







UNIVERSIDAD DE ZARAGOZA  
Centro Politécnico Superior  
Ingeniería Informática



### Proyecto Fin de Carrera

## DESARROLLO DE UNA APLICACIÓN PARA LOS OPERADORES DE TELEASISTENCIA INTEGRADA EN TELEFONÍA IP

AUTORA: LAURA LACARRA ARCOS

DIRECTOR: ROBERTO CASAS MILLÁN

PONENTE: JESÚS ALASTRUEY BENEDÉ

Departamento Informática e Ingeniería de Sistemas

Zaragoza, Julio 2010



*A mi familia, amigos y  
compañeros de trabajo*



## **RESUMEN**

La teleasistencia domiciliaria es un servicio de ayuda a personas mayores y/o discapacitadas. Las personas o clientes que contratan el servicio disponen de una instalación específica compuesta por un terminal (similar a un teléfono con un botón de emergencia) y varios dispositivos (sensores) que permiten ponerse en contacto con un centro de asistencia, por voluntad de la persona o activación propia de los dispositivos.

El centro de asistencia está formado por operadores, cuya misión es atender las alarmas que generen las personas y proporcionar una ayuda eficaz y personalizada ya sea movilizando familiares, conocidos u organismos de la comunidad.

Actualmente cada centro de teleasistencia está configurado a través de la red de teléfono convencional. Las llamadas llegan a una centralita de teleasistencia cuya funcionalidad es transformar los datos en digitales, gestionar los teléfonos de los operadores y traducir los protocolos de los equipos que están instalados en cada hogar. Una instalación con estas características es poco escalable y costosa.

El proyecto consiste en el desarrollo de un sistema para una empresa de teleasistencia en el cual los operadores puedan gestionar las alarmas recibidas en el centro y la información de las personas que contratan el servicio. El sistema estará integrado en una plataforma IP en el cual la gestión telefónica pasará de ser analógica a ser por VozIP y la gestión de los distintos protocolos, pasará de ser por hardware a software.

Los objetivos que se persiguen son dos: facilitar la tarea de los operadores en la atención de alarmas y ahorrar grandes costes a las empresas reduciendo el gasto de la infraestructura inicial y de la instalación de equipamiento de la empresa. Gracias a la VozIP, es posible una centralita de teleasistencia configurada en base a las necesidades de los clientes y operadores, de forma que sea susceptible a cambios de crecimiento sin ocasionar los costes que supondría un aumento de hardware, y ante todo, la asistencia será rápida y exitosa en beneficio de las personas.



# TABLA DE CONTENIDOS

1.	INTRODUCCIÓN .....	13
1.1	Introducción .....	13
1.2	Contexto .....	13
1.3	Motivación .....	13
1.4	Definición y alcance del sistema .....	14
1.5	Definición y alcance de la integración .....	14
1.6	Fases del trabajo .....	15
1.7	Herramientas utilizadas .....	15
1.8	Contenido de la documentación .....	16
2.	PLANIFICACION DEL SISTEMA .....	17
2.1	Metodología .....	17
2.2	Planificación.....	17
3.	ESTUDIO DEL ENTORNO.....	19
3.1	Contexto de Teleasistencia.....	19
3.2	Investigación de la arquitectura actual .....	20
3.3	Contexto de la voz IP .....	21
3.4	¿Por qué voz IP?.....	21
3.5	Arquitectura IP de AsisT .....	22
4.	ANÁLISIS DE LA INFORMACIÓN .....	24
4.1	Actividades realizadas en el análisis .....	24
4.2	Resumen de los requisitos de AsisT.....	24
4.3	Definición de actores y roles .....	25
4.4	Los subsistemas .....	25
5.	DISEÑO DE LA INFORMACIÓN .....	27
5.1	Resumen de la actividad del diseño .....	27
5.2	Patrones de diseño .....	27
5.2.1	Interfaz - gestión .....	27
5.3	Elección del tipo de aplicación.....	28
5.4	Elección del lenguaje de programación de AsisT .....	29
5.5	Elección del lenguaje de programación que integra la centralita IP con AsisT .....	29
6.	SISTEMA ASIST .....	31
6.1	Subsistema de bases de datos .....	31

6.1.1	Elección gestor de bases de datos .....	31
6.1.2	Implantación base de datos.....	31
6.1.3	Integración centralita IP con base de datos .....	31
6.2	Subsistema de atención de alarmas .....	32
6.2.1	Solución al aviso de alarmas .....	32
6.2.2	Elección del servidor de mensajería .....	33
6.2.3	Integración mensajería con AsisT .....	33
6.3	Subsistema teléfono.....	37
6.3.1	Centralita IP .....	37
6.3.2	Solución adoptada.....	38
6.3.3	Conexión con la centralita .....	38
6.3.4	Establecer una llamada .....	39
6.3.5	Aceptar una llamada entrante .....	39
6.3.6	Interfaz del teléfono.....	40
6.3.7	Integración con la base de datos .....	40
6.3.8	Chequeo de un equipo .....	40
6.4	Subsistema información .....	41
6.4.1	Contexto del funcionamiento.....	41
6.4.2	Cuadro de mandos .....	42
6.4.3	Gestión Alarmas .....	42
6.4.4	Gestión de la autentificación .....	44
6.4.5	Gestión de los usuarios de la aplicación .....	44
6.4.6	Gestión de los clientes de la aplicación .....	44
6.4.7	Buscar expedientes .....	44
6.4.8	Gestión de un expediente de un cliente .....	45
6.4.9	Gestión Historiales .....	46
6.5	Funcionalidades adicionales de la aplicación.....	47
6.5.1	Complementos .....	47
6.5.2	Traducciones.....	47
6.5.3	Soporte de distintos gestores de bases de datos.....	47
6.5.4	Permisos de usuarios .....	47
6.5.5	Control de errores .....	48
7.	CONCLUSIONES .....	49
7.1	Cumplimiento de los objetivos iniciales .....	49

