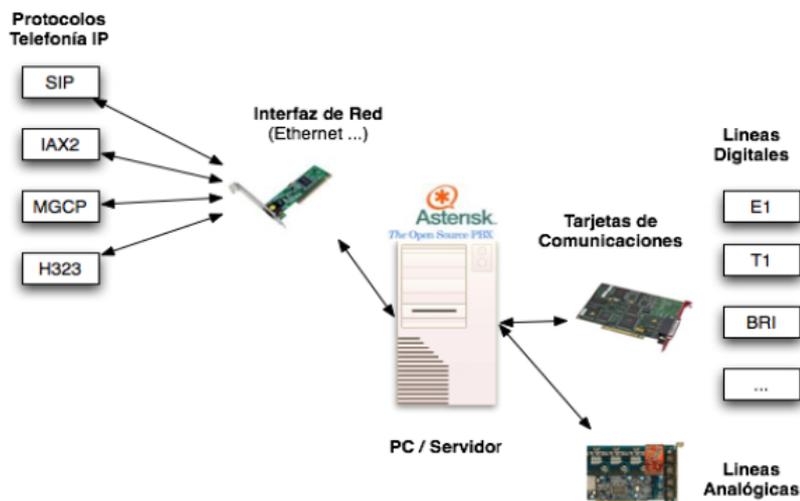


Ilustración 2: Esquema de funcionamiento de VoIP con Asterisk

2.1.1. Asterisk

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

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

Proporciona funcionalidades de una central telefónica (PBX⁴). Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una RDSI tanto básicos como primarios.

Mark Spencer, de Digium, inicialmente creó Asterisk y actualmente es su principal desarrollador, junto con otros programadores que han contribuido a corregir errores y añadir novedades y funcionalidades. Originalmente desarrollado para el sistema operativo GNU/Linux, Asterisk actualmente también se distribuye en versiones para los sistemas operativos BSD, MacOSX, Solaris y Microsoft Windows, aunque la plataforma nativa (GNU/Linux) es la mejor soportada de todas, por eso en este proyecto se decide utilizar Ubuntu como sistema operativo.

⁴ PBX (*Private Branch Exchange*): es cualquier centralita telefónica directamente conectada a la red pública de teléfono por medio de líneas troncales para gestionar tanto las llamadas internas como las entrantes y salientes con autonomía sobre cualquier otra central telefónica.

Asterisk incluye muchas características anteriormente sólo disponibles en costosos sistemas propietarios PBX como buzón de voz, conferencias, IVR, distribución automática de llamadas, y otras muchas más. Los usuarios pueden crear nuevas funcionalidades escribiendo un *dialplan* en el lenguaje de *script* de Asterisk o añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación soportado por Linux.

Para conectar teléfonos estándar analógicos son necesarias tarjetas electrónicas telefónicas FXS o FXO fabricadas por Digium u otros proveedores, ya que para conectar el servidor a una línea externa no basta con un simple módem.

Quizá lo más interesante de Asterisk es que soporta muchos protocolos VoIP como pueden ser SIP, H.323, IAX y MGCP. Asterisk puede interoperar con terminales IP actuando como un registrador y como *gateway* entre ambos.

Asterisk se empieza a adoptar en algunos entornos corporativos como una gran solución de bajo coste junto con SER (*Sip Express Router*).

Poco a poco, esta aplicación se ha convertido en la evolución de las tradicionales centralitas analógicas y digitales permitiendo también integración con la tecnología más actual: VoIP. Asterisk se convierte así en el mejor, más completo, avanzado y económico sistema de comunicaciones existente en la actualidad.

Otro aliciente es su capacidad de ser programada, permitiendo realizar labores que hasta el día de hoy lo llevaban realizando sistemas extremadamente costosos y complicados y, gracias a Asterisk, esta misma labor se realiza de una forma más económica lo que fomenta el uso de sistemas libres como Linux y estándares abiertos como SIP, H323 o IAX.

Una de las ventajas más interesantes es su posibilidad como sistema híbrido, ya que permite gestionar comunicaciones telefónicas tradicionales (analógicas, digitales, móviles,...) como comunicaciones IP mediante el uso de los protocolos estándar de VoIP.

Asterisk es un software completo en PBX, actúa en Linux y provee todas las configuraciones que se esperan de un PBX. Asterisk hace VoIP en los tres protocolos mencionados anteriormente (SIP, H323 e IAX) y puede interoperar con equipos de telefonía estándar básicas usando un hardware relativamente sin costo.

Asterisk provee servicios de *voicemail* con directorios, conferencias, respuesta de voz interactivo IVR, llamadas en espera. Tiene el soporte de tres tipos de formas de llamadas: servicios de llamada con identificación, ADSI, SIP y H323.

Asterisk no necesita ningún *hardware* adicional para el VoIP. Para interconectarse con algún tipo de telefonía digital o analógica.

Asterisk apoya una amplia gama de protocolos para el manejo y transmisión de interfaces de telefonía tradicional. Asterisk apoya al tipo de señalización estándar americano y europeo en asuntos de sistemas de telefonía, permitiendo ser un nexo entre las redes integradas de datos de voz de siguiente generación y la infraestructura existente.

Asterisk provee una base central de conmutación, con 4 API⁵ para la carga modular de los usos de telefonía, interfaces del hardware, dirección del formato del archivo y *Codecs*, permite la conmutación transparente de todas las interfaces soportadas, permitiendo que enlacen una diversidad de mixturas de sistemas de telefonía en una sola red.

Asterisk esta cuidadosamente desarrollado para máxima flexibilidad. Los API específicos son definidos en un sistema central PBX. Este centro avanzado maneja interconexión interna del PBX, abstraídos limpiamente por protocolos específicos, *Codecs*, e interfaces de *hardware* de aplicaciones de telefonía. 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.

- *PBX switching*: La esencia del Asterisk 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 llegando a varios software y hardware de 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 *codecs* para codificar y decodificar varios formatos de comprensión de audio usados en la industria de la telefonía. Un gran número de *codecs* están disponibles para satisfacer necesidades y obtener una mejor calidad del audio.
- *Organizador y manejador*: encargado de las tareas de bajo nivel y sistemas de manejo para un óptimo rendimiento bajo cualquier condición de carga.

⁵ API (*Application programming interface*): conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

La aplicación API provee el flexible uso de aplicaciones modulares para realizar cualquier acción flexible en demanda, también permite un desarrollo abierto de nuevas aplicaciones para satisfacer necesidades o situaciones únicas.

En conclusión, cargar todo el uso como módulos permite un sistema flexible, permitiendo al administrador diseñar la mejor y más satisfactoria trayectoria para los usuarios en el sistema PBX y también para modificar la trayectoria de llamadas para satisfacer las cambiantes necesidades de la comunicación que nos concierne.

Asterisk ofrece funcionalidades básicas PBX y características más avanzadas así como también interopera con sistemas básicos de telefonía estándar y sistemas de VoIP. Ofrece también características como: *voicemail*, conferencias, llamadas en espera y grabado de llamadas.

Asterisk trabaja en una plataforma Linux con o sin hardware que conecta su servidor a la red de telefonía global tradicional PSTN.

Con Asterisk la plataforma de telefonía, PBX, no solo tendrá un sistema PBX de alta calidad, sino que ofrece la posibilidad de telefonar de muchas maneras diferentes:

- Conectar empleados desde casa al PBX Office a través de una conexión de banda ancha.
- Dar a todos los empleados casillas de voz integradas con Internet o red IP privada.
- Construir aplicaciones interactivas de voz, que conecten un sistema ordinario.
- Música para clientes en espera.
- Llamadas en espera, donde se podrán monitorizar las llamadas entrantes y las que hay en espera. Una de las mejores aplicaciones *Open Source* para controlar y visualizar en tiempo real el estado de la centralita es *Flash Operator Panel* (FOP)
- Grabación de los datos de la llamada.
- Sistema de integración de reconocimiento de voz (como Sphinx)
- La capacidad de interfaz con líneas de teléfono normales.

Asterisk es una plataforma completa de telefonía para empresas. Los servidores de Asterisk son capaces de conectar varias oficinas utilizando un protocolo de intercambio de Asterisk (IAX), este protocolo apoya muchas conexiones simultáneas y funciona fácilmente sobre conexiones NAT, de esta manera puede

construir *routers* menos costosos para llamadas y tener un balance de carga ente el servidor de Asterisk y la red.

La configuración de Asterisk está completamente realizada en archivos de texto lo que hace un poco complicada la configuración, ya que para realizar labores sencillas precisa el manipular varios archivos de texto diferentes.

Asterisk se configura desde múltiples ficheros de configuración, cada uno para un determinado área, los más importantes son:

- Fichero de configuración maestro: asterisk.conf.
- Fichero de configuración de módulos: modules.conf.
- Canales:
 - iax.conf: canales *Inter Asterisk eXchange*
 - sip.conf: canales SIP
 - zapata.conf: telefonía analógica y digital.
 - h323.conf: canales H323.
 - mgcp.conf: canales MGCP
- Dialplan⁶:
 - extensions.conf: el propio Dialplan.
 - features.conf: Dialplan para métodos complementarios (transferencias, *call parking*, grabación de llamadas bajo demanda, ...)

La configuración de aplicaciones del Dialplan:

- meetme.conf: para salas de conferencias.
- musiconhold.conf: configuración de la música de espera.
- queues.conf: configuración de colas de llamadas.
- voicemail.conf: configuración de los buzones de voz.
- ...

La gestión de una llamada entrante por un canal puede realizarse utilizando diversas aplicaciones para áreas distintas:

- Comandos generales.
- Gestión de facturación.
- Gestión de llamadas.

⁶ Dialplan: Donde se configura toda la lógica de Asterisk. Es el encargado de enrutar las llamadas utilizando los proveedores configurados. También decide el tiempo de espera, mensajes de alerta, actuación ante ellos.

- Manipulación de variables.
- Buzón de voz.
- Presentación de llamada.
- Integración con aplicación.
- Control de flujo.
- Reproducción/Grabación.
- Colas y conferencias.

Toda la secuencia y programación del Dialplan es el verdadero núcleo del sistema de centralita de Asterisk, si bien, las siguientes funcionalidades se configuran en `features.conf`:

- Transferencias de llamadas: transferencia de llamadas entre diversos usuarios, independientemente de la tecnología que usen.
- *Call Parking*: Parking de llamadas.
- *Call Pickup*: Auto-transferencia de un teléfono que esté sonando.

La configuración de los canales de voz IP se realiza con la manipulación de `sip.conf` e `iax.conf`. La instalación crea ficheros de ejemplo con la sintaxis bastante comentada a modo de guía.

- `sip.conf`

En este fichero se definen:

- Variables generales de SIP.
- Clientes SIP.
- Servidores SIP.

- `iax.conf`

En este fichero se definen todos los enlaces que se realizan usando el protocolo IAX, al igual que en SIP, se pueden definir:

- Variables generales de IAX.
- Clientes IAX.
- Servidores IAX.

2.1.2. XLite



Ilustración 3: XLite 3.0

XLite es un *softphone*, se conoce como *softphone* al *software* que simula tanto en aspecto (ver Ilustración 3) como en funcionamiento, se trata de *software* que se ejecuta en estaciones o servidores de trabajo. Permiten establecer llamadas de Voz sobre IP. El audio es capturado desde:

- Micrófono Incorporado.
- Entrada de línea (Micrófono Externo).
- Dispositivos de entrada de audio USB.
- Dispositivos *Bluetooth*.

Para poder utilizarlo es necesario tenerlo instalado en el ordenador.

2.2. Plataforma

La solución que se expone, OBELISK, utiliza PHP, MySQL, Apache, Festival, VMWare y Ubuntu. La elección de éstas herramienta se detalla a continuación.

2.2.1. PHP

PHP es el lenguaje de lado servidor más extendido en la *Web*. Es un lenguaje que ha tenido una gran aceptación en la comunidad de desarrolladores, debido a la potencia y simplicidad que lo caracterizan, así como al soporte generalizado en la mayoría de los servidores de *hosting*.

PHP permite embeber sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz, es decir, se pueden escribir *scripts* dentro del código HTML⁷. Su mayor interés reside en la gran cantidad de funciones que tiene para la explotación de bases de datos de una manera sencilla.

PHP, aunque es multiplataforma, ha sido concebido inicialmente para entornos UNIX y las tareas fundamentales están definidas como funciones.

La filosofía de PHP es ser progresivamente construido por colaboradores desinteresados que implementan nuevas funciones en nuevas versiones del lenguaje.

PHP es utilizado en este proyecto por todos los motivos mencionados anteriormente y por los que se nombran a continuación:

- Lenguaje orientado al desarrollo de aplicaciones *Web* dinámicas con acceso a información almacenada en Bases de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos (en este caso MySQL).
- Amplia documentación en su página oficial, entre lo que destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que es de fácil acceso.
- El programador puede aplicar cualquier técnica de programación que le permita escribir código ordenado, estructurado y manejable.

⁷ HTML (*Hyper Text Markup Language*): es el lenguaje de marcado (etiquetas) predominante para la elaboración de páginas *Web*. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos.

2.2.2. *MySQL*

MySQL es un sistema de gestión de bases de datos relacionales, con licencia bajo la GPL⁸ de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Consta de dos versiones, una comercial y otra libre (en este proyecto se utiliza la libre)

La versión libre es muy utilizada debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de unas API en gran cantidad de lenguajes (entre ellos PHP).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y claves, manteniendo un muy buen nivel de seguridad en los datos.

Es elegida para este proyecto por todas estas características, porque es bastante ligera y porque la funcionalidad que oferta cumple de manera satisfactoria las necesidades de OBELISK.

2.2.3. *Apache*

Es un servidor *Web* HTTP⁹ de código abierto para plataformas Unix entre otras, que implementa el protocolo HTTP y la noción de sitio virtual. Tiene una amplia aceptación en la red.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido.

⁸ GPL (*General Public License*) de GNU: es una licencia creada por la *Free Software Foundation* en 1989, y está orientada principalmente a proteger la libre distribución, modificación y uso de *software*. Su propósito es declarar que el *software* cubierto por esta licencia es *software* libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

⁹ HTTP (*Hypertext Transfer Protocol*): es el protocolo usado en cada transacción de la *World Wide Web*.

Características:

- Es multiplataforma, aunque idealmente está preparado para funcionar bajo Linux.
- Muy sencillo de configurar.
- Es *Open Source*.
- Amplias librerías de PHP a disposición de los desarrolladores.
- Posee diversos módulos que permiten incorporarle nuevas funcionalidades.
- Al ser muy popular es fácil conseguir soporte.

2.2.4. VMWare

Es un sistema de virtualización por software. Un sistema virtual por *software* es un programa que simula un sistema físico (un ordenador, un *hardware*) con unas características de *hardware* determinadas. Cuando se ejecuta el programa (simulador), proporciona un ambiente de ejecución similar a todos los efectos a un ordenador físico (excepto en el puro acceso físico al *hardware* simulado), con CPU, BIOS, tarjeta gráfica, memoria RAM, tarjeta de red, sistema de sonido, conexión USB, disco duro, etc.

Un virtualizador por *software* permite ejecutar (simular) varios ordenadores (sistemas operativos) dentro de un mismo *hardware* de manera simultánea, permitiendo así el mayor aprovechamiento de recursos. No obstante, y al ser una capa intermedia entre el sistema físico y el sistema operativo que funciona en el hardware emulado, la velocidad de ejecución de este último es menor, pero en la mayoría de los casos es suficiente para usarse en entornos de producción.

VMWare virtualiza una plataforma x86¹⁰ de forma que la mayor parte de las instrucciones en VMWare se ejecutan directamente sobre el *hardware* físico.

VMWare proporciona productos gratuitos que permiten tanto crear máquinas virtuales como hacerlas correr.

Se elige la utilización de máquina virtual para montar OBELISK para poder probar todo lo posible con todo lo que va surgiendo a lo largo del proyecto y

¹⁰ X86: es la denominación genérica dada a ciertos microprocesadores de la familia Intel, sus compatibilidades y la arquitectura básica a la que estos procesadores pertenecen por la terminación de sus nombres numéricos.

realizando copias de la máquina virtual cada vez que se realiza un cambio. Esto lo que simula es la creación de imágenes (*Ghost*) del sistema.

Resulta mucho más cómodo a la hora de trabajar que las imágenes por la comodidad que supone copiar un archivo en lugar de tener que crear una imagen. Estas copias son utilizadas en el momento en el que existe alguna inconsistencia en el sistema causada por algún cambio realizado. Es más sencillo volver a lanzar la copia de seguridad realizada que instalar una imagen.

Se elige VMWare porque ante la oferta existente de este tipo de herramientas, esta ofrece varios productos gratuitos y por haber trabajado con anterioridad con ella.

2.2.5. *Ubuntu*

Ubuntu es una distribución GNU/Linux¹¹ que ofrece un sistema operativo predominantemente enfocado a ordenadores de escritorio aunque también proporciona soporte para servidores.

Basada en Debian GNU/Linux, Ubuntu concentra su objetivo en la facilidad de uso, la libertad de uso, los lanzamientos regulares (cada 6 meses) y la facilidad en la instalación.

El proyecto Ubuntu está totalmente basado en los principios del desarrollo de software de código abierto.

Sin embargo, esto significa algo más que estar disponible a coste cero. La filosofía del *software* libre establece que la gente pueda usar el *software* de todas las formas que considere "socialmente útiles": el código está disponible para poder descargarlo, modificarlo y usarlo del modo que más le convenga. Por tanto, además del hecho de que el *software* libre normalmente puede conseguirse sin coste alguno, esta libertad también tiene ventajas técnicas: al desarrollar programas puede utilizarse el trabajo de los demás y construir a

¹¹ GNU/Linux: es un sistema operativo GNU que usa como núcleo el *kernel* Linux.

El proyecto GNU nació en 1984 de la iniciativa de Richard Stallman con el objetivo de desarrollar un sistema operativo basado en UNIX, pero libre: el sistema GNU. El proyecto GNU está fuertemente unido a la filosofía del Software Libre, que es eje central en los proyectos que como Ubuntu derivan de él.

El *kernel* Linux es el corazón del sistema operativo Ubuntu. Es esencialmente un clónico del núcleo de UNIX creado de manera independiente.

partir de éste. Con el *software* no libre, esto no es posible, y para desarrollar un programa, se tiene que empezar desde cero. Por esta razón, el desarrollo de *software* libre es rápido y eficiente.

La filosofía de Ubuntu se basa en los siguientes principios:

- Ubuntu es gratuito, y no hay un coste adicional para la «edición profesional».
- Para hacer que Ubuntu pueda ser usado por el mayor número de personas posible, Ubuntu emplea las mejores herramientas de traducción y accesibilidad que la comunidad del *Software* Libre es capaz de ofrecer.
- Ubuntu publica de manera regular y predecible, una nueva versión cada seis meses. Puede usar la versión estable o probar y ayudar a mejorar la versión en desarrollo.
- Ubuntu está totalmente comprometido con los principios de desarrollo del *software* de código abierto.

En este proyecto se toma la decisión de utilizar Ubuntu por lo explicado anteriormente y además porque Asterisk nos exige que el sistema operativo sobre el que funciona correctamente sea Linux.

2.2.6. Festival Speech Synthesis System

Es una aplicación orientada a usuarios con disminuciones visuales. Consiste en una herramienta que reconoce caracteres y los pasa a sonidos, estos sistemas son conocidos como TTS¹². Existe para varios lenguajes (ya que la pronunciación es diferente según el lenguaje).

Fue desarrollada en un principio por el Centro de Investigación de Tecnologías del Lenguaje de la Universidad de Edinburgo.

Incluye documentación para desarrollar sistemas de síntesis de voz con varios API, siendo un entorno ideal para el desarrollo de síntesis de voz.

En este proyecto se utiliza porque es *software* libre, funciona sobre Linux, cumple las expectativas necesarias, no es pesado y contiene la conversión de texto a voz en castellano.

¹² TTS: Viene de 'Text To Speech' cuya traducción textual sería texto a voz.

3. Análisis

“La parte más difícil de construir un sistema de software es decidir qué construir. Ninguna parte del trabajo estropea tanto el sistema resultante si se hace mal. Ninguna parte es más difícil de rectificar después” (Fred Brooks)

3.1. Introducción

El modelo de análisis, que en realidad es una serie de modelos, es la primera representación técnica de un sistema.

El análisis de requisitos genera la especificación de características operacionales de *software*; indica la interfaz del *software* con otros elementos del sistema, y establece las restricciones que debe tener el *software*. El análisis de requisitos permite que el analista se extienda sobre requerimientos básicos establecidos durante tareas anteriores a la ingeniería de requisitos y construya modelos que representen escenarios del usuario, actividades funcionales, clases de problemas y sus relaciones, el comportamiento de las clases y el sistema y, a medida que se transforma, el flujo de datos.

El análisis de requisitos le proporciona al diseñador de *software* una representación de información, función y comportamiento que puede trasladar a diseños arquitectónicos, de interfaz y en el nivel de componentes. Por último, el modelado de análisis y la especificación de requisitos ofrecen al desarrollador y al cliente los medios para evaluar la calidad una vez construido el *software*.

(Pressman, 2006)

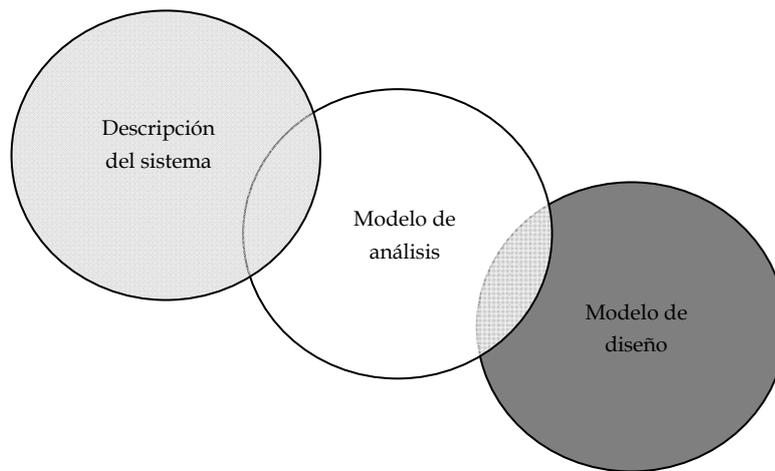


Ilustración 4: El modelo de análisis como un puente entre la descripción del sistema y el modelo de diseño (Pressman, 2006)

Se quiere desarrollar un sistema que a través de una página *Web* ofrezca toda la funcionalidad de un teléfono IP ofreciendo, además, el poder poseer una agenda de teléfono, editar el perfil de usuario y permita enviar mensajes de voz programados (VoiceMemo). La funcionalidad mencionada será para cualquier usuario.

A parte se cuenta con un administrador que además de tener toda la funcionalidad que tiene un usuario va a poder contar con la parte de administración de usuarios.

El funcionamiento del sistema será el siguiente:

El administrador dará de alta a los usuarios introduciendo los datos necesarios para ello (nombre, apellido, correo electrónico,...). Posteriormente se manda el usuario y la contraseña al correo electrónico con el cual se ha dado de alta el usuario.

Cuando una persona abre la página de acceso a la aplicación se encuentra con que tiene que introducir un usuario y una contraseña. Una vez hecho esto, el sistema comprueba si el usuario pertenece a la aplicación y en caso de ser así, la siguiente comprobación que hace es si se trata de un 'usuario' (usuario sin privilegios) o de un 'administrador' (usuario con privilegios de administración), según lo que se resuelva en dicha comprobación el sistema cargará la interfaz correspondiente.

Una vez registrado, el **usuario** puede realizar las siguientes operaciones:

- Editar su perfil: El usuario puede modificar el nombre, los apellidos y la contraseña.
- Realizar llamadas: El usuario puede realizar llamadas desde la *Web* a números dados de alta en el sistema.
- Programar llamadas: El usuario puede programar una llamada, para ello necesitará elegir los destinatarios (que deben estar en su agenda), escribir el mensaje que quieran comunicar y elegir la fecha (incluyendo la hora) a la que se desea enviar el mensaje. Los usuarios podrán modificar y borrar los mensajes que no hayan sido enviados.
- Editar y consultar su agenda: El usuario puede crear, borrar y modificar datos en su agenda.
- El usuario no podrá darse de baja por sí mismo. Las altas, las bajas y las modificaciones de los usuarios (diferentes de las que puede hacerse uno mismo en su perfil) serán controladas por el administrador.

Se registra el **administrador**, el sistema lo lleva directamente a la pantalla perteneciente al entorno del administrador. Las funciones ofrecidas en este caso son las mismas que en el caso de usuario más la de administración de usuario que se describe a continuación:

- Usuarios: donde el administrador podrá hacer las gestiones de administración de usuarios (altas, bajas y modificaciones)

3.2. Especificación de Requisitos del *Software* (ERS)

La ERS se puede definir como la documentación de los requisitos esenciales (funciones, rendimiento, diseño, restricciones y atributos) del *software* y de sus interfaces externos (IEEE, 1990)

La ERS pretende facilitar su utilización por el equipo del proyecto para el trabajo técnico y para la interacción con el usuario, ya que es un elemento clave dentro de la documentación necesaria para el desarrollo de *software*.

En la ERS se incluye información coherente con las necesidades reales del usuario que se desean satisfacer. Dicha información se comunica de forma eficaz de tal manera que se pueda comprender perfectamente.

Por ello se hace la siguiente especificación de requisitos utilizando el siguiente tipo de plantilla (ver Tabla 1).

| IDENTIFICADOR | | | |
|---------------|--|-------------|--|
| Prioridad | | Fuente | |
| Necesidad | | Estabilidad | |
| Descripción | | | |

Tabla 1: Formato de los requisitos

A continuación se detallan los campos que los componen:

- **Identificador:** código único que identifica el requisito.
- **Prioridad:** determina la prioridad del requisito respecto a otros. Su valor puede ser: alta, media o baja.
- **Fuente:** determina el origen del requisito, que en la mayoría de los casos será el cliente.
- **Necesidad:** determina la importancia que tiene dentro del sistema el cumplimiento del requisito. Su valor puede ser: esencial, opcional o deseable.
- **Estabilidad:** define si el requisito es prescindible o si es posible que en un futuro pueda ser modificado o incluso eliminado de la aplicación. Su valor puede ser: alta, media o baja.
- **Descripción:** nombre y breve descripción textual del requisito.

3.2.1. Requisitos de usuario

Los requisitos de usuario se dividen en dos grupos principales según su naturaleza:

- **Requisitos de capacidad:** Recogen todas las funcionalidades y características que debe contener la futura aplicación. Concretamente describen operaciones o secuencias de operaciones que el *software* debe ser capaz de realizar.
- **Requisitos de restricción:** Definen las restricciones impuestas por los usuarios sobre la manera en que el *software* resuelve los requisitos de capacidad.

3.2.1.1. Requisitos de capacidad

| RUC-01 | | | |
|--------------------|--|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Acceso al sistema:</i></p> <p>Los usuarios deben acceder al sistema mediante la introducción de su teléfono y su contraseña.</p> | | |

Tabla 2: RUC 'Acceso al sistema'

| RUC-02 | | | |
|--------------------|---|--------------------|----------|
| Prioridad | Media | Fuente | Analista |
| Necesidad | Deseable | Estabilidad | Alta |
| Descripción | <p><i>Desconexión del sistema:</i></p> <p>Los usuarios podrán cerrar su sesión en el sistema.</p> | | |

Tabla 3: RUC 'Desconexión del sistema'

| RUC-03 | | | |
|--------------------|---|--------------------|----------|
| Prioridad | Media | Fuente | Analista |
| Necesidad | Opcional | Estabilidad | Alta |
| Descripción | <p><i>Modificar contraseña:</i></p> <p>Los usuarios podrán modificar su contraseña siempre que quieran.</p> | | |

Tabla 4: RUC 'Modificar contraseña'

| RUC-04 | | | |
|------------------|----------|--------------------|----------|
| Prioridad | Media | Fuente | Analista |
| Necesidad | Opcional | Estabilidad | Media |

| | |
|--------------------|---|
| Descripción | <p><i>Modificar nombre:</i></p> <p>Los usuarios podrán modificar su nombre siempre que quieran.</p> |
|--------------------|---|

Tabla 5: RUC 'Modificar nombre'

| RUC-05 | | | |
|--------------------|---|--------------------|----------|
| Prioridad | Media | Fuente | Analista |
| Necesidad | Opcional | Estabilidad | Media |
| Descripción | <p><i>Modificar apellido:</i></p> <p>Los usuarios podrán modificar su apellido siempre que quieran.</p> | | |

Tabla 6: RUC 'Modificar apellido'

| RUC-06 | | | |
|--------------------|--|--------------------|----------|
| Prioridad | Media | Fuente | Analista |
| Necesidad | Opcional | Estabilidad | Alta |
| Descripción | <p><i>Modificar correo electrónico:</i></p> <p>El administrador podrá modificar el correo electrónico con el que se registra a los usuarios en la aplicación siempre que quiera.</p> | | |

Tabla 7: RUC 'Modificar correo electrónico'

| RUC-07 | | | |
|--------------------|--|--------------------|---------|
| Prioridad | Media | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Gestionar usuarios:</i></p> <p>El administrador se encargará de gestionar a todos los usuarios que deseen hacer uso de la aplicación, por lo que podrá:</p> <ul style="list-style-type: none"> • Dar de alta usuarios. • Modificar datos de los usuarios. • Consultar los usuarios que existen en el sistema. | | |

| | |
|--|---|
| | <ul style="list-style-type: none"> • Dar de baja usuarios. |
|--|---|

Tabla 8: RUC 'Gestionar usuarios'

| RUC-08 | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Media | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Gestionar agenda:</i></p> <p>El usuario podrá gestionar los contactos de su agenda, por lo que podrá:</p> <ul style="list-style-type: none"> • Crear un nuevo contacto • Modificar contacto. • Consultar agenda. • Borrar contacto. | | |

Tabla 9: RUC 'Gestionar agenda'

| RUC-09 | | | |
|--------------------|--|--------------------|---------|
| Prioridad | Media | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Gestionar mensajes:</i></p> <p>El usuario podrá gestionar los mensajes programados, por lo que podrá:</p> <ul style="list-style-type: none"> • Crear un nuevo mensaje. • Modificar mensaje (texto o destinatarios). • Consultar mensajes programados. • Borrar mensajes. | | |

Tabla 10: RUC 'Gestionar mensajes programados'

3.2.1.2. Requisitos de restricción

| RUR-01 |
|--------|
|--------|

| | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Deseable | Estabilidad | Alta |
| Descripción | <p><i>Entorno amigable:</i></p> <p>La interfaz gráfica de la aplicación debe ser de fácil manejo e intuitiva, ya que será utilizada por usuarios que no tienen la obligación de tener conocimientos informáticos.</p> | | |

Tabla 11: RUR 'Entorno amigable'

| RUR-02 | | | |
|--------------------|--|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Entorno seguro:</i></p> <p>Los usuarios deben trabajar en un entorno seguro. Será necesario antes de entrar en el sistema que se identifiquen introduciendo teléfono y contraseña.</p> | | |

Tabla 12: RUR 'Entorno seguro'

| RUR-03 | | | |
|--------------------|--|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Opcional | Estabilidad | Media |
| Descripción | <p><i>Entorno operativo:</i></p> <p>La aplicación será compatible con los navegadores Microsoft Internet Explorer 8, Mozilla Firefox 3, Chrome o superior.</p> | | |

Tabla 13: RUR 'Entorno operativo'

| RUR-04 | | | |
|--------------------|-----------------------|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Baja |
| Descripción | <p><i>Idioma:</i></p> | | |

| | |
|--|---|
| | La interfaz gráfica del sistema se encuentra en castellano. |
|--|---|

Tabla 14: RUR 'Idioma'

| RUR-05 | | | |
|--------------------|---|--------------------|----------|
| Prioridad | Alta | Fuente | Analista |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Arquitectura del sistema:</i></p> <p>El sistema seguirá una arquitectura en tres capas (presentación, negocio y datos) consiguiendo una mayor reutilización y modularidad de la aplicación.</p> | | |

Tabla 15: RUR 'Arquitectura del sistema'

| RUR-06 | | | |
|--------------------|--|--------------------|----------|
| Prioridad | Alta | Fuente | Analista |
| Necesidad | Deseable | Estabilidad | Alta |
| Descripción | <p><i>Sistema operativo:</i></p> <p>La aplicación servidora, debe funcionar en Unix.</p> | | |

Tabla 16: RUR 'Sistema operativo'

3.2.2. Requisitos Software

Los requisitos software describen principalmente lo que debe de hacer el sistema y tienen un alto grado de importancia en cuanto a verificación y seguimiento de la aplicación, ya que ésta deberá cubrir todos y cada uno de los requisitos identificados para afirmar que está completa y el funcionamiento de los mismos se comprobará mediante el plan de pruebas.

Antes de comenzar a enumerar los requisitos es conveniente dar una definición de cada uno de los grupos en los que se engloban:

- **Requisitos funcionales (RSF):** Definen qué tiene que hacer el sistema.
- **Requisitos de interfaz (RSI):** Definen el modo en que interactúan y se comunican las interfaces y diferentes módulos del sistema.

- **Requisitos de recursos (RSR):** Define los recursos mínimos para que la aplicación pueda ponerse en marcha.
- **Requisitos de comprobación (RSC):** Define todas las comprobaciones que deberá realizar el sistema para que la interacción usuario-sistema sea la adecuada.
- **Requisitos de documentación (RSD):** Define las características que debe cumplir la documentación de la aplicación.
- **Requisitos de seguridad (RSS):** Define las características que debe cumplir la aplicación para ser segura.

3.2.2.1. Requisitos *Software* funcionales

| RSF-01 | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <i>Identificarse:</i> Para acceder al sistema, los usuarios deben identificarse introduciendo su teléfono y su contraseña. | | |

Tabla 17: RSF 'Identificarse'

| RSF-02 | | | |
|--------------------|--|--------------------|----------|
| Prioridad | Media | Fuente | Analista |
| Necesidad | Deseable | Estabilidad | Alta |
| Descripción | <i>Cerrar sesión:</i> Cualquier usuario que se haya identificado en el sistema, podrá cerrar su sesión. | | |

Tabla 18: RSF 'Cerrar sesión'

RSF-03

| | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Media | Fuente | Cliente |
| Necesidad | Opcional | Estabilidad | Alta |
| Descripción | <p><i>Cambiar contraseña:</i></p> <p>Cualquier usuario que se haya identificado en el sistema, tiene la posibilidad de cambiar su contraseña de acceso. Para cambiar la contraseña el usuario debe introducir su antigua contraseña y dos veces la nueva.</p> | | |

Tabla 19: RSF 'Cambiar contraseña'

| RSF-04 | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Media | Fuente | Cliente |
| Necesidad | Opcional | Estabilidad | Alta |
| Descripción | <p><i>Consultar usuario:</i></p> <p>El administrador podrá consultar los datos de los usuarios con el fin de visualizarlos, modificarlos o incluso borrarlos.</p> | | |

Tabla 20: RSF 'Consultar usuario'

| RSF-05 | | | |
|--------------------|--|--------------------|---------|
| Prioridad | Media | Fuente | Cliente |
| Necesidad | Opcional | Estabilidad | Media |
| Descripción | <p><i>Borrar usuario:</i></p> <p>Tras localizar a un usuario en la lista de usuario ordenada alfabéticamente por el apellido, el administrador podrá borrarlo si lo desea.</p> | | |

Tabla 21: RSF 'Borrar usuario'

| RSF-06 | | | |
|------------------|----------|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |

| | |
|--------------------|--|
| Descripción | <p><i>Crear mensaje:</i></p> <p>Un usuario podrá crear tantos mensajes programados como le permita el sistema. Tendrá que indicar los destinatarios, el contenido del mensaje y la fecha (con hora) en que será enviado.</p> |
|--------------------|--|

Tabla 22: RSF 'Crear mensaje'

| RSF-07 | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Listar mensajes:</i></p> <p>Al entrar en la página de VoiceMemo el usuario visualizará una lista de sus mensajes. En esa lista podrá seleccionar el mensaje para borrarlo o para modificarlo.</p> | | |

Tabla 23: RSF 'Listar mensajes'

| RSF-08 | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Modificar mensaje:</i></p> <p>El usuario podrá modificar el contenido, los destinatarios y la fecha de envío del mensaje.</p> | | |

Tabla 24: RSF 'Modificar mensaje'

| RSF-09 | | | |
|--------------------|--|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Borrar mensaje:</i></p> <p>El usuario podrá borrar cualquier mensaje, tanto los enviados como los no enviados. Si borra un mensaje que no ha sido enviado éste ya no será enviado.</p> | | |

Tabla 25: RSF 'Borrar mensaje'

| RSF-10 | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Crear contacto:</i></p> <p>El usuario podrá añadir un nuevo contacto con su número de teléfono. El contacto constará de nombre y de número de teléfono.</p> | | |

Tabla 26: RSF 'Crear contacto'

| RSF-11 | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Modificar contacto:</i></p> <p>El usuario podrá cambiar el nombre o el número de teléfono de un contacto.</p> | | |

Tabla 27: RSF 'Modificar contacto'

| RSF-12 | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Borrar contacto:</i></p> <p>El usuario podrá eliminar un contacto seleccionado.</p> | | |

Tabla 28: RSF 'Borrar contacto'

3.2.2.2. Requisitos software de interfaz

RSI-01

| | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Deseable | Estabilidad | Alta |
| Descripción | <p><i>Entorno amigable:</i></p> <p>La interfaz gráfica de la aplicación debe ser de fácil manejo e intuitiva, ya que será utilizada por usuarios que pueden o no contar con conocimientos informáticos.</p> | | |

Tabla 29: RSI 'Entorno amigable'

| RSI-02 | | | |
|--------------------|--|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Baja |
| Descripción | <p><i>Idioma:</i></p> <p>La interfaz gráfica del sistema se encuentra en castellano.</p> | | |

Tabla 30: RSI 'Idioma'

| RSI-03 | | | |
|--------------------|--|--------------------|---------------|
| Prioridad | Alta | Fuente | Administrador |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Menú:</i></p> <p>La interfaz gráfica mantendrá un menú con todas las acciones que puede realizar el usuario (en el caso del administrador será con las acciones que puede realizar el administrador)</p> | | |

Tabla 31: RSI 'Menú'

3.2.2.3. Requisitos *software* de recursos

| RSR-01 | | | |
|------------------|----------|--------------------|---------------|
| Prioridad | Alta | Fuente | Administrador |
| Necesidad | Esencial | Estabilidad | Alta |

| | |
|--------------------|---|
| Descripción | <p><i>Software:</i></p> <p>Se utilizará MySQL como Gestor de Bases de Datos como pasarela entre los almacenes de datos y la <i>Web</i>.</p> |
|--------------------|---|

Tabla 32: RSR 'Software'

| RSR-02 | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Opcional | Estabilidad | Media |
| Descripción | <p><i>Entorno operativo:</i></p> <p>La aplicación será compatible con los navegadores Microsoft Internet Explorer 8, Mozilla Firefox 3 y Chrome o superior.</p> | | |