7.2	Futuro de AsisT .....	49
7.3	Valoración Personal .....	50
	ACRÓNIMOS .....	55
	BIBLIOGRAFÍA .....	56
	GLOSARIO .....	58

## **Lista de figuras**

Figura 2-1 Planificación .....	18
Figura 3-1 Instalación de teleasistencia.....	19
Figura 3-2 Arquitectura actual.....	20
Figura 3-3 Arquitectura IP de AsisT .....	22
Figura 4-1 Subsistemas.....	26
Figura 5-1 Interfaz - gestión .....	28
Figura 6-1 Mensajería modelo publicador / suscriptor.....	33
Figura 6-2 Aviso de alarma .....	34
Figura 6-3 Ventana aviso de alarma.....	34
Figura 6-4 Caputura de alarma .....	35
Figura 6-5 Error en captura .....	36
Figura 6-6 Alarma capturada.....	37
Figura 6-7 Plataforma IP .....	38
Figura 6-8 Conexión con la centralita .....	39
Figura 6-9 Realizar una llamada.....	39
Figura 6-10 <i>Softphone</i> sonando - hablando – en espera .....	40
Figura 6-11 Inicio aplicación AsisT .....	41
Figura 6-12 Ficha operador .....	42
Figura 6-13 Gestión de alarmas.....	43
Figura 6-14 Expediente cliente.....	45

## **Lista de Tablas**

Tabla B-1 Estado: situación-motivo.....	<b>¡Error! Marcador no definido.</b>
---	--------------------------------------



# 1. INTRODUCCIÓN

---

## 1.1 Introducción

El objetivo de este proyecto es desarrollar un sistema de atención de alarmas, llamado AsisT, para los usuarios de teleasistencia de forma que todos los usuarios u operadores que están en sus puestos de trabajo, reciban una alarma, la atiendan y quede registrada.

La gran novedad es que el sistema esté integrado en una plataforma IP y que el teléfono analógico, pase a estar contenido a la aplicación. La plataforma IP debe estar configurada a medida para la teleasistencia, de forma que reciba las alarmas que activan los clientes que contratan el servicio y gestione las llamadas entrantes y salientes.

La gran ventaja de la voz IP, es el ahorro en costes que supone para la empresa que aporta el servicio: en primer lugar, por la instalación y mantenimiento y en segundo lugar, por ser una solución escalable, de forma que reduce el coste en caso de producirse un incremento en el número de operadores y/o número de clientes.

En esta memoria se explica e ilustra todas las fases que se han llevado en el desarrollo para que al final, el sistema AsisT, sea una realidad.

---

## 1.2 Contexto

El proyecto se ha realizado en la empresa Diaple Networking S.L. Uno de los servicios informáticos que ofrece la empresa Diaple es el de telefonía VozIP.

La empresa disponía la tecnología y los conocimientos por adoptar una solución diferente a la teleasistencia basándose en voz IP. La idea era construir la plataforma con dicha tecnología de forma que:

- La instalación y el equipamiento de la infraestructura por Voz IP abarate los costes en la empresa.
- La plataforma IP solucione la problemática de la poca escalabilidad del hardware.

AsisT, el software de gestión de alarmas, es un componente de la plataforma de teleasistencia. Los datos para la gestión de los clientes beneficiarios del servicio, está basado en el convenio marco [5] establecida por IMSERSO [3] y FEMP [4]. Este documento explica y detalla la información que debe contener la aplicación.

---

## 1.3 Motivación

Para empezar, existe un interés social, ya que la teleasistencia domiciliaria es un servicio que concierne a la vida de las personas. Aporta mejoras en la calidad de vida, ya que los beneficiarios del servicio reciben asistencia las 24 horas del día y todos los días del año o meramente, contribuye a su tranquilidad al saber que con sólo pulsar un botón, puede recibir ayuda, recordatorios de asuntos personales o conversación. Todas las mejoras que puedan producirse, beneficiará a este colectivo, y muy probablemente, a las personas de nuestro entorno.

Por otro lado, se propone un sistema totalmente nuevo en el mercado, ya que actualmente no hay ninguno con estas características. Las ventajas que aporta la telefonía IP serán

trasladadas a gran escala a una empresa de teleasistencia, lo que supone sin entrar en detalle, un ahorro económico importante.

Por último, las perspectivas de futuro previstas. Las empresas de teleasistencia están buscando soluciones al alto coste de la instalación y mantenimiento de la plataforma actual. Un nuevo sistema con estas características prometería un gran futuro en mejorar el software, incorporando más información que beneficie a los clientes e integrando nuevas tecnologías existentes del mercado.