Tabla 33: RSR 'Entorno operativo'

| RSR-03 | | | |
|--------------------|--|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Baja |
| Descripción | <p><i>Idioma:</i></p> <p>La interfaz gráfica del sistema se encuentra en castellano.</p> | | |

Tabla 34: RUR 'Idioma'

| RSR-04 | | | |
|--------------------|--|--------------------|----------|
| Prioridad | Alta | Fuente | Analista |
| Necesidad | Deseable | Estabilidad | Alta |
| Descripción | <p><i>Sistema operativo:</i></p> <p>La aplicación servidora, debe funcionar en Unix.</p> | | |

Tabla 35: RSR 'Sistema operativo'

3.2.2.4. Requisitos *software* de comprobación

| RSC-01 | | | |
|--------------------|--|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Identificador de usuario:</i></p> <p>Será un número de 4 dígitos facilitado por la aplicación. El identificador de usuario será único.</p> | | |

Tabla 36: RSC 'Identificador de usuario'

| RSC-02 | | | |
|--------------------|--|--------------------|----------|
| Prioridad | Alta | Fuente | Analista |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Repetir contraseña:</i></p> <p>Se comprobará que las contraseñas introducidas en los campos 'contraseña nueva' y 'repetir contraseña' son iguales.</p> | | |

Tabla 37: RSC 'Repetir contraseña'

| RSC-03 | | | |
|--------------------|--|--------------------|----------|
| Prioridad | Alta | Fuente | Analista |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <p><i>Comprobar correo:</i></p> <p>Comprobar que el correo electrónico introducido tiene el formato adecuado y que detrás de la @ se encuentra un nombre de dominio.</p> | | |

Tabla 38: RSC 'Comprobar correo'

| RSC-04 | | | |
|------------------|------|---------------|----------|
| Prioridad | Alta | Fuente | Analista |

| | | | |
|--------------------|--|--------------------|------|
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <i>Comprobar campos:</i> Comprobar que todos los campos obligatorios han sido rellenados. | | |

Tabla 39: RSC 'Comprobar campos'

3.2.2.5. Requisitos software de documentación

| RSD-01 | | | |
|--------------------|---|--------------------|----------|
| Prioridad | Alta | Fuente | Analista |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <i>Manual de usuario:</i> Se entregará junto al software final un manual de usuario completo y detallado, para cada uno de los perfiles que pueden utilizar la aplicación. | | |

Tabla 40: RSD 'Manual de usuario'

| RSD-02 | | | |
|--------------------|--|--------------------|----------|
| Prioridad | Alta | Fuente | Analista |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <i>Manual de instalación:</i> Se entregará junto al software final un manual de instalación indicado cómo deben ser instaladas y configuradas las herramientas necesarias en el servidor para que funcione correctamente OBELISK. | | |

Tabla 41: RSD 'Manual de instalación'

3.2.2.6. Requisitos software de seguridad

| RSS-01 | | | |
|--------------------|---|--------------------|---------|
| Prioridad | Alta | Fuente | Cliente |
| Necesidad | Esencial | Estabilidad | Alta |
| Descripción | <i>Entorno seguro:</i> Los usuarios deben trabajar en un entorno seguro. Será necesario antes de entrar en el sistema que se identifiquen introduciendo teléfono y contraseña. | | |

Tabla 42: RSS 'Entorno seguro'

4. Diseño

“Primero ve que el diseño sea sabio y justo: averiguado esto, persíguelo resueltamente; no por un rechazo dejes ir el propósito que te has resuelto efectuar.” (William Shakespeare)

4.1. Introducción

La ingeniería de diseño comienza cuando la primera iteración de la ingeniería de requisitos llega a su fin. La finalidad del diseño de *software* es aplicar un conjunto de principios, conceptos y prácticas que conducen al desarrollo de un sistema o producto de alta calidad. La meta del diseño es crear un modelo de *software* que implemente todos los requisitos del cliente de manera correcta y complazca a aquellos que lo usen. Los ingenieros de diseño deben examinar por medio de muchas alternativas de diseño y converger en la solución que mejor cumpla las necesidades de los interesados en el proyecto.

El proceso de diseño avanza de una visión general de *software* a una visión más estrecha que define el detalle requerido para implementar un sistema. El proceso comienza con un enfoque en la arquitectura. Se definen los subsistemas; se establecen mecanismos de comunicación entre los subsistemas; se identifican los componentes; y se desarrolla una descripción detallada de cada componente. (Pressman, 2006)

4.2. Diseño de tres capas

El diseño de tres capas (*Three-tier*) es una arquitectura cliente servidor en la cual la interfaz de usuario, el proceso lógico (modelo de negocio), el almacenamiento

de los datos y el acceso a los mismos son construidos y mantenidos como módulos independientes, frecuentemente en plataformas separadas.

El diseño de tres capas es una arquitectura *software* y un patrón de diseño de *software*.

Aparte de las ventajas del software modular con interfaces bien definidas, la arquitectura de tres capas permite que alguna de las tres capas pueda ser actualizada o reemplazada independientemente de que los requisitos o la tecnología cambien.

Normalmente, la interfaz de usuario se ejecuta en un PC de escritorio o estación de trabajo y utiliza una interfaz de usuario gráfica estándar, la lógica de proceso funcional puede constar de uno o más módulos independientes que se ejecutan en una estación de trabajo o servidor de aplicaciones, y un servidor de base de datos o computadora central contiene la lógica de los datos de almacenamiento. La capa intermedia puede dividirse en múltiples niveles a su vez (en cuyo caso la estructura general se llama una "arquitectura n-capas").

La arquitectura de tres capas posee las siguientes capas:

Capa de presentación

Es el nivel más alto de la aplicación. La capa de presentación muestra información acerca de los servicios tales como navegación entre distintos productos, compras y contenido de la cesta de la compra. Se comunica con las otras capas presentando los datos de salida a la capa del navegador/cliente y todas las demás capas de la red.

Capa de aplicación (Lógica de negocio/Capa lógica/Capa de acceso a los datos/Capa intermedia)

La capa lógica se obtiene de la capa de presentación y controla la funcionalidad de la aplicación mediante la realización de un proceso detallado.

Capa de datos

Esta capa consiste en los servidores de bases de datos. Aquí la información es guardada y recuperada. Esta capa mantiene los datos independientes de los servidores de aplicación o de la lógica de negocio, de esta manera se mejora la escalabilidad y el rendimiento.

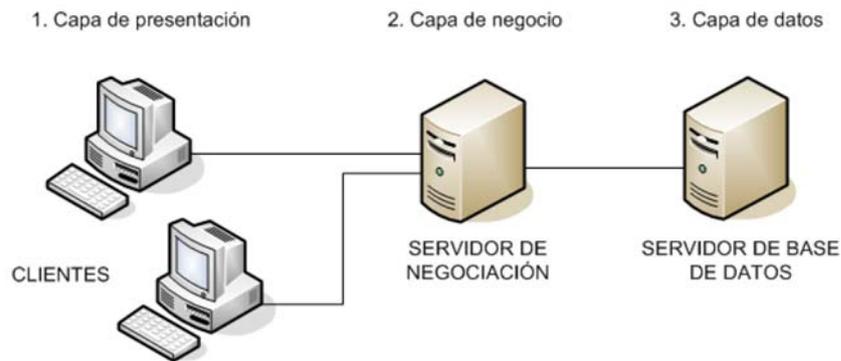


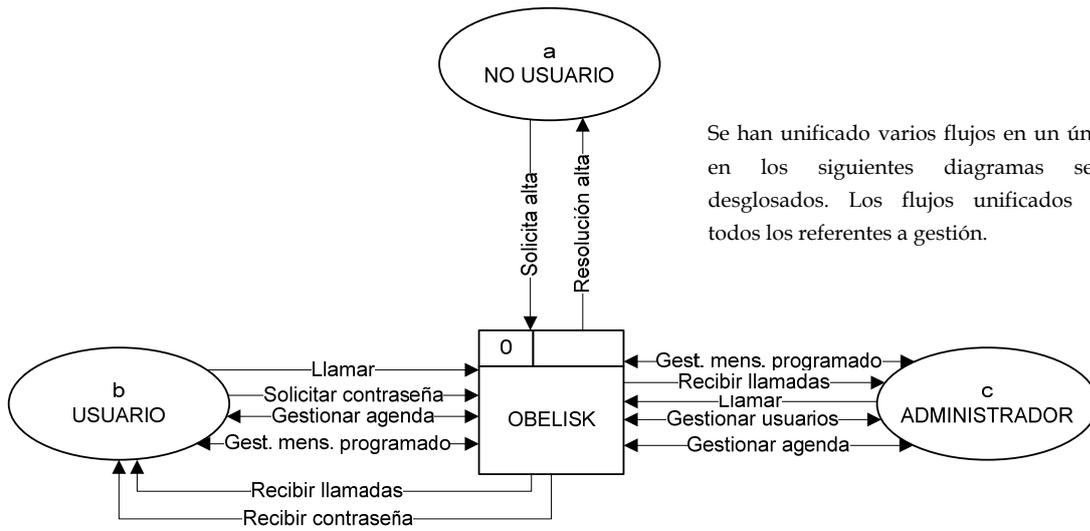
Ilustración 5: Arquitectura de tres capas

4.3. Diagrama de Flujo de Datos (DFD)

Se pretende comprender y contrastar con el cliente lo que el sistema debe ser capaz de hacer. Los ingenieros de *software* han desarrollado notaciones que superan varios de los problemas del lenguaje natural, que van desde las notaciones gráficas con partes de lenguaje hasta las matemáticas. (de Amescua Seco, y otros, 2003)

Un DFD es un diagrama en forma de red que representa el flujo de datos y las transformaciones que se aplican sobre ellos al moverse desde la entrada hasta la salida del sistema. Se utiliza para modelar las funciones del sistema y los datos que fluyen entre ellas a distintos niveles de abstracción. El sistema, por tanto, se modelará mediante un conjunto de DFD nivelados en donde los niveles superiores definen las funciones del sistema de forma general y los niveles inferiores definen estas funciones en niveles más detallados (Piattini, y otros, 2003)

En el diagrama de contexto se localizan todas las entidades externas que van a proporcionar y/o consumir información, así como la composición. En este caso es importante conocer la composición, ya que hay entidades externas que tienen una interfaz muy compleja con OBELISK, por lo que hay que abstraer los datos de forma unitaria, van a aparecer como si fuesen un único flujo, pero en los siguientes DFD se van a desglosar en los flujos que corresponde para que se vea de forma más clara (ver Ilustración 6).



Se han unificado varios flujos en un único, en los siguientes diagramas serán desglosados. Los flujos unificados son todos los referentes a gestión.

Ilustración 6: Diagrama de contexto

En el diagrama de sistema se muestran las funciones principales. Se centra en cada proceso señalando sus entradas y salidas. En este diagrama se pueden ver los diferentes almacenes que se utilizan (ver Ilustración 7).

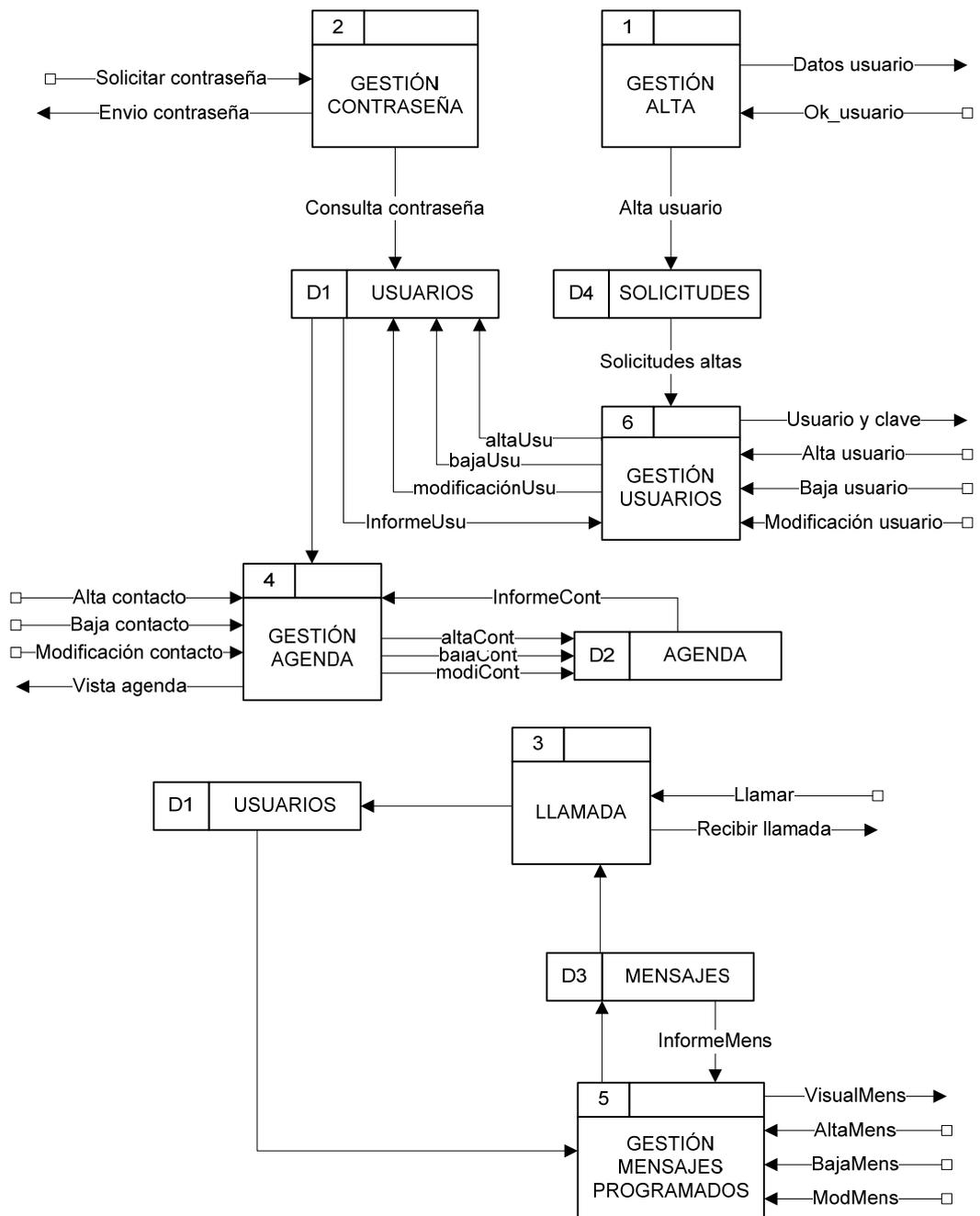


Ilustración 7: Diagrama de Sistema

4.4. Modelo entidad/relación extendido (E/R)

El diseño de bases de datos (BD) consiste en describir la estructura de la BD de forma que se represente fielmente la parcela del mundo real que se quiere

almacenar. Ello se realiza mediante un proceso de abstracción (que se denomina modelado) que se apoya en un modelo de datos. Un modelo de datos es el instrumento que se aplica a un universo del discurso (UD) para obtener una estructura de datos que se denomina esquema de la BD. A lo largo del desarrollo de una BD se utilizan varios modelos de datos que nos permiten representar la realidad según las distintas fases de una metodología y según distintos niveles de abstracción.

El modelo E/R consiste en un conjunto de conceptos, reglas y notaciones que se utiliza en la fase de análisis de requisitos. (de Amescua Seco, y otros, 2003)

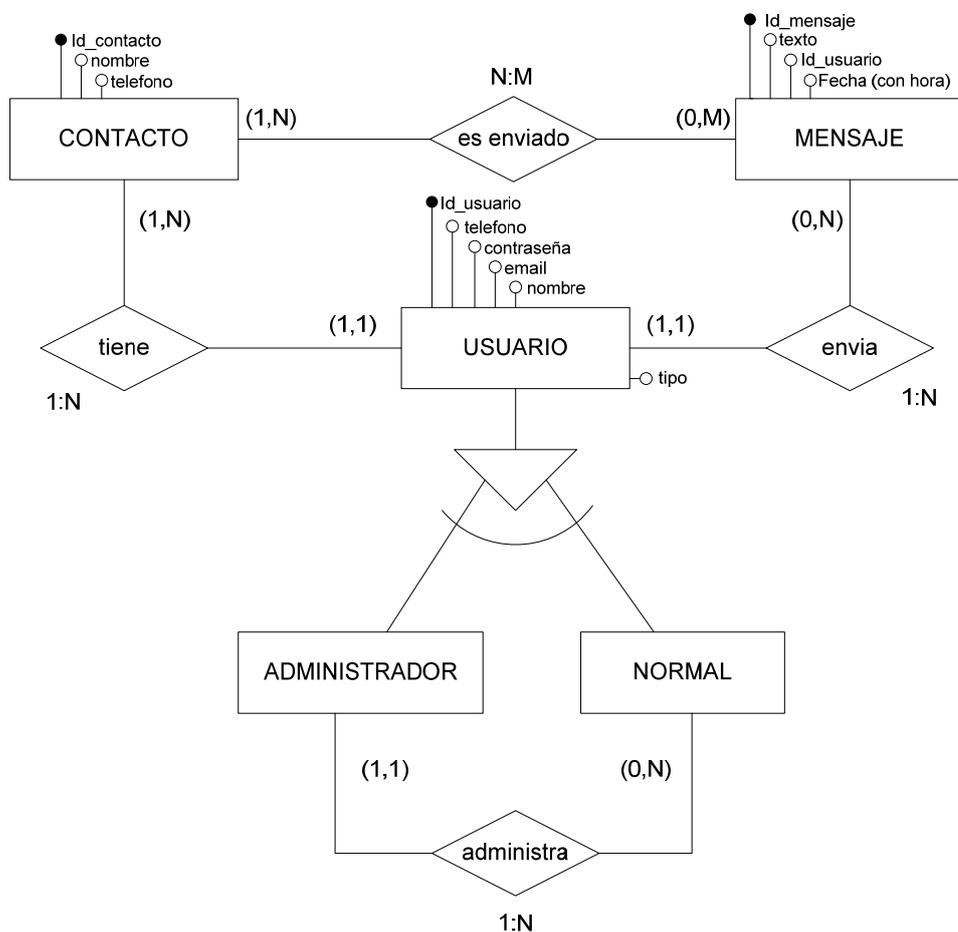


Ilustración 8: Modelo E/R

4.5. Diagramas de navegación

Los diagramas de navegación permiten ver de forma rápida cómo un usuario con unas características determinadas puede realizar determinadas acciones a través de la página *Web* y acceder a unos u otros sitios. La navegación debe ser simple y consistente. También debe estar diseñada de modo que sea intuitiva y predecible. Esto es, el usuario debe entender cómo moverse por la aplicación *Web* sin tener que buscar vínculos o instrucciones de navegación.

En el caso de OBELISK no se da gran importancia a la apariencia visual, ya que se trata de un prototipo y se realizará lo más sencillo posible para que luego sea el cliente el que decida la apariencia de su producto.

A continuación se muestran los diagramas de navegación que corresponden a los diferentes tipos de roles de usuarios. En primer lugar se muestra el diagrama de navegación de alguien que es *no_usuario*, luego se muestra el diagrama de usuario y por último se muestra el diagrama del administrador.

4.5.1. Diagrama de navegación *no_usuario*

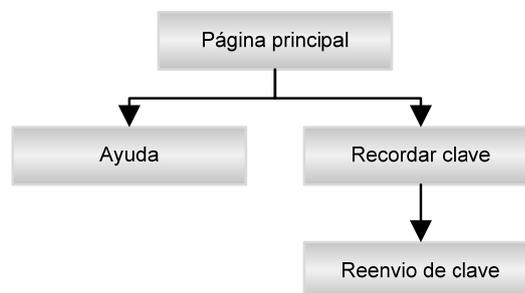


Ilustración 9: Diagrama de navegación de *no_usuario*

4.5.2. Diagrama de navegación usuario

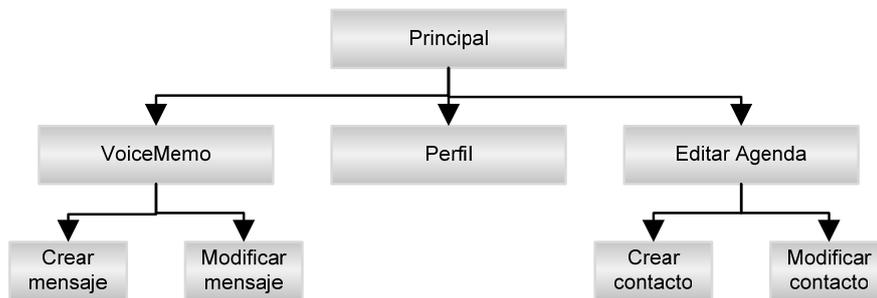


Ilustración 10: Diagrama de navegación de usuario

4.5.3. Diagrama de navegación administrador

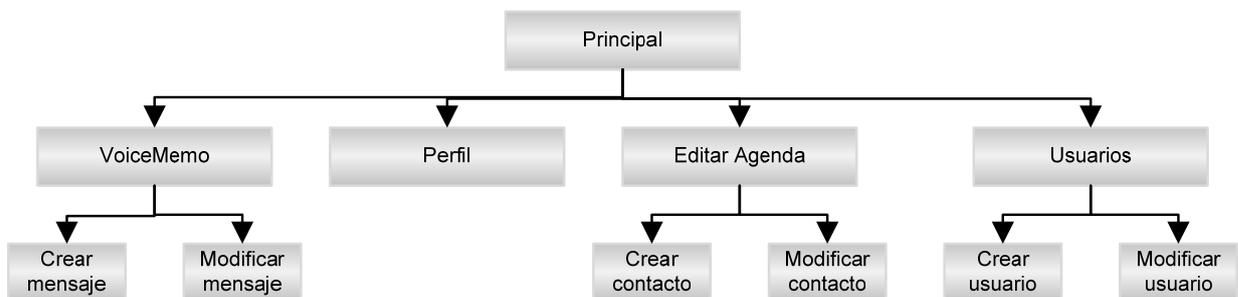


Ilustración 11: Diagrama de navegación de administrador

4.6. Diseño de Base de Datos (modelo relacional)

El modelo relacional tiene como objetivos:

- Independencia física
- Independencia lógica
- Flexibilidad
- Uniformidad
- Sencillez

En este modelo se representa la relación (en forma de tabla) en la que se distinguen los atributos (columnas) y las tuplas (filas), que son las ocurrencias de la relación. El modelo relacional presenta una regla denominada “integridad de entidad”, que establece que “ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo”

En el caso de OBELISK y basándose en el esquema E/R (ver Ilustración 8) se cuentan con las siguientes tablas relacionadas como explica el dibujo siguiente (ver Ilustración 12)

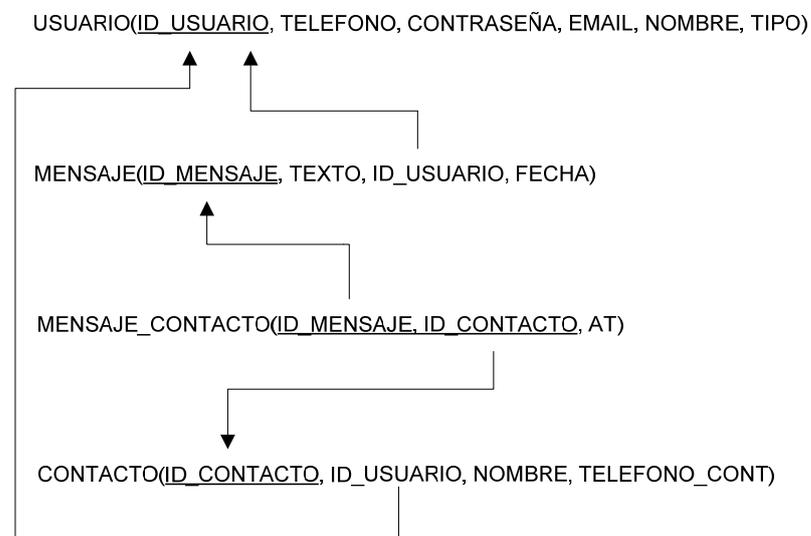


Ilustración 12: Esquema relacional

5. Implementación e implantación

5.1. Descripción de la implementación

En este apartado se va a tratar la manera en la que intervienen cada uno de los componentes del proyecto en el proyecto, y su forma de trabajar en cada momento y como se integran entre ellos.

La aplicación está desarrollada en PHP, HTML y JavaScript. La mayor parte del código es PHP.

5.1.1. Interfaz con Base de Datos

La interfaz *Web* y la base de datos están relacionadas mediante un fichero en PHP que hace todas las gestiones necesarias con la base de datos. El fichero se llama `acceso_bbdd.inc` que es utilizado como una librería en todos los ficheros en los que es necesario realizar gestiones con la base de datos.

5.1.2. Base de Datos con Asterisk

Se necesita la relación de estos dos componentes tanto para la generación de los ficheros de usuarios de Asterisk como para la generación de los ficheros necesarios para realizar las llamadas programadas. Cada vez que se realiza un cambio en la base de datos que debe estar registrado en los ficheros de Asterisk, éstos se regeneran. El fichero encargado de generar los archivos de configuración de Asterisk se llama `generarUsuariosAsterisk.inc`, este fichero, al igual que el anterior, es una librería y sus funciones son llamadas en el momento en el que existe un cambio en la base de datos. La función de este

fichero es recorrer toda la base de datos y generar los ficheros de configuración de Asterisk con los nuevos cambios.

5.1.3. Integración de Asterisk y Festival

Esta combinación de componentes se utiliza para la funcionalidad de los mensajes programados (VoiceMemo). Cuando un usuario decide crear un mensaje para realizar una llamada programada a un número determinado de contactos, automáticamente se registra en la base de datos.

A su vez se crean varios directorios:

- *mensajes/* : en este directorio se generan tanto los ficheros de texto como los ficheros de audio, se genera un único fichero de texto y un único fichero de audio por mensaje. La descripción de los ficheros es la siguiente:
 - Se genera el fichero de texto: este fichero de texto contiene el mensaje escrito por el usuario y el nombre del usuario. Con este fichero y utilizando Festival se generará el archivo de audio.
 - Se genera el fichero de audio: este fichero es generado mediante Festival, para que pueda ser reproducido por Asterisk en una llamada telefónica está grabado a 8000Hz. El fichero de audio es llamado desde Asterisk y es reproducido en la llamada.
- *call/* : en este directorio se generan los archivos *.call*, que son los encargados de indicarle a Asterisk a quién tiene que llamar y qué fichero de audio es el que tiene que reproducir. Se genera un *.call* por mensaje y contacto.
- *script/* : en este directorio se crean los *scripts*, estos ficheros son los encargados de mover los ficheros *.call* a la cola de Asterisk para ser reproducidos. Como queremos que los *scripts* se ejecuten a una fecha y hora determinada (la indicada por el usuario) pues se utiliza el comando 'at' al cual se le pasan los parámetros indicándole la hora y el *script* que debe ejecutar. Se genera un *script* por mensaje y contacto.

Toda esta gestión de ficheros se realiza en la librería *audio.inc*.

5.2. Implantación

En este apartado se va a hablar de cómo se pretende implantar en una empresa la nueva aplicación y qué es necesario para ello.

Para comenzar con esta nueva aplicación en una empresa empezariamos con un grupo reducido de usuarios, es decir, se comenzaría con una fase de prueba en la empresa, para que puedan comprobar si los usuarios se adaptan fácilmente a la nueva aplicación y que vean que realmente funciona como se les ha explicado, sin necesidad ninguna de tener que invertir en nuevo material, exceptuando la instalación del servidor. Más adelante, se pasaría a la migración completa a la nueva aplicación.

Antes de todo esto se realizará un estudio de lo que existe actualmente en la empresa. Si disponen de terminales IP se pueden seguir manteniendo a la vez que la nueva aplicación, ya que son totalmente compatible. OBELISK no está pensado para sustituir a los terminales de VoIP sino que está pensado para: facilitar la movilidad de sus empleados pudiendo atender a sus llamadas en cualquier lugar; facilitar la gestión de usuarios; y proporcionando una nueva funcionalidad de llamadas programadas.

Para poder llevar a cabo este proyecto se precisa disponer de una máquina que trabaje como servidor. Tal y como está desarrollado el proyecto es necesario disponer de un servidor con el siguiente software:

- Ubuntu
- Asterisk (centralita de VoIP)
- Festival (text to speech)
- Apache (servidor de páginas *Web*)
- MySQL (base de datos)
- PHP (desarrollo)

5.2.1. Ubuntu

Instalación de Ubuntu:

- Se descarga el sistema operativo:
<http://www.ubuntu.com/desktop/get-ubuntu/download>
- Se instala.

5.2.2. Asterisk

Instalación de Asterisk:

- Se puede instalar utilizando `apt-get install`, pero se recomienda utilizar *Synaptic Package Manager* ya que no hay que indicar la versión, directamente descarga la más estable e indica las dependencias de otros paquetes, que también se deben instalar.

5.2.3. Festival

Instalación de Festival:

- Se instala utilizando *Synaptic Package Manager*.
- Descargamos una de las voces de Guadalinex de http://forja.guadalinex.org/frs/?group_id=21&release_id=109 en el caso de OBELISK se ha optado por la voz de Pedro. Se descomprime el paquete y se instala.
- Se modifica el fichero `/usr/share/festival/voices.scm` para indicar cuál es la voz que se quiere utilizar de la siguiente manera:

```
325 (defvar default-voice-priority-list
326 '(kal_diphone
327   cmu_us_bdl_arctic_hts
328   cmu_us_jmk_arctic_hts
329   cmu_us_slt_arctic_hts
330   cmu_us_awb_arctic_hts
331   ; cstr_rpx_nina_multisyn ; restricted license (lexicon)
332   ; cstr_rpx_jon_multisyn ; restricted license (lexicon)
333   ; cstr_edi_awb_arctic_multisyn ; restricted license (lexicon)
334   ; cstr_us_awb_arctic_multisyn
335   ked_diphone
336   don_diphone
337   rab_diphone
338   en1_mbrola
339   us1_mbrola
340   us2_mbrola
341   us3_mbrola
342   gsw_diphone ;; not publically distributed
343   ; et_diphone
344   JAndalucia_Indisys_MP_es_pa_diphone
345 )
```

Ilustración 13: Fichero para la configuración de Festival

5.2.4. Apache, MySQL y PHP (LAMP¹³)

Instalación de Sistema LAMP (se va a realizar desde el terminal):

- `sudo apt-get install apache2`
- El directorio en el que se almacenan los documentos es `/var/www`
- `sudo apt-get install php5 libapache2-mod-php5 php5-cli php5-mysql`
- Se reinicia Apache con: `sudo /etc/init.d/apache2 restart`
- `sudo apt-get install mysql-server`
- `sudo apt-get install mysql-client mysql-admin mysql-query-browser libmysqlclient15-dev`
- `sudo apt-get install phpmyadmin`

Ya estaría instalado todo lo necesario para poner en funcionamiento OBELISK.

¹³ LAMP: Linux, Apache, MySQL y PHP. El acrónimo 'LAMP' se refiere a un conjunto de subsistemas de *software* necesarios para alcanzar una solución global, en este caso configurar sitios *Web* o Servidores dinámicos con un esfuerzo reducido. La combinación de estas tecnologías es usado primariamente para definir la infraestructura de un servidor *Web*, utilizando un paradigma de programación para desarrollo.

6. Presupuesto

6.1. Planificación

En este apartado se muestra el tiempo empleado para el desarrollo del proyecto. Dicha planificación se ha realizado mediante la herramienta Microsoft Project 2007, y se detalla el tiempo empleado para cada fase del proyecto.

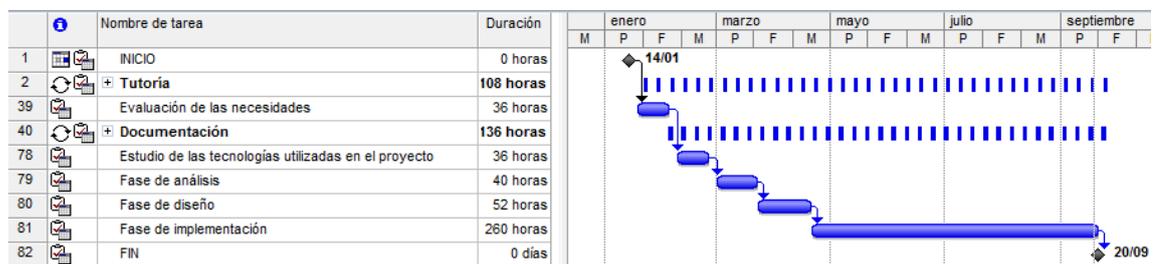


Ilustración 14: Diagrama Gantt de OBELISK

Con un total de 632 horas.

6.2. Presupuesto

En este apartado se muestran los costes del proyecto. Se detallarán las herramientas *software* utilizadas, herramientas *hardware*, recursos humanos, etc. exponiendo tanto su coste económico como su funcionalidad en este proyecto.

6.2.1. Recursos materiales

- **Herramientas software:**
 - **VMWare Workstation:** Herramienta utilizada para montar el servidor *Web*. Coste 153,14€
 - **Ubuntu:** Distribución de Linux Ubuntu 9.10. Sistema operativo utilizado en la máquina donde se desarrolla el proyecto. Coste 0€.
 - **VIM:** Herramienta utilizada para desarrollar el código fuente de toda la aplicación. Coste 0€.
 - **PHP:** Herramientas utilizadas para desarrollar aplicaciones *Web* en lenguaje PHP, HTML y JavaScript. Coste 0€.
 - **MySQL:** Base de Datos utilizada. Coste 0€
 - **Apache:** Servidor *Web*. Coste 0€
 - **Asterisk:** Centralita de Voz IP utilizada para la comunicación entre usuarios de la aplicación y núcleo central del proyecto, ya que parte del proyecto desarrollado sirve para simplificar el alta de los usuarios en esta centralita. Coste 0€.
 - **Festival:** Software utilizado para la conversión del texto a voz en las llamadas programadas. Coste 0€.
 - **X-Lite:** Emulador de teléfono (Softphone) utilizado para realizar las pruebas de comunicación entre la aplicación *Web* y un teléfono que no esté utilizando la aplicación *Web*.
 - **Microsoft Office:** Utilizado para la elaboración de la memoria. Coste 129,99€.
 - **Microsoft Project 2007:** Herramienta utilizada para realizar la estimación de costes y tiempo de cada fase del proyecto. Coste 799€.
 - **Adobe Acrobat Writer 5:** Herramienta utilizada para crear un fichero con el formato adecuado para poder ser visualizado por la aplicación Adobe Acrobat Reader. Coste 160€.

- **Herramientas hardware:**
 - Ordenador personal con las siguientes características:
 - Microprocesador Pentium® Dual-Core T4200.
 - 4GB de Memoria RAM.
 - 300GB de Disco duro.

Precio: 599€

| Recursos | Total |
|------------------------------|------------------|
| Herramientas <i>Software</i> | 1.242,13€ |
| Herramientas <i>Hardware</i> | 599€ |
| TOTAL | 1.841,13€ |

Tabla 43: Recursos materiales

6.2.2. Recursos humanos

En la elaboración del proyecto se requiere personal informático cualificado, el cual deberá adoptar en determinados momentos diferentes tipos de roles que se adapten a cada una de las actividades que forman el proyecto. La siguiente tabla muestra el coste de cada rol utilizado en la elaboración del proyecto:

| Cargo | Sueldo Neto | Sueldo Bruto | Coste/Hora |
|------------------------------|---------------|--------------|------------|
| Analista/Diseñador | 1.552€/mes | 28.000€/año | 25€ |
| Responsable de Documentación | 1.241,60€/mes | 22.400€/año | 20€ |
| Programador | 1.241,60€/mes | 22.400€/año | 20€ |

Tabla 44: Salarios por categoría

- Coste/Hora indica el sueldo bruto en una hora de trabajo.
- Sueldo Bruto indica el sueldo bruto anual, con 14 pagas mensuales.
- Sueldo Neto indica el sueldo neto mensual. Descontando el IRPF (20%) y Seguridad Social (2,4%).
- Para obtener los sueldo se han consultado fuentes de páginas *Web* de ofertas de empleo (Infojobs, Tecnoempleo) y se ha consultado con distintos compañeros trabajando en empresas para contrastar la información.

Tenemos en cuenta que:

- La jornada laboral es de 4 horas diarias.

- Son 20 los días laborales al mes.

Este proyecto ha sido realizado por un informático, el cual adoptó roles distintos para desarrollar cada una de las actividades que forman el proyecto. La siguiente tabla muestra el coste total de cada rol.

| Cargo | Horas | Coste/Hora | Total |
|------------------------------|-------|------------|----------------|
| Analista/Diseñador | 272 | 25€ | 6.800€ |
| Responsable de Documentación | 136 | 20€ | 2.720€ |
| Programador | 260 | 20€ | 5.200€ |
| TOTAL | | | 14.720€ |

Tabla 45: Gastos de personal imputables al proyecto

6.2.3. Gastos indirectos

La siguiente tabla muestra el total de los gastos indirectos que repercutieron en los gastos totales para el desarrollo del proyecto.

| Descripción | Coste |
|--------------------------------|-----------------------------------|
| Productos de limpieza | Incluido en los costes indirectos |
| Electricidad | Incluido en los costes indirectos |
| Agua | Incluido en los costes indirectos |
| Alquiler del local | Incluido en los costes indirectos |
| Amortización inmobiliario | Incluido en los costes indirectos |
| Gastos de comunidad | Incluido en los costes indirectos |
| Costes de estructura | Incluido en los costes indirectos |
| Costes indirectos (20%) | 14.720 * 0,20 = 2.944€ |

Tabla 46: Gastos indirectos

6.2.4. Resumen del presupuesto

En este apartado se muestra una tabla con los gastos totales del proyecto y la suma de los mismos.

| Descripción | Coste |
|---------------------|-------------------|
| Recursos Materiales | 1.841,13€ |
| Recursos Humanos | 14.720€ |
| Costes Indirectos | 2.944€ |
| TOTAL | 19.505,13€ |

Tabla 47: Resumen del presupuesto

Todos estos datos han sido calculados utilizando las plantillas de rúbrica.

7. Conclusiones y trabajos futuros

7.1. Conclusiones

7.1.1. Sobre el proyecto

Tras la realización del proyecto, se han obtenido las siguientes conclusiones:

- Se ha definido el problema y se ha ofrecido la base de conocimiento necesaria para su entendimiento en el estado de la cuestión basándose en los conocimientos adquiridos en la carrera en asignaturas relacionadas con Ingeniería del *Software*.
- La solución realizada es una aportación original, no existiendo nada anteriormente igual. Esta solución ha sido posible gracias al estudio y conocimiento de adquirido en asignaturas de Arquitectura de Computadores, Sistemas Operativos, Redes y Bases de Datos.
- El presupuesto es ajustado, contemplando los aspectos necesarios para realizar una estimación real de los costes de realización del mismo.
- Se ha aprendido a realizar aplicaciones *Web*, enmarcadas en la arquitectura cliente-servidor.
- Para la realización del proyecto se han incluido referencias a fuentes solventes y reputadas en el ámbito del mismo, que están disponibles en la bibliografía.