---

## 1.4 Definición y alcance del sistema

El objetivo del proyecto es el desarrollo de la aplicación AsisT, destinado a una empresa de teleasistencia donde prestan el servicio los operadores. El software por tanto, ofrece varias vertientes de gestión:

- Los usuarios u operadores que hagan uso de la aplicación.
- Los clientes que contratan el servicio de teleasistencia.
- La asistencia de alarmas.

Los objetivos que se persiguen en este proyecto son analizar, desarrollar e implementar:

- Un sistema basándose en el convenio-marco de IMSERSO y FEMP, de forma que los datos de los clientes y la gestión de alarmas cumplan las especificaciones [5].
- La gestión de los usuarios u operadores de la aplicación y los clientes.
- Los avisos en tiempo real de las alarmas, de forma que los operadores puedan capturarlas en exclusividad.
- Un teléfono integrado en la aplicación, capaz de aceptar o realizar una llamada a través de una centralita IP.
- Un sistema robusto, escalable, multiplataforma y que cumpla la ley de protección de datos.

---

## 1.5 Definición y alcance de la integración

El sistema AsisT está integrado en una plataforma IP. La plataforma IP está desarrollada por el equipo de sistemas de Diaple Networking. En este apartado se pretende definir en qué consiste la integración y qué escapa de los objetivos del proyecto.

En la plataforma IP, se dispone de la infraestructura necesaria:

- La interconexión entre la red de telefonía básica, RTB y la centralita IP.
- El despliegue de software necesario: servidores, líneas de conexión de datos, sistemas de almacenamiento, etc.

La centralita está configurada a medida y tiene la lógica necesaria para aceptar y realizar llamadas, así como gestionarlas en la red local del centro que esté instalado. Además, en lo referente a la teleasistencia, es capaz de recibir las alarmas de distintos protocolos de terminales, que tienen los clientes instalados en cada hogar.

Toda la parte de instalación y gestión de la plataforma, queda ajena a los objetivos del proyecto. No obstante, dado que AsisT está integrado en la misma, ha sido necesaria analizarla, estudiarla y comprenderla para el desarrollo del proyecto.

El sistema AsisT, parte desde el punto que ha llegado una alarma al centro de teleasistencia. Queda dentro de los objetivos del proyecto abordar la solución de cómo los operadores son notificados de una alarma, así como, el desarrollo del teléfono en la aplicación, también llamado *softphone*, para que se comunique con la centralita.

---

## 1.6 Fases del trabajo

En un principio, se recopiló toda la información aportada en el convenio-marco de IMSERSO [3] y FEMP [4], en el boletín que se emitió en el 2007. En el documento [5] está el contexto de la teleasistencia en lo relativo a definiciones del servicio y especificaciones técnicas tanto de hardware y software.

Partiendo de las conclusiones obtenidas del convenio, se estudió el entorno: la tecnología actual y la arquitectura implementada por el equipo de sistemas de Diaple. El siguiente paso fue escoger la metodología Métrica 3 [30] y empezar las fases de desarrollo del sistema. Se realizó un completo análisis y diseño del sistema AsisT, para posteriormente ponerlo en práctica en la implementación.

Durante el desarrollo, era imprescindible comprender y cooperar con el equipo de trabajo encargado de desarrollar la plataforma IP. Eventualmente, se realizaron reuniones para la integración y puesta a punto de la plataforma.

---

## 1.7 Herramientas utilizadas

El sistema ha sido elaborado con software libre. El software libre ofrece multitud de ventajas. Los motivos que han llevado a su elección:

- Económico: el ahorro de licencias tanto en el desarrollo como en la implementación repercute en los costes totales del producto.
- Corrección más rápida y segura de fallos: el software está soportado por una comunidad cuyos fallos se revisan de forma que asegure la calidad del software.
- Formatos estándar: evitar incompatibilidades de formatos.

Las herramientas de desarrollo utilizadas han sido:

- NetBeans 6,8 [13]: IDE de programación en Java [13].
- Aptana Studio: 2.0.4 [14]: IDE de programación en PHP [32].
- MySQL Query Browser [44]: herramienta de gestión de bases de datos en MySQL [9].
- Subversion: sistema de control de versiones integrado en NetBeans. Es un repositorio central de código muy útil para trabajar en grupo y en diferentes puestos de trabajo [37].
- Navegadores: Mozilla Firefox [42].

Las herramientas auxiliares empleadas han sido:

- Herramientas de prototipos: Pencil v1.0 [31].
- Herramientas de modelado de diagramas: Dia v.0.97, Umbrello UML Modeler [35].
- Editor de imágenes: GIMP [34].
- Herramienta de procesamiento de textos y diagramas: OppenOffice.org [36].

## 1.8 Contenido de la documentación

Este apartado pretende encaminar la lectura de la documentación aportada.

La siguiente sección, presenta la metodología empleada, así como los estándares que se han utilizado.

Posteriormente, se presenta un estudio del contexto de la teleasistencia y la arquitectura actual, detectando las ventajas y desventajas que tiene, frente a una plataforma IP. Se muestra además, la plataforma desarrollada para AsisT en la sección 3.5 y se presentan posibles soluciones en el ANEXO A.