A nivel personal, este proyecto me ha servido para afianzar conocimientos adquiridos durante mis estudios y cómo relacionar todo ello en una misma solución. Por otro lado me ha servido para adquirir nuevos conocimientos sobre VoIP, programación (nuevos lenguajes), bases de datos y servidores *Web*. Ha

sido muy satisfactorio para mí poder ver de esta forma como se relaciona todo entre sí.

7.2. Trabajos futuros

Como trabajo futuro se podrían ampliar los siguientes aspectos:

- Ampliación de la aplicación que controlase el buzón de voz de cada usuario.
- Ampliación que pasase mensajes en el buzón de voz a texto y que lo enviase al correo electrónico.
- Gestión, por parte del usuario, de llamadas perdidas.
- Que exista videoconferencia.
- El poder realizar en una misma llamada una multiconferencia con varios participantes al mismo tiempo.

8. Apéndices

8.1. Apéndice A. Bibliografía

(de Amescua Seco y otros, 2003) de Amescua Seco, Antonio, y otros. 2003. *Análisis y diseño estructurado y orientado a objetos de sistemas informáticos.* Madrid : McGraw Hill, 2003.

(IEEE, 1990) IEEE. 1990. *Computer Dictionary. Compilation of IEEE Standard Computer Glossaries.* Nueva York (EE.UU.) : IEEE Computer Society, 1990. IEEE Std. 610-1990.

(Piattini, y otros, 2003) Piattini, Mario G., y otros. 2003. *Análisis y diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software.* Madrid : Ra-Ma, 2003.

(Pressman, 2006) Pressman, Roger S. 2006. *Ingeniería del Software. Un enfoque práctico.* s.l. : McGraw Hill, 2006.

(Tanenbaum, 2003) Tanenbaum, Andrew S. 2003. *Redes de computadoras.* Mexico : Pearson Educación, 2003.

(Van Meggelen, 2005) Van Meggelen, Smith and Madsen. 2005. *Asterisk. The Future of Telephony.* s.l. : O'REILLY, 2005.

8.2. Apéndice B. Manual de usuario

La persona que entre a la aplicación se va a encontrar con la siguiente pantalla (ver Ilustración 15):

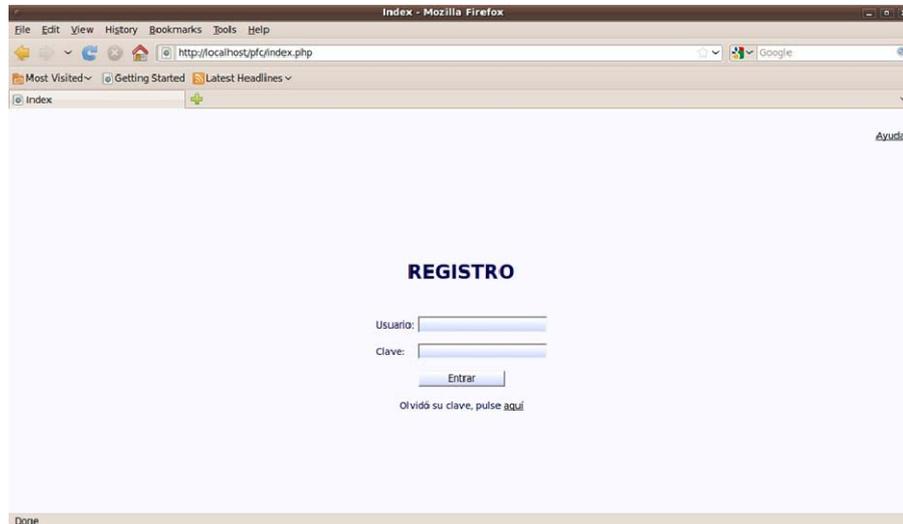


Ilustración 15: Inicio (Manual Usuario)

En esta pantalla debe introducir su usuario (número de teléfono) y la clave. Existe una ayuda que detalla los datos que hay que introducir y también existe un acceso para recordar los datos en el caso de que los hayamos olvidado. En la Ilustración 16 aparecen recuadrados.

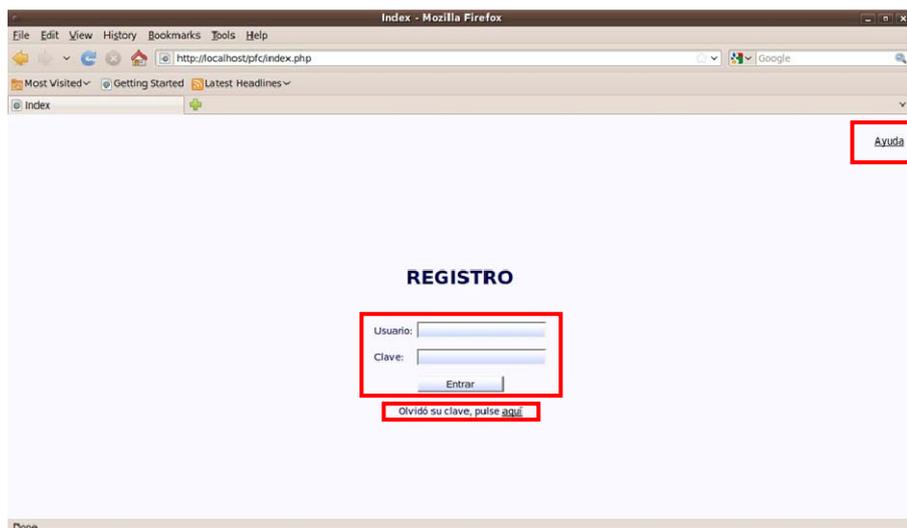


Ilustración 16: Inicio (Manual Usuario)

En la Ilustración 17 se puede ver a las pantallas a las que se accede pulsando las diferentes opciones que ofrece la página de inicio. Si se pulsa en la ayuda, se ofrece una breve información general de lo que debe hacerse para entrar en la aplicación; si se introduce un usuario y una clave correctos la aplicación conduce al usuario a la pantalla principal de usuario; y si el usuario no recuerda su clave y quiere recordarla tiene un *link* para poder volver a solicitar sus datos de acceso a la aplicación.

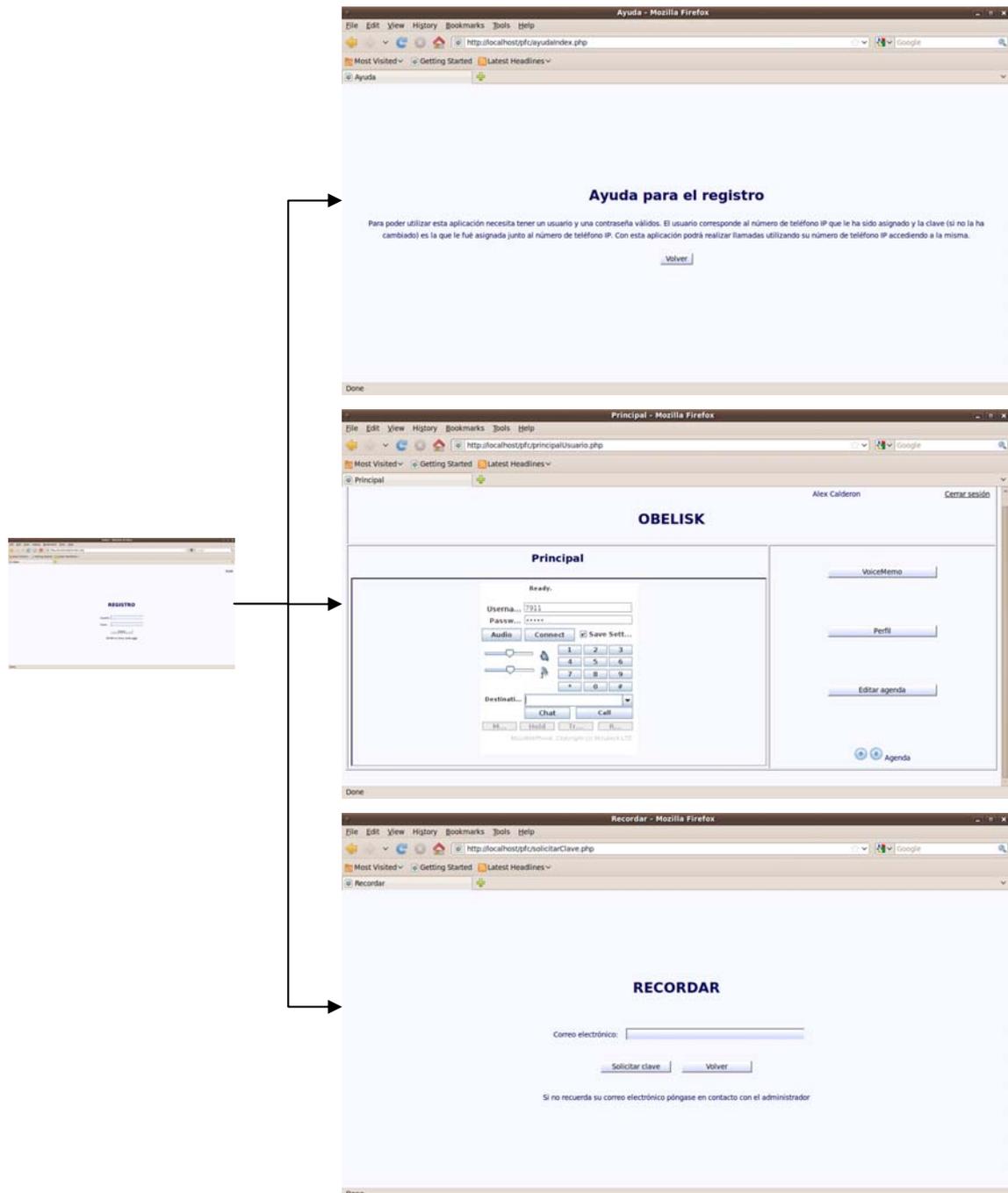


Ilustración 17: Inicio (Manual usuario)

En la pantalla de recordar, que se ve en la Ilustración 17, se escribe el correo electrónico con el que el usuario está registrado, en la Ilustración 18 se muestran las diferentes posibilidades a las que se llegan desde este punto.



Ilustración 18: Recordar contraseña (Manual usuario)

Basándose en la Ilustración 18 se ve que desde Recordar si el usuario introduce un correo electrónico registrado por la aplicación lanza un mensaje y se envía el usuario y la clave al correo electrónico del usuario; si el correo no está registrado lanza un mensaje informando de ello al usuario; y si el formato de correo no es el adecuado se lanza una alarma.

Una vez que el usuario está registrado la primera pantalla con la que se encuentra es la de principal (ver Ilustración 19).

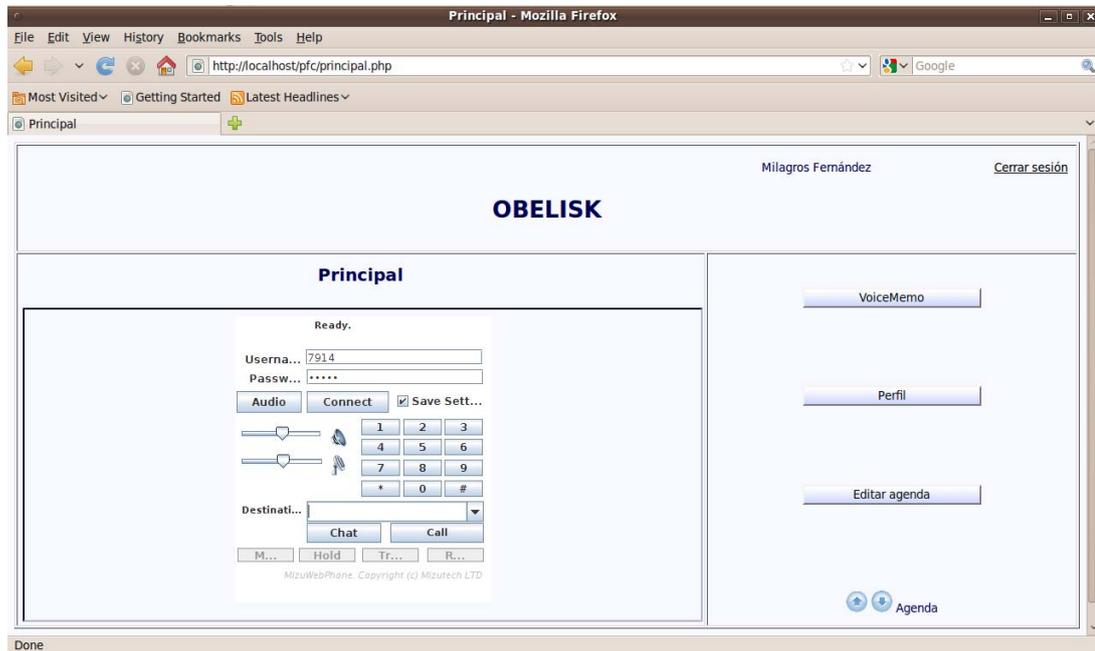


Ilustración 19: Pantalla principal (Manual usuario)

En esta pantalla si se pulsa sobre agenda, se desplegará la agenda y mostrar los contactos de la agenda que tenga el usuario (ver Ilustración 20). Podrá seleccionar cualquiera de ellos y automáticamente se cargará en el teléfono para realizar una llamada.

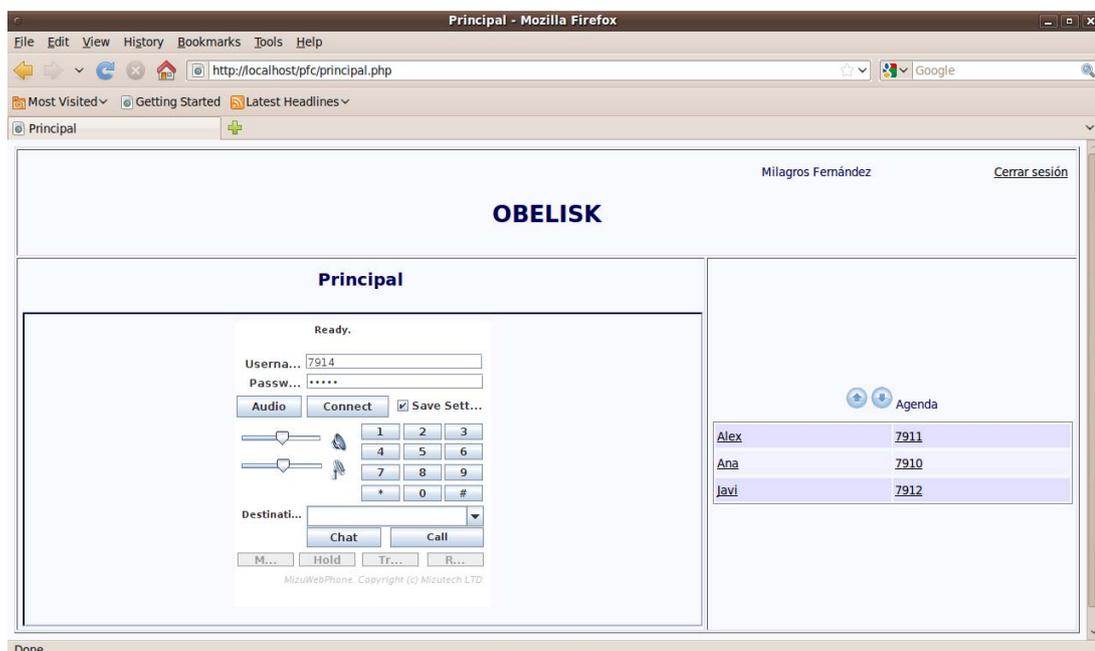


Ilustración 20: Principal con agenda (Manual de usuario)

El usuario desde la pantalla principal puede acceder a las diferentes funcionalidades de la aplicación. En la Ilustración 22 se puede ver un diagrama de a dónde se puede ir desde la pantalla principal.



Ilustración 21: Navegación usuario (Manual usuario)

'VoiceMemo'



Ilustración 22: Navegación usuario I (Manual usuario)

VoiceMemo es una funcionalidad utilizada para enviar un mensaje a uno o varios contactos a una hora determinada un día determinado. Lo que hace es llamar a los contactos seleccionados y les lee el mensaje del usuario que lo envía en la fecha que el usuario haya decidido. En primer lugar, cuando se accede a

VoiceMemo se puede ver un listado de los mensajes que el usuario haya creado (ver

Ilustración 23)

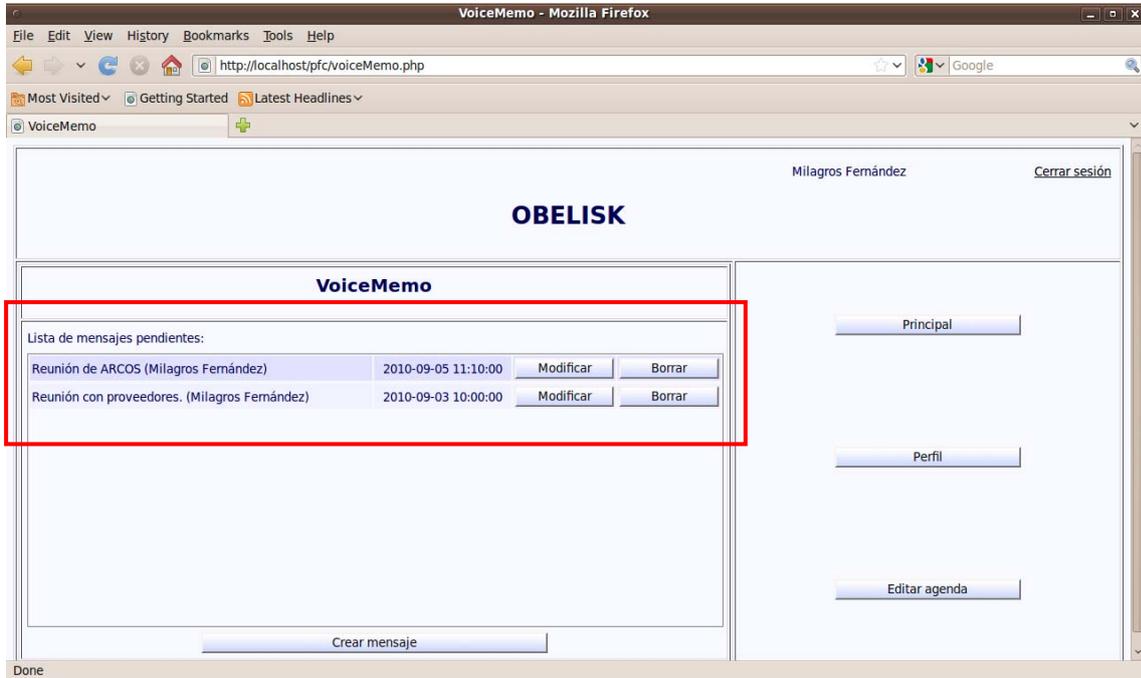


Ilustración 23: VoiceMemo (Manual usuario)

Desde esta pantalla (Ilustración 23) se puede borrar un mensaje, modificarlo o crear uno nuevo. Al igual que ir a cualquier otro sitio de la aplicación. Si se pulsa en 'Crear mensaje' aparecerá una pantalla como la siguiente (ver Ilustración 24):

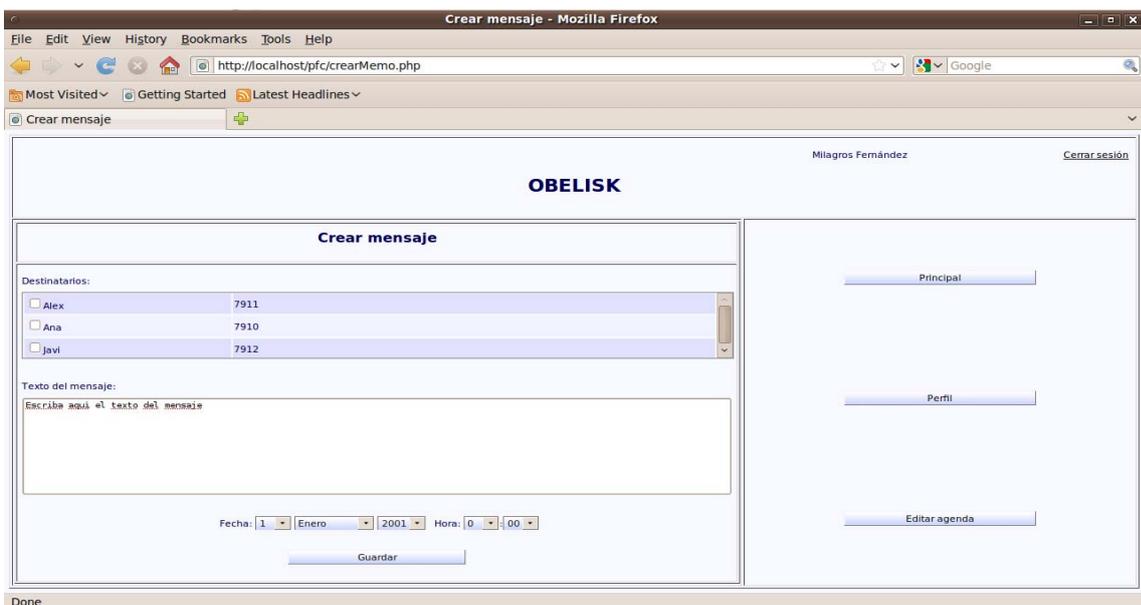


Ilustración 24: Crear mensaje (Manual usuario)

Cuando se pulsa a 'Modificar' se cargan los datos de dicho mensaje en pantalla exceptuando la fecha, como puede verse en la Ilustración 25.

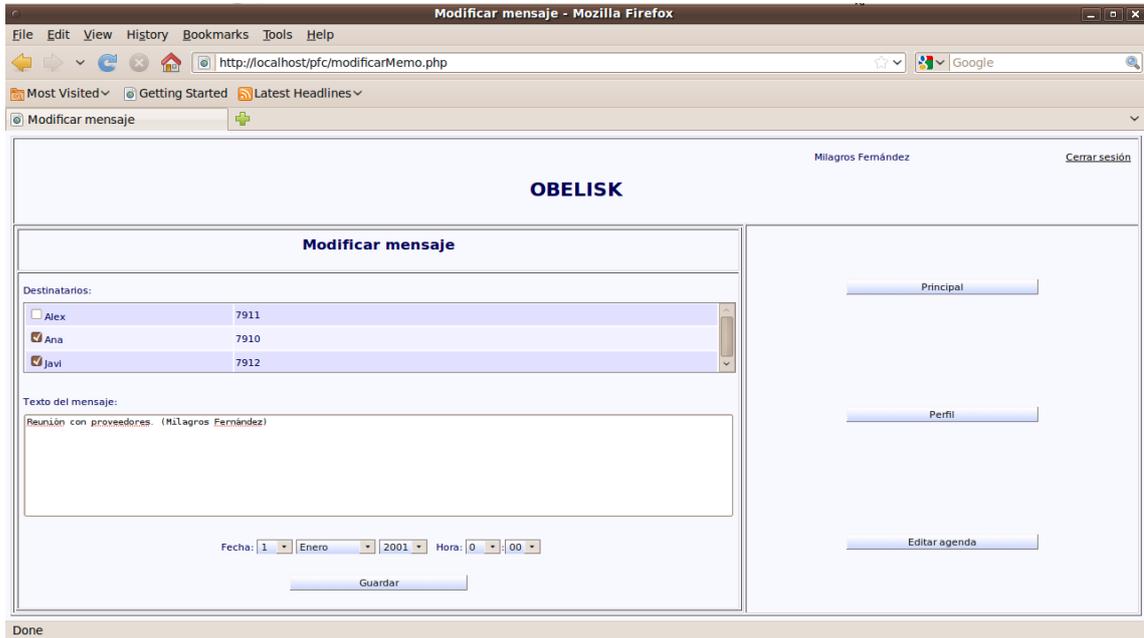


Ilustración 25: Modificar mensaje (Manual usuario)

Si se pulsa a 'Borrar' un mensaje, el mensaje será borrado y se mostrará la lista de mensajes actualizada.

'Perfil'

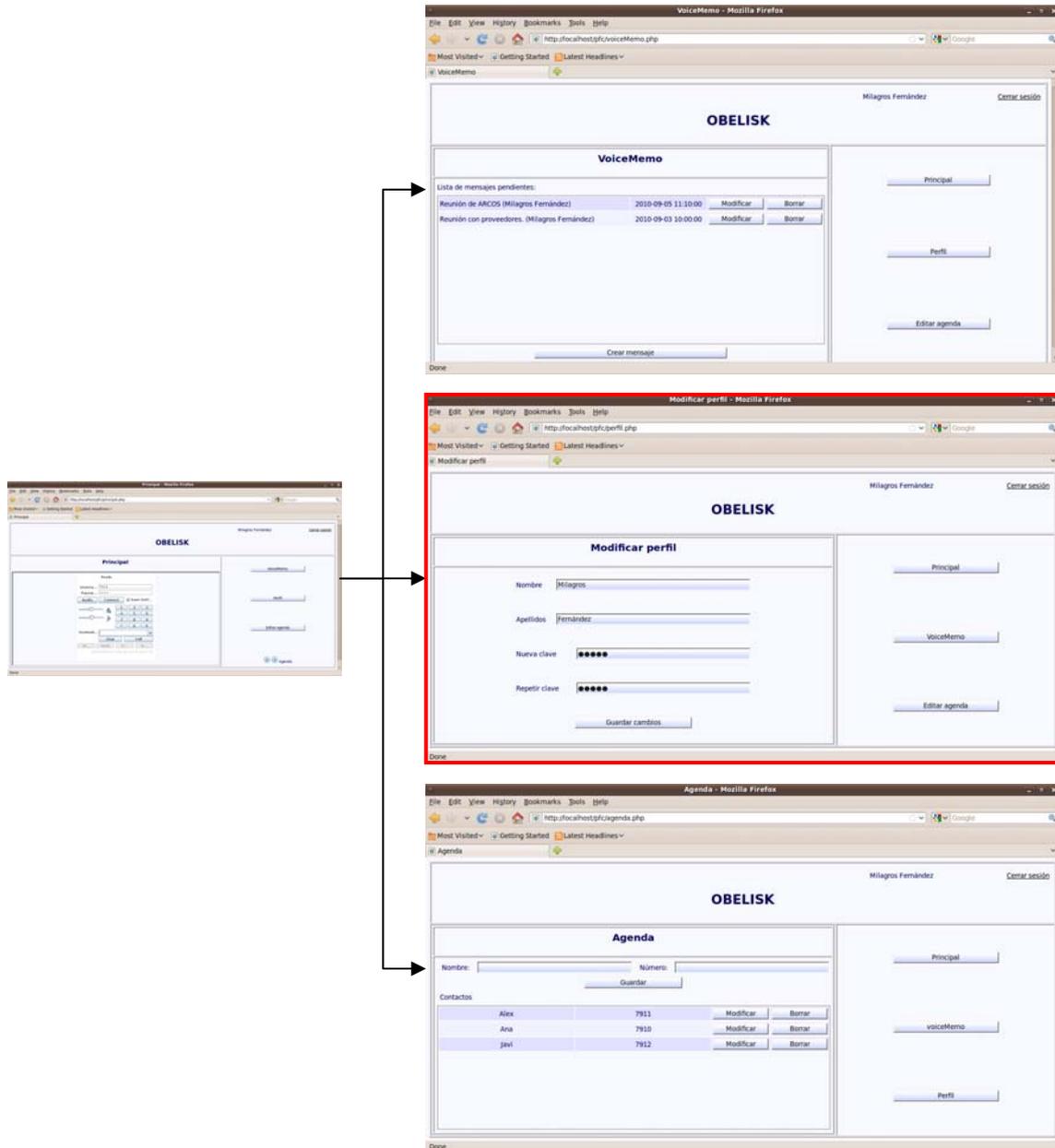


Ilustración 26: Navegación usuario II (Manual usuario)

'Perfil' es una funcionalidad que permite cambiar el nombre, el apellido y la contraseña con la que se accede a la aplicación. En primer momento se cargan los datos con la contraseña, pero sin que se vean los caracteres, para en el caso en el que el usuario no quiera cambiar su contraseña se mantenga la actual (ver Ilustración 27):

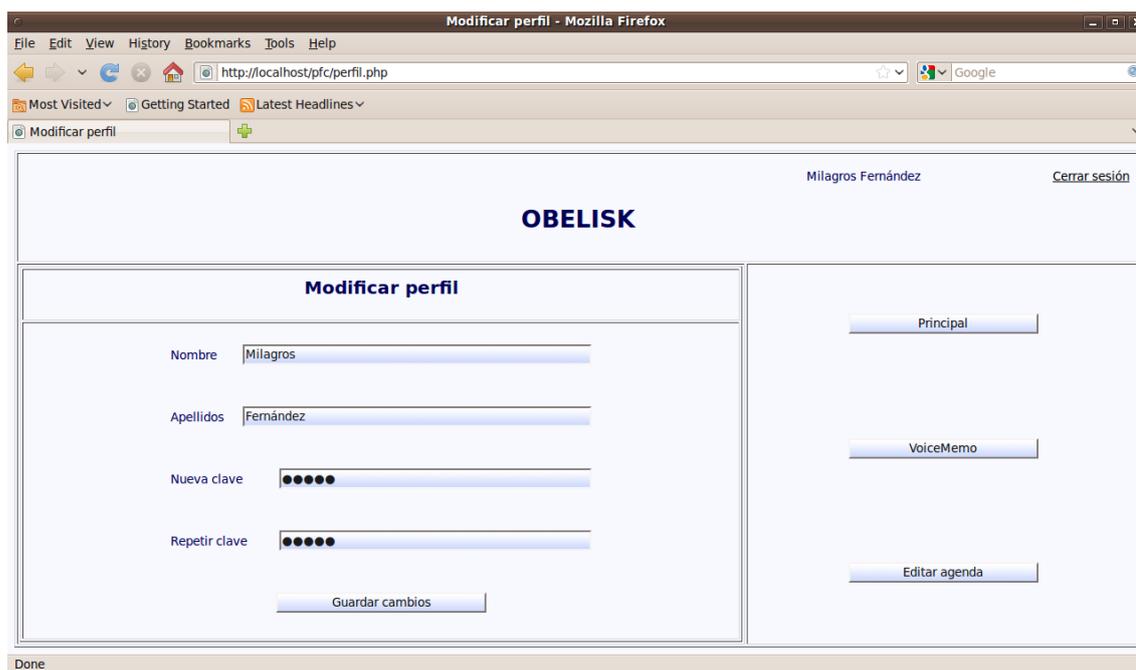


Ilustración 27: Modificar perfil (Manual usuario)

'Editar agenda'

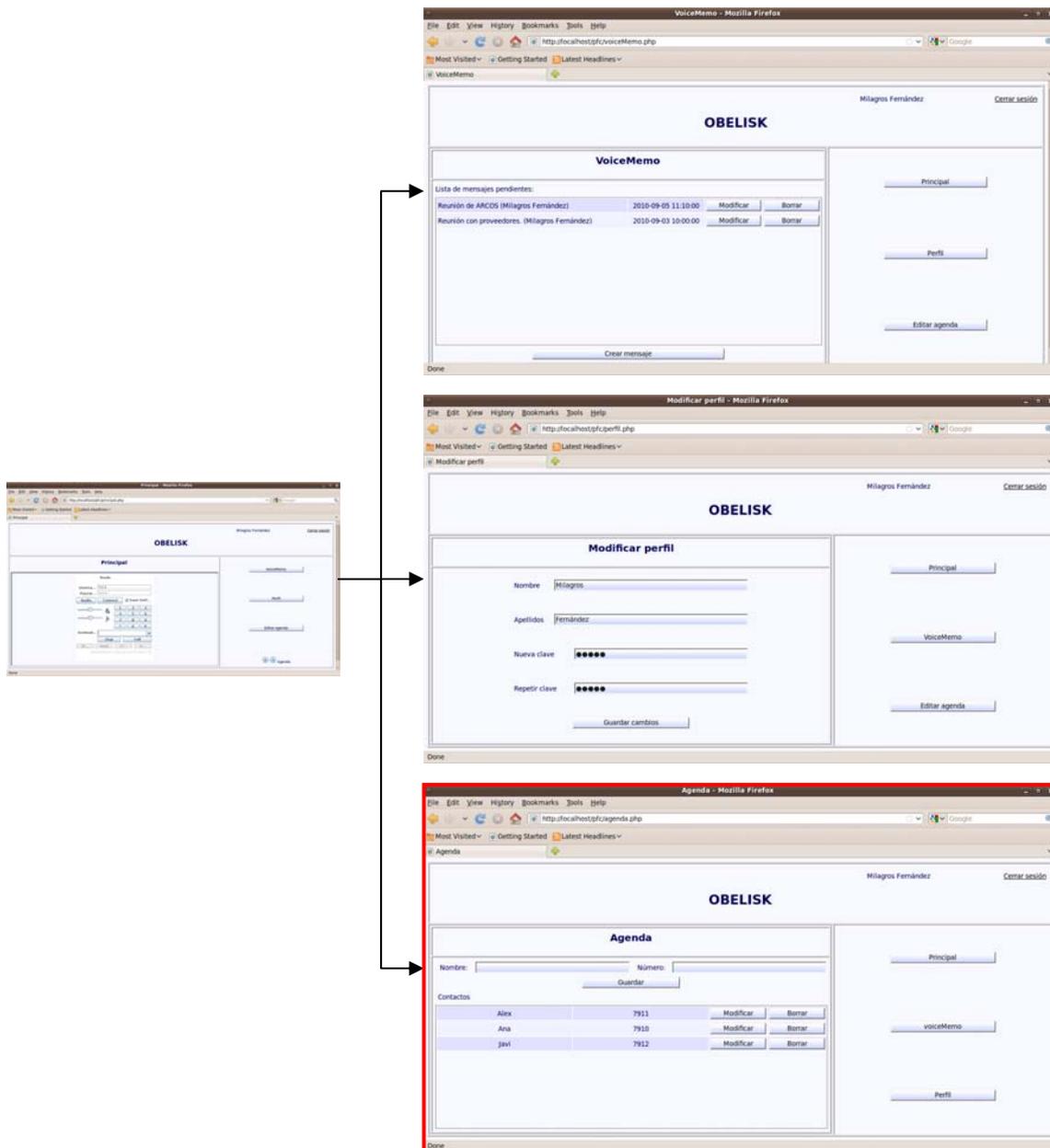


Ilustración 28: Navegación usuario III (Manual usuario)

'Editar agenda' es una funcionalidad que permite administrar los contactos de la agenda. Se pueden gestionar la inserción, modificación y borrado de cada contacto. En la Ilustración 29 se puede ver la pantalla principal de esta gestión.

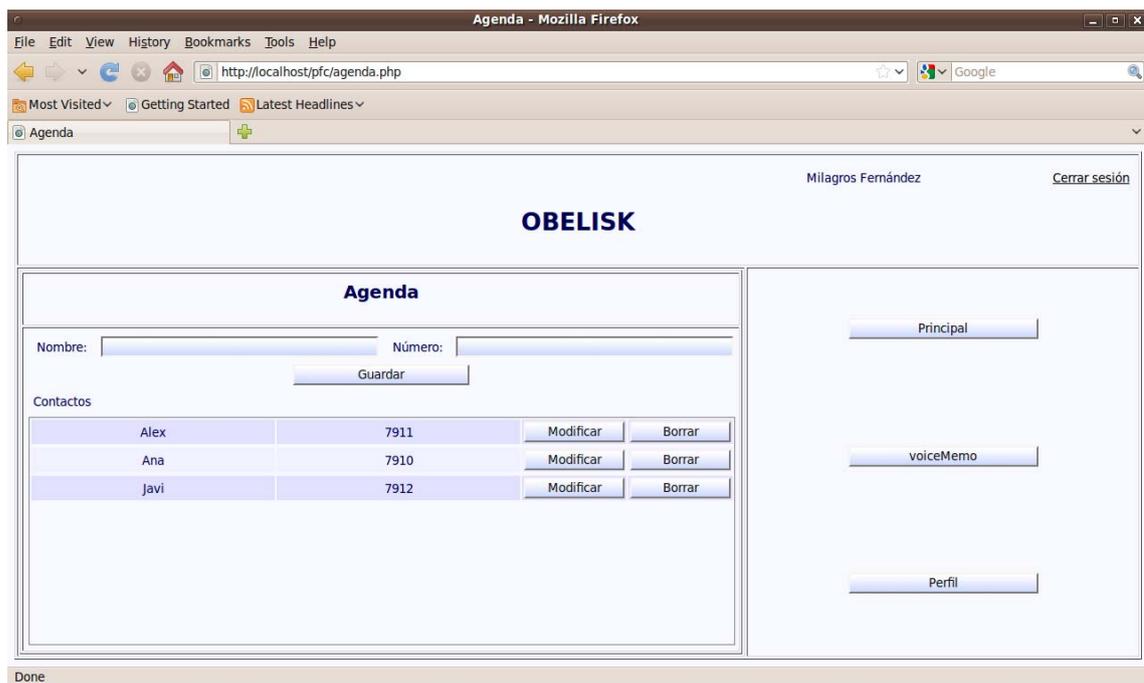


Ilustración 29: Editar agenda (Manual usuario)

En esta pantalla se listan los contactos del usuario por orden alfabético. Al lado de cada contacto se ve la opción de 'Modificar' y 'Borrar'. Para dar de alta un nuevo usuario bastaría con introducir un nombre, un número de teléfono y dar a guardar. Cuando se pulsa modificar se cargan los datos actuales del usuario como se puede ver en la Ilustración 30:

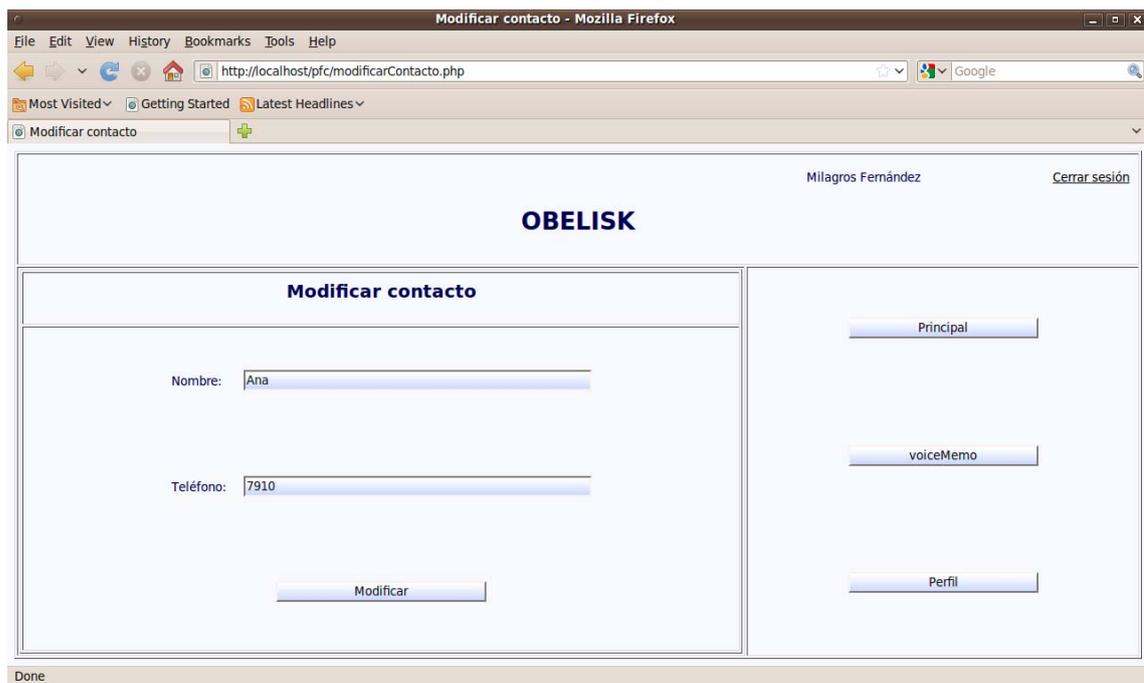


Ilustración 30: Modificar contacto (Manual usuario)

En esa pantalla se modificarían los datos y se pulsaría en modificar y ya quedaría con los datos nuevos introducidos.