El desarrollo del sistema prosigue explicando las conclusiones destacadas en el análisis y desarrollo del sistema AsisT, secciones 4 y 5 de este documento. En el ANEXO B y el ANEXO C respectivamente, se detallan en profundidad.

En la sección 6 se describe el sistema completo, describiendo las soluciones y elecciones tomadas y destacando la funcionalidad implementada. Por otro lado, en el ANEXO D se detallan aspectos técnicos implementados y en el ANEXO E se presenta la completa aplicación con el “Manual de usuario”.

Finalmente, en la sección 7 se presentan las conclusiones alcanzadas una vez realizado el proyecto.

## 2. PLANIFICACION DEL SISTEMA

---

### 2.1 Metodología

En la actualidad existen varios estándares de metodología de desarrollo el software: La metodología utilizada para el desarrollo del sistema ha sido Métrica 3 [30]. Los motivos han sido:

- Conocimiento previo de Métrica 2.1.
- Familiarización y recomendación de la empresa con dicha metodología.
- Es completa y detallada.
- Es propiedad intelectual del ministerio de administraciones públicas de España.
- Disponible de la documentación en Internet, en la página oficial [30].

Métrica 3 se divide en grandes bloques y cada bloque tiene distintas actividades que se subdividen en tareas. La forma de abordar el problema ha sido seleccionar los bloques beneficiosos, realizar las actividades más significativas y relevantes de forma que cada actividad aportara nueva información a los siguientes pasos y completar de esta forma y progresivamente el desarrollo del producto.

Los bloques que se han seguido más exhaustivamente son:

- Planificación del sistema de información:
  - o Tareas a realizar, el estudio inicial del contexto, la situación actual, los sistemas de información, definir el plan de trabajo, etc.
- Análisis del sistema de información:
  - o Actividades para el análisis del sistema. Disponible en el ANEXO B.
- Diseño del sistema de información:
  - o Actividades para el diseño del sistema. Disponible en el ANEXO C.
- Construcción del sistema de información:
  - o Implementación y pruebas del sistema. Disponible en el ANEXO D.
  - o Manual de usuario. Disponible en el ANEXO E.

Métrica ha servido de guía para realizar el desarrollo del producto. Esto significa que, en ocasiones, se ha seguido otro orden al efectuar las actividades que se han considerado prioritarias. También se han omitido algunas que resultaban redundantes o se han fusionado varias, para mejorar la legibilidad.

El ciclo de vida del desarrollo ha sido en cascada mejorado, es decir, tras cada etapa se han revisado las anteriores actividades.

El desarrollo de la implementación se ha realizado por módulos. Una vez implementado un módulo, se han realizado las pruebas del mismo, se han integrado con el resto de módulos y se han realizado las pruebas de integración, siguiendo la línea de cascada por módulos.

---

### 2.2 Planificación

En el inicio del proyecto, una vez evaluada la metodología a utilizar y las tareas a realizar, se hizo una planificación inicial, siguiendo los pasos de la “Actividad de Planificación” de Métrica 3.

La estimación era de 7 meses distribuidos desde Octubre a Abril. La planificación sufrió riesgos, detalles que se explican en las conclusiones de los ANEXOS B, D y C.

## Planificación del Sistema

---

La última planificación desarrollada y realizada es la que se muestra en la Figura 2-1.

		Nombre	Duración	Inicio	Terminado	Predecesores
1		■ <b>Análisis</b>	<b>159 days?</b>	<b>1/10/09 8:00</b>	<b>11/05/10 17:00</b>	
2		Lectura y familiarización con el sistema	6,5 days	1/10/09 8:00	9/10/09 13:00	
3		Estudio de la arquitectura	4 days?	19/10/09 8:00	22/10/09 17:00	2
4		Estudio del entorno tecnológico	1 day	23/10/09 8:00	23/10/09 17:00	3
5		Estudio herramientas UML	2 days	26/10/09 8:00	27/10/09 17:00	4
6		Division en Subsistemas	1 day	28/10/09 8:00	28/10/09 17:00	5
7		Division en módulos	2 days	29/10/09 8:00	30/10/09 17:00	6
8		Ánalisis E/R	5 days	2/11/09 8:00	6/11/09 17:00	7
9		Análisis Clases	3 days	9/11/09 8:00	11/11/09 17:00	8
10		Prototipo Interfaz	7 days	12/11/09 8:00	20/11/09 17:00	9
11		Revisión Análisis	3 days	23/11/09 8:00	25/11/09 17:00	10
12		■ <b>Diseño</b>	<b>119 days?</b>	<b>26/11/09 8:00</b>	<b>11/05/10 17:00</b>	<b>11</b>
13		Estudio Patrones de diseño	1 day	26/11/09 8:00	26/11/09 17:00	
14		Patrones de diseño	2 days	27/11/09 8:00	30/11/09 17:00	13
15		Escenarios	2 days?	1/12/09 8:00	2/12/09 17:00	14
16		Casos de Uso	7 days	3/12/09 8:00	11/12/09 17:00	15
17		Diagrama de Actividad	2 days	14/12/09 8:00	15/12/09 17:00	16
18		Diagrama de Clases	2 days	16/12/09 8:00	17/12/09 17:00	17
19		Revisión del Diseño	2 days	18/12/09 8:00	21/12/09 17:00	18
20		■ <b>Implementación</b>	<b>101 days?</b>	<b>22/12/09 8:00</b>	<b>11/05/10 17:00</b>	<b>19</b>
21		Estudio Gestor BD	1 day?	22/12/09 8:00	22/12/09 17:00	
22		Implementación BD	7 days	23/12/09 8:00	31/12/09 17:00	21
23		Preparación librerías y entorno	5 days	22/12/09 8:00	28/12/09 17:00	
24		Implementación cargador de módulos	4 days	29/12/09 8:00	1/01/10 17:00	23
25		Módulo autenticación	4 days	4/01/10 8:00	7/01/10 17:00	24
26		Módulo Gestión Usuarios	5 days	8/01/10 8:00	14/01/10 17:00	25
27		Módulo Gestión Cliente	5 days	20/01/10 8:00	26/01/10 17:00	26
28		Módulo Expediente	30 days	27/01/10 8:00	9/03/10 17:00	27
29		Módulo Alarmas	10 days	10/03/10 8:00	23/03/10 17:00	28
30		Subsistema aviso alarmas	15 days	24/03/10 8:00	13/04/10 17:00	29
31		Subsistema teléfono	20 days	14/04/10 8:00	11/05/10 17:00	30

**Figura 2-1 Planificación**

## 3. ESTUDIO DEL ENTORNO

### 3.1 Contexto de Teleasistencia

Una empresa de teleasistencia dispone de uno o varios centros en los que ofrece servicio. Entre otros trabajadores, el centro cuenta con un equipo de operadores y un equipo preparado para desplazarse al domicilio del cliente según la incidencia: técnico, emergencia, etc.

Una entidad local (un ayuntamiento, una comarca, un barrio, etc) puede solicitar contratar el servicio de teleasistencia en su jurisdicción. La solicitud pasa a concurso y se le asigna una empresa privada para la prestación del servicio a la entidad.

El centro de teleasistencia puede subdividirse en grupos de operadores para que gestionen un área, distrito o entidad. De esta forma las alarmas, según el origen sean atendidas por el grupo específico.

Un cliente solicita a la entidad local o a una empresa, el servicio de teleasistencia. Si cumple los requisitos, se instalará en su hogar un terminal y una unidad de control remoto, UCR. El cliente puede disponer además de varios dispositivos o equipos tal y como aparece en la Figura 3-1 Instalación de teleasistencia. La instalación pertenece a un hogar y tiene un código de identificación de usuario (CIU). Si el cliente se muda de vivienda, la instalación también.

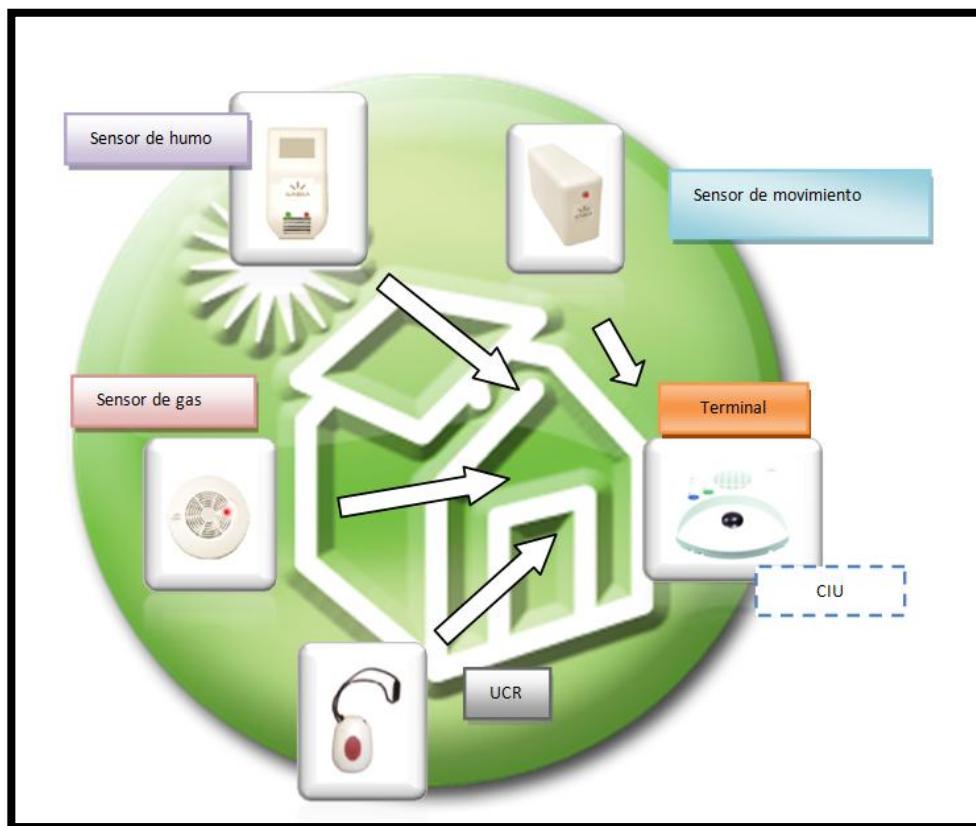


Figura 3-1 Instalación de teleasistencia

Una alarma es originada por un equipo. Si el equipo es el terminal o la UCR, se inicia una alarma por la pulsación del botón, y se produce una llamada al centro de teleasistencia. Por el contrario, si la alarma es producida por otro dispositivo, no se inicia llamada. En este caso es una incidencia técnica que también tendrá que asistir el centro.