8.3. Apéndice C. Manual del administrador

La persona que entre a la aplicación se va a encontrar con la siguiente pantalla (ver Ilustración 31 Ilustración 15):

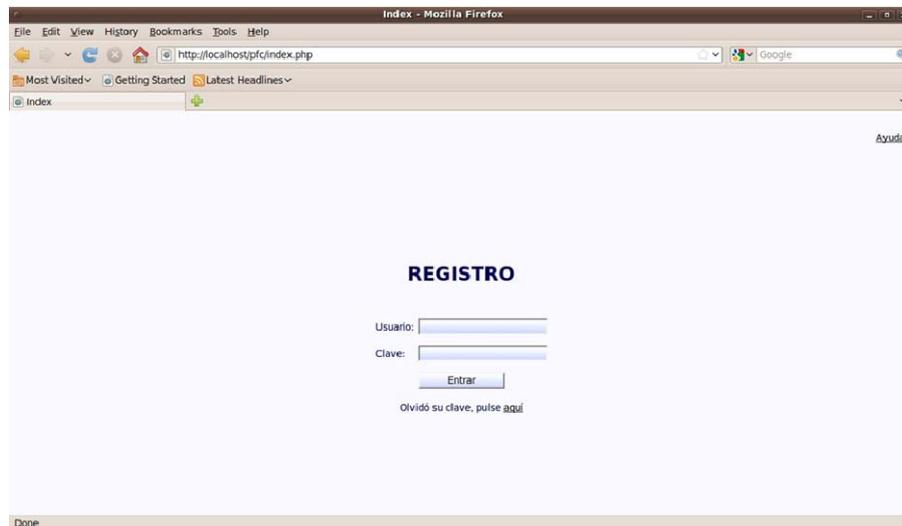


Ilustración 31: Inicio (Manual administrador)

En esta pantalla debe introducir su usuario (número de teléfono) y la clave. Existe una ayuda que detalla los datos que hay que introducir y también existe un acceso para recordar los datos en el caso de que los hayamos olvidado. En la Ilustración 32 aparecen recuadrados.

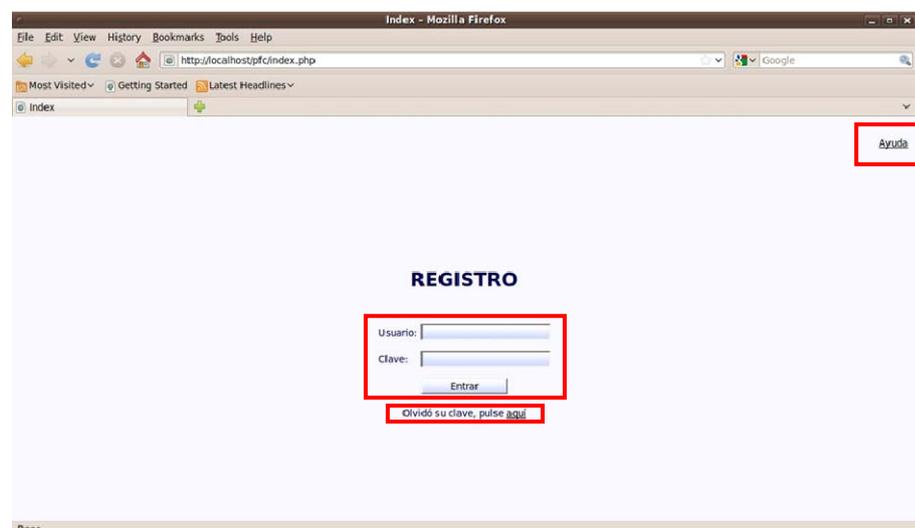


Ilustración 32: Inicio (Manual administrador)

En la Ilustración 33 se puede ver a las pantallas a las que se accede pulsando las diferentes opciones que ofrece la página de inicio. Si se pulsa en la ayuda, se ofrece una breve información general de lo que debe hacerse para entrar en la aplicación; si se introduce un usuario y una clave correctos la aplicación conduce al usuario a la pantalla principal de administración; y si el usuario no recuerda su clave y quiere recordarla tiene un *link* para poder volver a solicitar sus datos de acceso a la aplicación.



Ilustración 33: Inicio (Manual administrador)

En la pantalla de recordar, que se ve en la Ilustración 33, se escribe el correo electrónico con el que el usuario está registrado, en la Ilustración 34 se muestran las diferentes posibilidades a las que se llegan desde este punto.

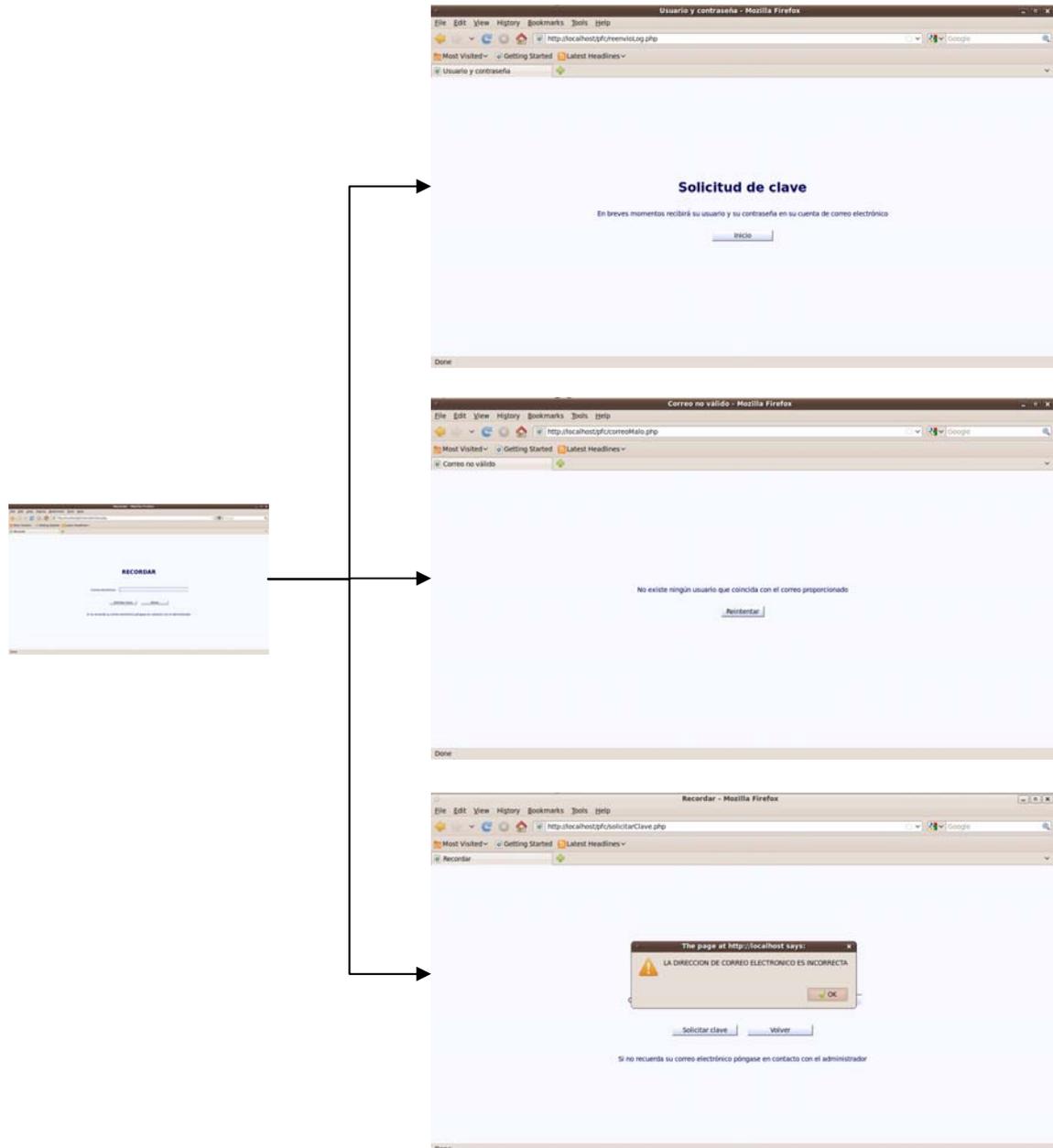


Ilustración 34: Recordar contraseña (Manual administrador)

Basándose en la Ilustración 34 se ve que desde Recordar si el usuario introduce un correo electrónico registrado por la aplicación lanza un mensaje y se envía el usuario y la clave al correo electrónico del usuario; si el correo no está registrado lanza un mensaje informando de ello al usuario; y si el formato de correo no es el adecuado se lanza una alarma.

Una vez que el administrador está registrado la primera pantalla con la que se encuentra es la de principal (ver Ilustración 35).

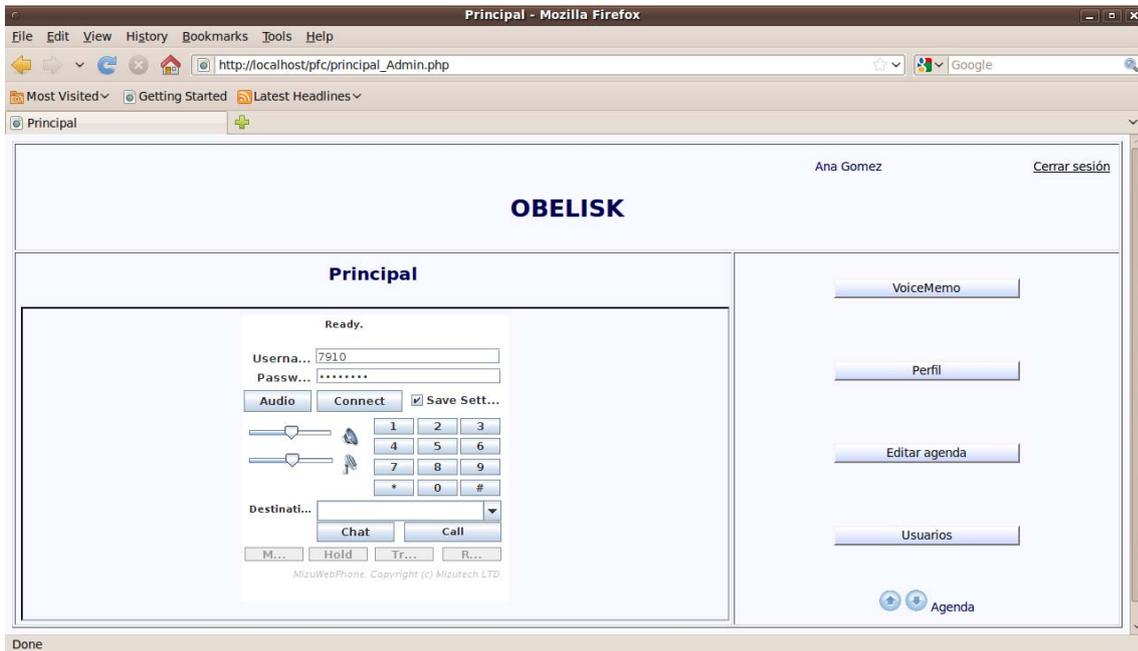


Ilustración 35: Pantalla principal (Manual administrador)

En esta pantalla si se pulsa sobre agenda, se desplegará la agenda y mostrar los contactos de la agenda que tenga el administrador (ver Ilustración 36). Podrá seleccionar cualquiera de ellos y automáticamente se cargará en el teléfono para realizar una llamada.

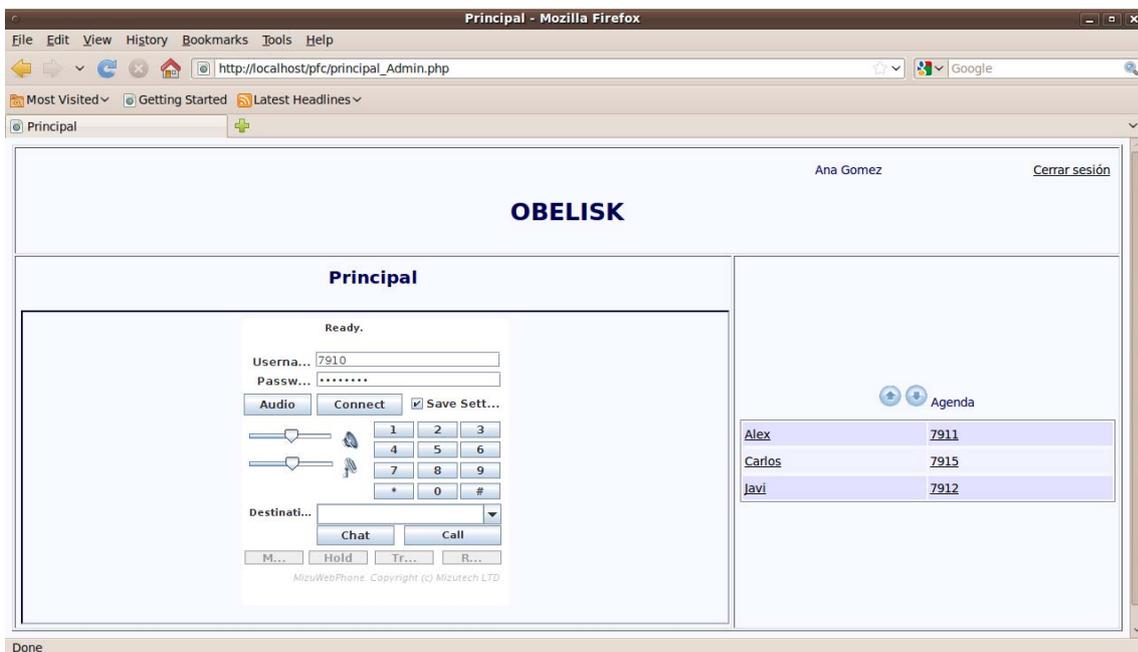


Ilustración 36: Principal con agenda (Manual administrador)

El usuario desde la pantalla principal puede acceder a las diferentes funcionalidades de la aplicación. En la Ilustración 37 se puede ver un diagrama de a dónde se puede ir desde la pantalla principal.



Ilustración 37: Navegación administrador (Manual administrador)

'VoiceMemo'



Ilustración 38: Navegación administrador I (Manual administrador)

VoiceMemo es una funcionalidad utilizada para enviar un mensaje a uno o varios contactos a una hora determinada un día determinado. Lo que hace es llamar a los contactos seleccionados y les lee el mensaje del usuario que lo envía en la fecha que el usuario haya decidido. En primer lugar, cuando se accede a VoiceMemo se puede ver un listado de los mensajes que el usuario haya creado (ver Ilustración 39)

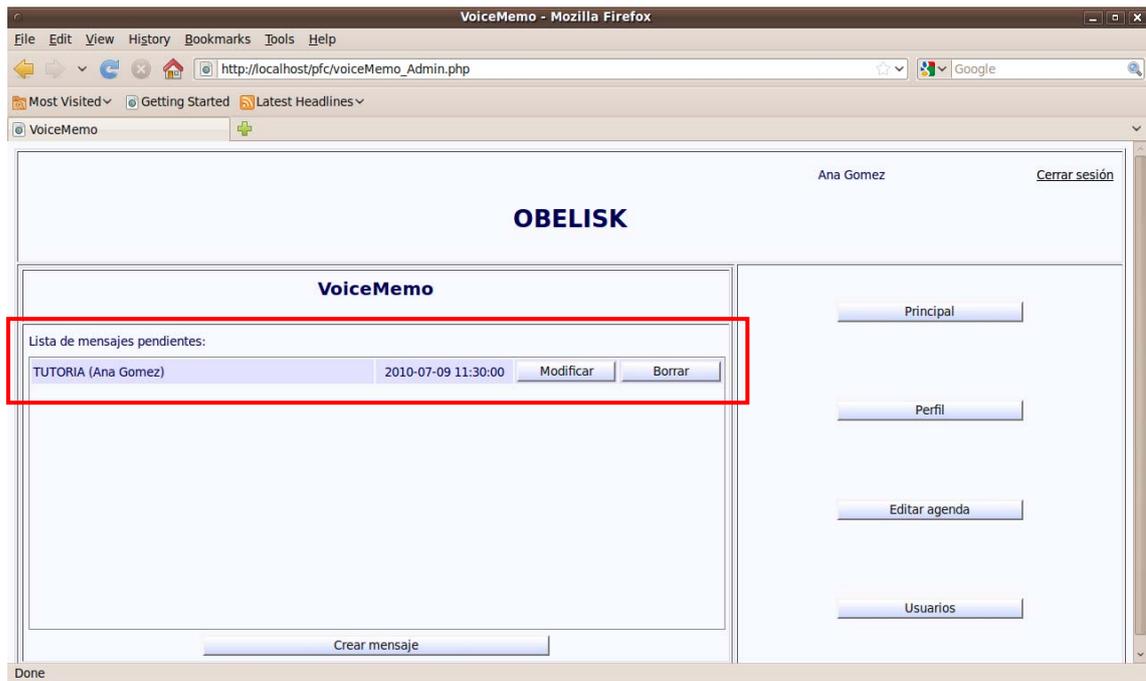


Ilustración 39: VoiceMemo (Manual administrador)

Desde esta pantalla (Ilustración 39) se puede borrar un mensaje, modificarlo o crear uno nuevo. Al igual que ir a cualquier otro sitio de la aplicación. Si se pulsa en 'Crear mensaje' aparecerá una pantalla como la siguiente (ver Ilustración 40):

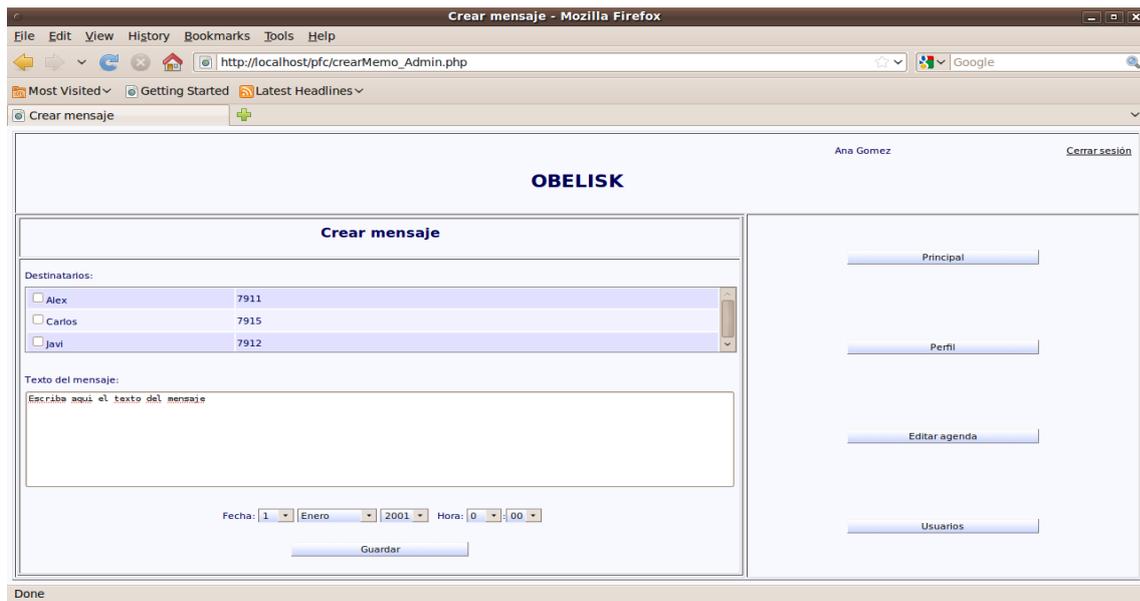


Ilustración 40: Crear mensaje (Manual administrador)

Cuando se pulsa a 'Modificar' se cargan los datos de dicho mensaje en pantalla exceptuando la fecha, como puede verse en la Ilustración 41.