Todos los dispositivos están conectados al terminal que sabe el equipo que la ha originado y manda la señal al centro de teleasistencia.

Para más detalle, véase las definiciones de teleasistencia en el GLOSARIO , página 58.

### 3.2 Investigación de la arquitectura actual

Actualmente el sistema de teleasistencia está basado en la red de telefónica básica, RTB. Esto significa que cada vez que una instalación que posee un cliente origina una alarma, la alarma es una llamada por la red de teléfono convencional.

En la siguiente Figura 3-2, se observa la arquitectura actual. La llamada llega a través de la RTB al centro de teleasistencia. El centro dispone de una centralita específica que da servicio a un determinado número de líneas de teléfono. Cada teléfono está conectado directamente a la centralita.

Los operadores disponen de una aplicación para la atención de los clientes de teleasistencia. La centralita está configurada a medida por medio de un servidor que convierte los datos a CTI (conocido como integración de telefonía y datos), para que los operadores puedan gestionarlos.

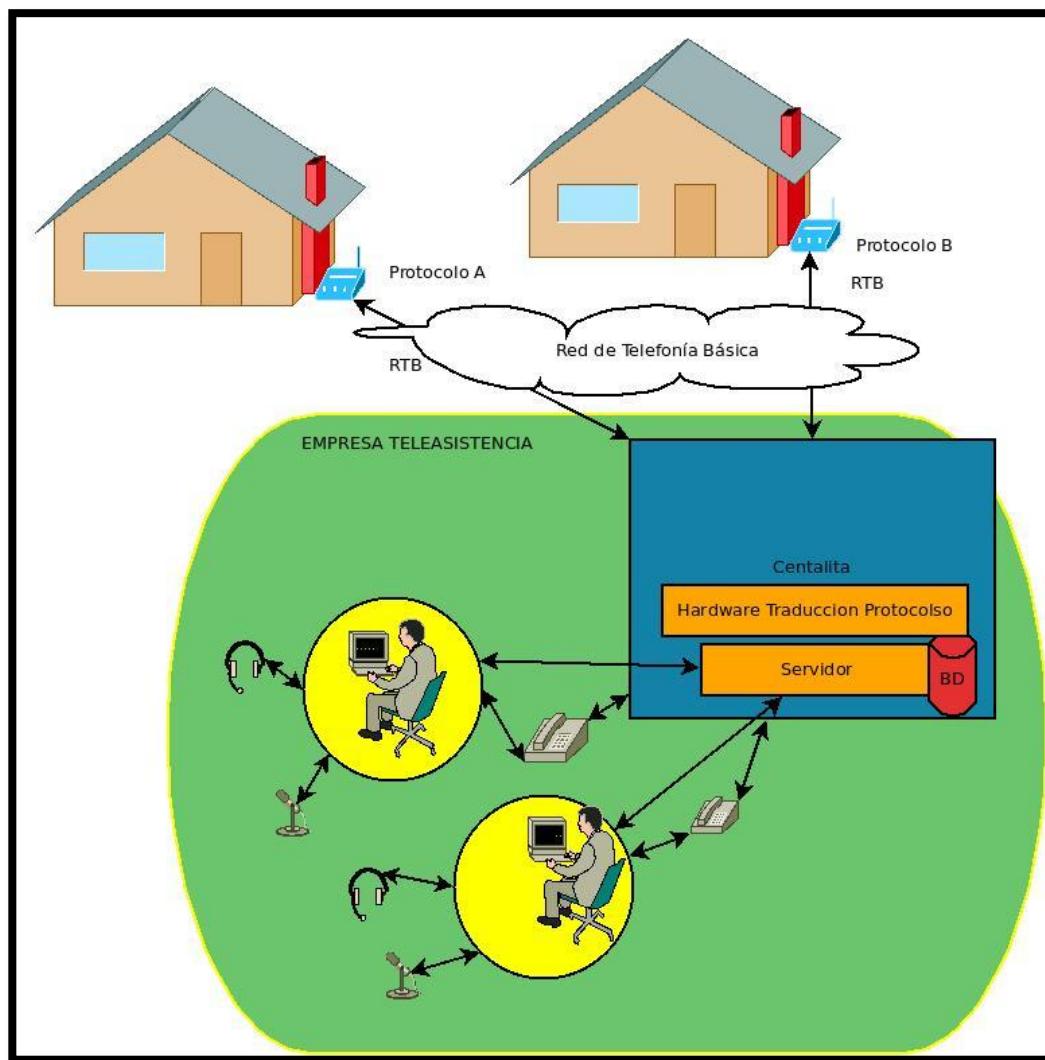


Figura 3-2 Arquitectura actual

Cada fabricante de equipos tiene sus propios protocolos, implementados mediante secuencias de marcación por tonos, DTMF. Para decodificar la información de las alarmas procedentes de los equipos, las empresas cuentan con un hardware capaz de traducir los distintos protocolos.

Cada protocolo es soportado por una tarjeta que traduce un número limitado de terminales simultáneamente. El dispositivo hardware está limitado por el número de ranuras en las que insertar las tarjetas.