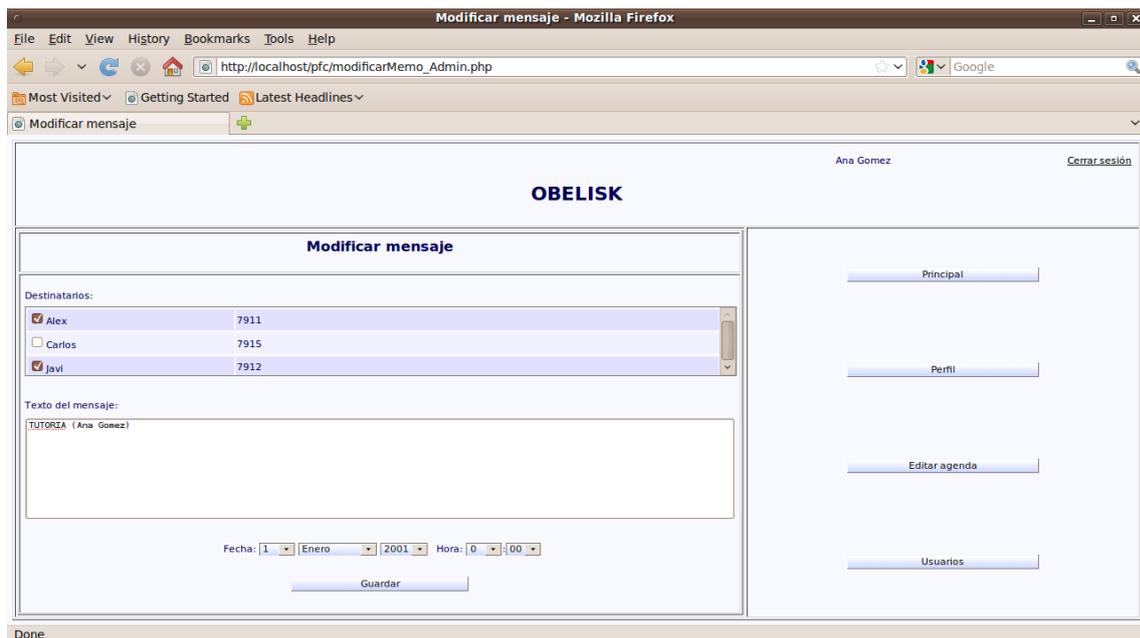


Ilustración 41: Modificar mensaje (Manual administrador)

Si se pulsa a 'Borrar' un mensaje, el mensaje será borrado y se mostrará la lista de mensajes actualizada.

'Perfil'



Ilustración 42: Navegación administrador II (Manual administrador)

'Perfil' es una funcionalidad que permite cambiar el nombre, el apellido y la contraseña con la que se accede a la aplicación. En primer momento se cargan

los datos con la contraseña, pero sin que se vean los caracteres, para en el caso en el que el usuario no quiera cambiar su contraseña se mantenga la actual (ver Ilustración 43):

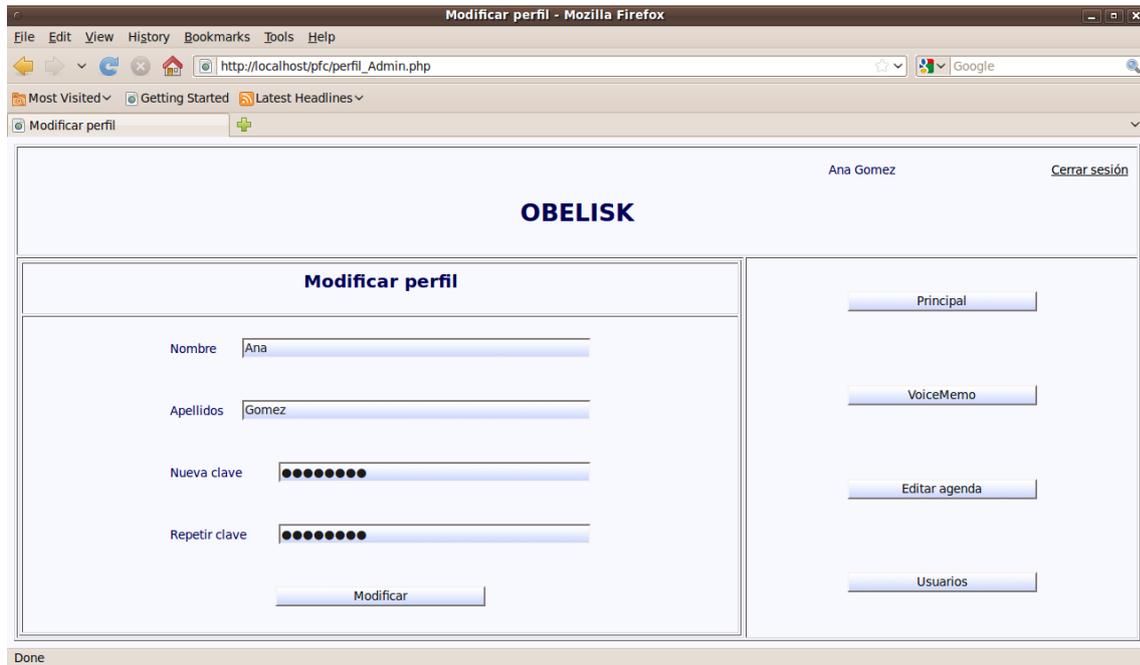


Ilustración 43: Modificar perfil (Manual administrador)

'Editar agenda'



Ilustración 44: Navegación administrador III (Manual administrador)

‘Editar agenda’ es una funcionalidad que permite administrar los contactos de la agenda. Se pueden gestionar la inserción, modificación y borrado de cada contacto. En la Ilustración 45 se puede ver la pantalla principal de esta gestión.

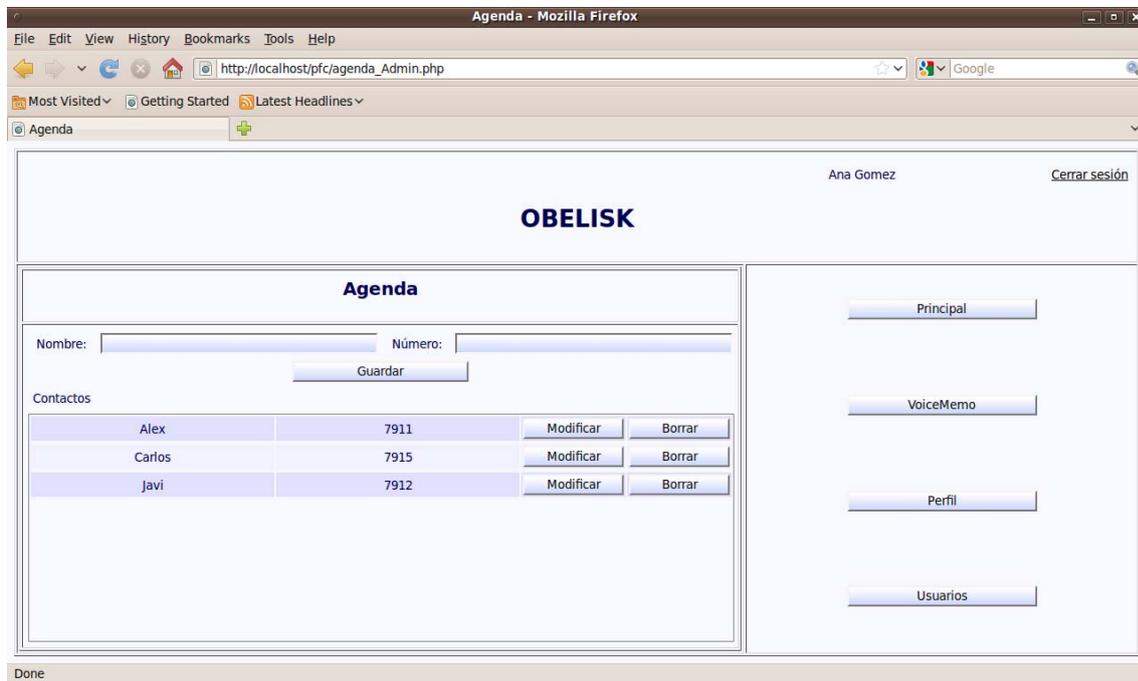


Ilustración 45: Editar agenda (Manual administrador)

En esta pantalla se listan los contactos del usuario por orden alfabético. Al lado de cada contacto se ve la opción de ‘Modificar’ y ‘Borrar’. Para dar de alta un nuevo contacto bastaría con introducir un nombre, un número de teléfono y dar a guardar. Cuando se pulsa modificar se cargan los datos actuales del usuario como se puede ver en la Ilustración 46:

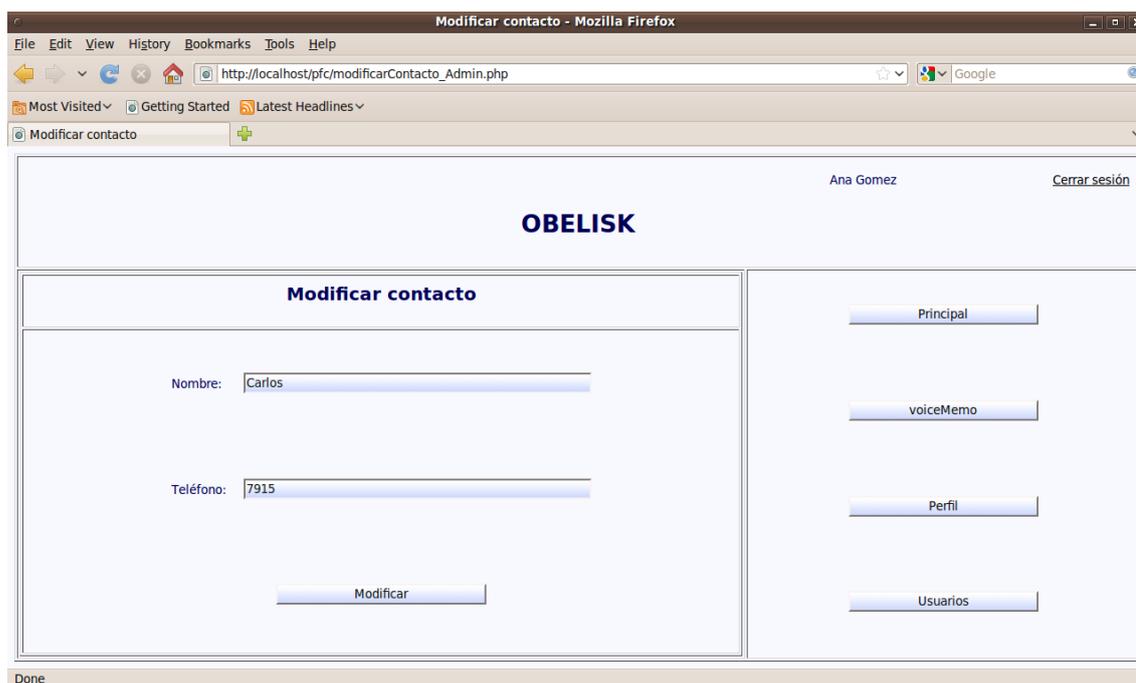


Ilustración 46: Modificar contacto (Manual administrador)

En esa pantalla se modificarían los datos y se pulsaría en modificar y ya quedaría con los datos nuevos introducidos.

'Usuarios'



Ilustración 47: Navegación administrador IV (Manual administrador)

Es una funcionalidad que facilita el alta, la baja y la modificación de usuarios en una centralita de voz, en este caso Asterisk. La aplicación asigna automáticamente el número de teléfono, por lo que no habría problemas de

duplicados de teléfonos. En la Ilustración 48 se puede ver la pantalla principal de la gestión de usuarios.

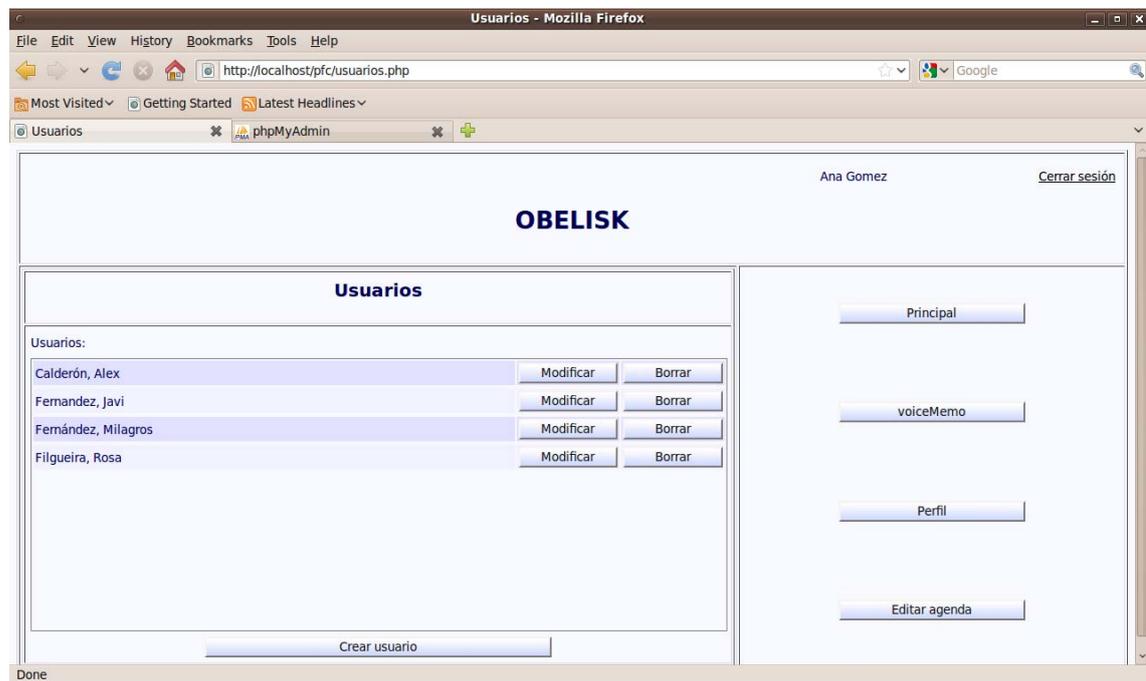


Ilustración 48: Usuarios (Manual administrador)

Desde esta pantalla se puede acceder a la creación de un nuevo usuario. En la Ilustración 49 se puede apreciar lo que se dijo anteriormente de la asignación automática del número de teléfono.

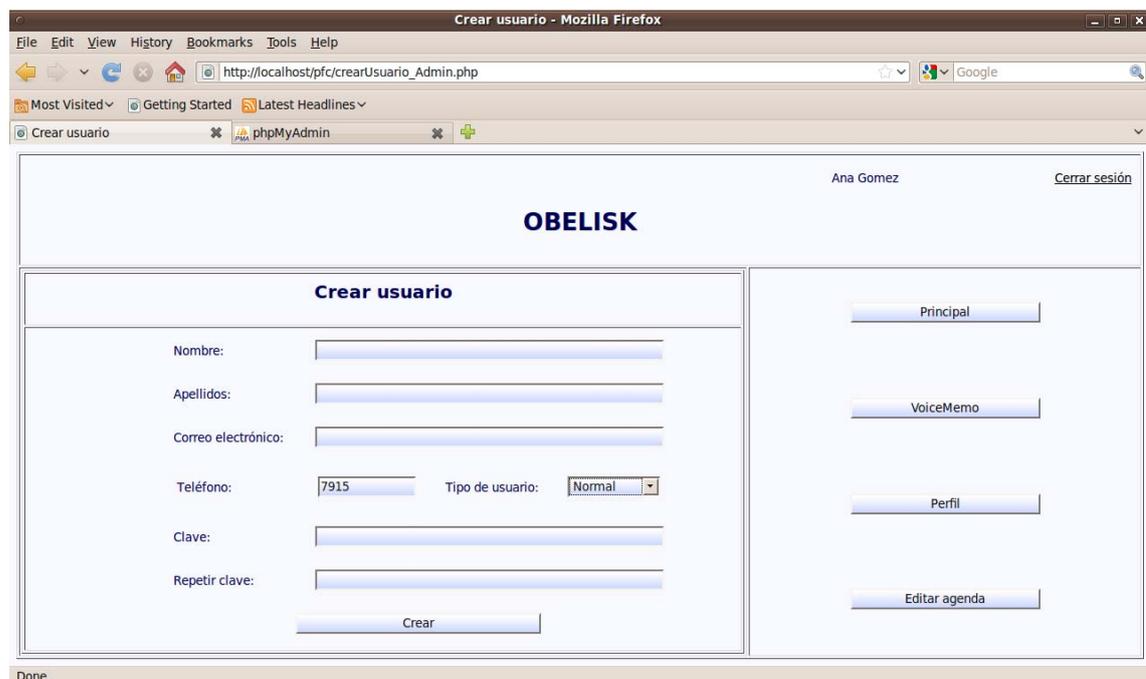


Ilustración 49: Crear usuario (Manual administrador)

En la pantalla de creación se asigna el tipo de usuario que se está creando con un menú desplegable. Todos los datos, excepto el número de teléfono podrán ser modificados en cualquier momento por el administrador utilizando la opción de Modificar usuario que se ve en Ilustración 50 que se muestra a continuación:

The screenshot shows a web browser window titled "Modificar usuario - Mozilla Firefox". The address bar contains "http://localhost/pfc/modificarUsuario_Admin.php". The browser tabs include "Modificar usuario" and "phpMyAdmin". The page content is as follows:

Top right: Ana Gomez [Cerrar sesión](#)

Center: **OBELISK**

Form Title: **Modificar usuario**

Form Fields:

- Nombre:
- Apellidos:
- Correo electrónico:
- Teléfono: Tipo de usuario:
- Clave:
- Repetir clave:

Form Button:

Right Panel Buttons:

-
-
-
-

Bottom status bar: Done

Ilustración 50: Modificar usuario (Manual administrador)