Las instalaciones actuales tienen una serie de inconvenientes:

- Alto coste de la centralita: tanto la inversión inicial como el mantenimiento de las mismas elevan los costes de la empresa.
- Alto coste de la solución hardware multiprotocolo: la solución por hardware supone un elevado coste económico.
- Limitación de equipos de clientes de un mismo fabricante: la arquitectura del hardware de traducción de protocolos dificulta la escalabilidad.
- La centralita tiene limitado el número de operadores: si no se dispone de más conexiones para los teléfonos, la centralita, queda obsoleta.

La empresa Verklizan ofrece la solución más completa y cara del momento. Su plataforma UMO soluciona el multiprotocolo mediante hardware e incorpora una centralita analógica para los puestos de los operadores [45].

Por otro lado, la empresa Sabia [46] presenta:

- La central de alarmas B2000: la centralita analógica encargada de la gestión de alarmas y llamadas.
- Puesto del operador B2301: dispositivo externo a la computadora del operador que permite comunicarse con la persona que ha efectuado la alarma.
- Software CSTAtención: muestra las alarmas y expedientes de un cliente.

---

### 3.3 Contexto de la voz IP

La voz IP se define como un conjunto de recursos que permiten que la señal de voz viaje a través de Internet por medio de los protocolos de Internet [21].

Una instalación IP necesita la tecnología necesaria para transformar la red de teléfono convencional en protocolo IP. Esta instalación puede estar en el centro, o se puede obtener contratando el servicio a una empresa especializada en ello.

Una vez insertada la información de voz en un paquete IP y señalizada, se gestiona en una centralita IP para que finalmente se comunique con el resto de dispositivos conectados en la red.

El teléfono IP destino puede ser un dispositivo hardware conectado a la red local o integrado en una aplicación, lo que se conoce como *softphone*.

---

### 3.4 ¿Por qué voz IP?

Si comparamos una empresa con infraestructura voz IP con otra analógica, la gran diferencia es económica.

Para empezar, la inversión inicial es mayor en la analógica debido al coste de sus componentes. Para continuar, si se desea aumentar la instalación, dependiendo de la plataforma:

- Analógica: Un teléfono convencional está conectado directamente a una centralita analógica. El número de conexiones que la centralita soporta está limitado por el número de conexiones analógicas que posea.
- Voz IP: Un teléfono IP está conectado a la red local. La centralita IP no necesitará modificarse a no ser que se sobrepase ampliamente, el número de usuarios respecto a lo previsto inicialmente. Por ejemplo: se haya instalado un servidor configurado para 300-400 operadores y crezca por encima de 500. La solución actual supone la duplicación de hardware, y por consiguiente, de precio.

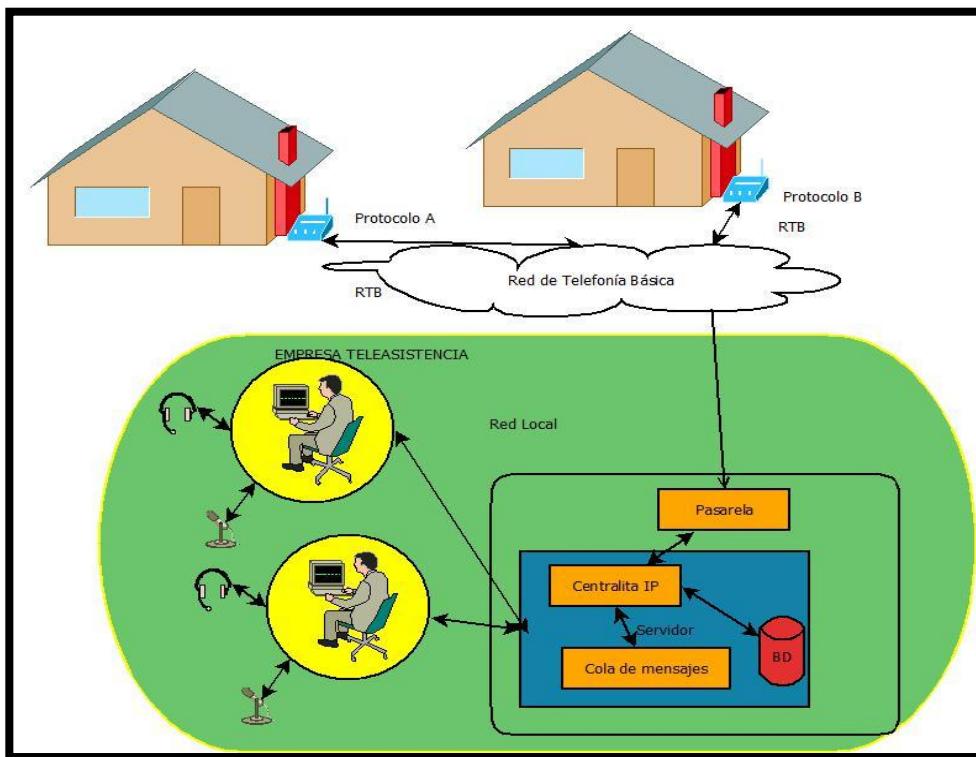
La solución IP es más escalable, incluso muy cómoda en caso de ser subcontratada:

- Si se subcontrata el servicio de telefonía: sólo es necesario solicitar nuevos canales concurrentes a la empresa que ofrece el servicio.
- Si no se subcontrata el servicio: se necesita contratar nuevas líneas de telefonía y adquirir una pasarela para alimentar las líneas. Esta solución es mucho más rentable que la analógica.

Además, una centralita IP puede estar configurada a medida de forma que pueda interpretar los distintos protocolos de terminales. De esta forma, evita el coste adicional que supone adquirir el hardware multiprotocolo y hace el sistema escalable en cuanto al número de terminales soportados por centro. Es decir, si en un área todos los clientes de teleasistencia utilizan los terminales de un mismo fabricante, se garantice dar el servicio a todos.

### 3.5 Arquitectura IP de AsisT

La plataforma en la que se sustenta AsisT es la que se muestra en la Figura 3-3.



**Figura 3-3 Arquitectura IP de AsisT**

En el centro de teleasistencia se instala un servidor que contiene:

- Centralita IP.
- Servidor de cola de mensajería.
- Gestor de base de datos.

En la red local de la empresa está el servidor y los puestos de los operadores.

Esta arquitectura permite que no haya problemas en caso de caída de Internet y que en caso de realizar una llamada de emergencia (112), la centralita gestione la llamada en el área local conveniente (problema si la centralita está en un servidor externo).

En el ANEXO A, se detallan e ilustran las distintas soluciones tomadas por la empresa Diaple Networking.

## 4. ANÁLISIS DE LA INFORMACIÓN

---

### 4.1 Actividades realizadas en el análisis

Para el desarrollo de AsisT se ha realizado un completo análisis de la información siguiendo la metodología de Métrica 3 [30]. La documentación con todas las conclusiones y gráficos está disponible en el ANEXO B.

En esta fase se ha realizado las siguientes tareas:

- Análisis de la información del convenio-marco [5].
- Análisis de los requisitos para un usuario de teleasistencia:
  - o Confección de un catálogo de requisitos funcionales y técnicos.
  - o Analizar los subsistemas en los que está compuesto.
  - o Confección de “casos de uso” de un usuario que accede al sistema AsisT.
- Análisis de la información:
  - o Confección del diagrama entidad – relación.
  - o Confección del modelo entidad – relación.
- Análisis de las clases:
  - o Confección de un diagrama de clases.
- Análisis de los patrones de las ventanas:
  - o Confección de los prototipos de las ventanas.

### 4.2 Resumen de los requisitos de AsisT

Se necesita un sistema capaz de estar instalado en varios puestos de trabajo de una misma red local. Los usuarios, desde su puesto de trabajo deben recibir en tiempo real y de forma simultánea un aviso de alarma. Sólo uno de ellos podrá capturarla y atenderla. Mientras esté atendiendo, no recibirá más alarmas.

Una vez que ha recibido la alarma podrá observar la información personal de la persona que la ha originado. Para la atención de las incidencias, ya sea con llamada o sin llamada, el operador deberá cumplimentar el tipo de alarma y realizar tantas actuaciones como sea necesario para atender satisfactoriamente al cliente.

El sistema se encargará de establecer la llamada para cada actuación seleccionada. La llamada debe realizarse sin marcar el número y será el operador el que libere cuando finalice la conversación.

Todas las alarmas que se generen deben estar almacenadas, así como la información de la gestión que realicen los usuarios. También es necesario tener la información de los clientes en lo referente a datos personales, vivienda, instalación, personas a recurrir en caso de emergencia, asistencia sanitaria, situación del servicio y últimas llamadas atendidas y recibidas. Dicha información debe poder ser creada, modificada o eliminada por un usuario.

El usuario debe poder observar las alarmas que ha atendido y sus detalles, así como las llamadas que ha realizado o aceptado.

La aplicación debe soportar distintos perfiles de usuarios. Se debe implementar la gestión del alta de nuevos usuarios para la aplicación y nuevos clientes.

La aplicación debe ser rápida, segura y cumplir la ley de protección de datos [12]. También debe ser escalable, de forma que pueda expandirse en funcionalidad e información. Debe poder ejecutarse en distintos sistemas operativos.

---

## 4.3 Definición de actores y roles

La aplicación AsisT está analizada para adaptarse a los distintos roles que pueda tener una empresa de teleasistencia, de forma que se gestione los permisos de acceso de funcionalidad apropiados.

En esta versión se han identificado tres actores. Se llama indistintamente con el término usuario a cualquier persona que utilice la aplicación sin importar el rol, por el contrario, para especificar el usuario según el rol se atiende a la siguiente terminología:

- Súper - administrador: acceso a toda la operativa de la aplicación. Tiene privilegios:
  - o Gestión de los usuarios de la aplicación: alta / baja / modificación.
  - o Gestión de los clientes del servicio: alta / baja / modificación del expediente.
  - o Funcionalidad de los operadores.
- Administrador:
  - o Gestión de los usuarios de la aplicación: alta / baja / modificación.
  - o Gestión de los clientes del servicio: alta / baja / modificación del expediente.
- Operador:
  - o Ver información y modificación del expediente de los clientes.
  - o Gestión de alarmas: atención de alarmas entrantes y ver alarmas atendidas y activas.
  - o Ver información de las llamadas entrantes y salientes.

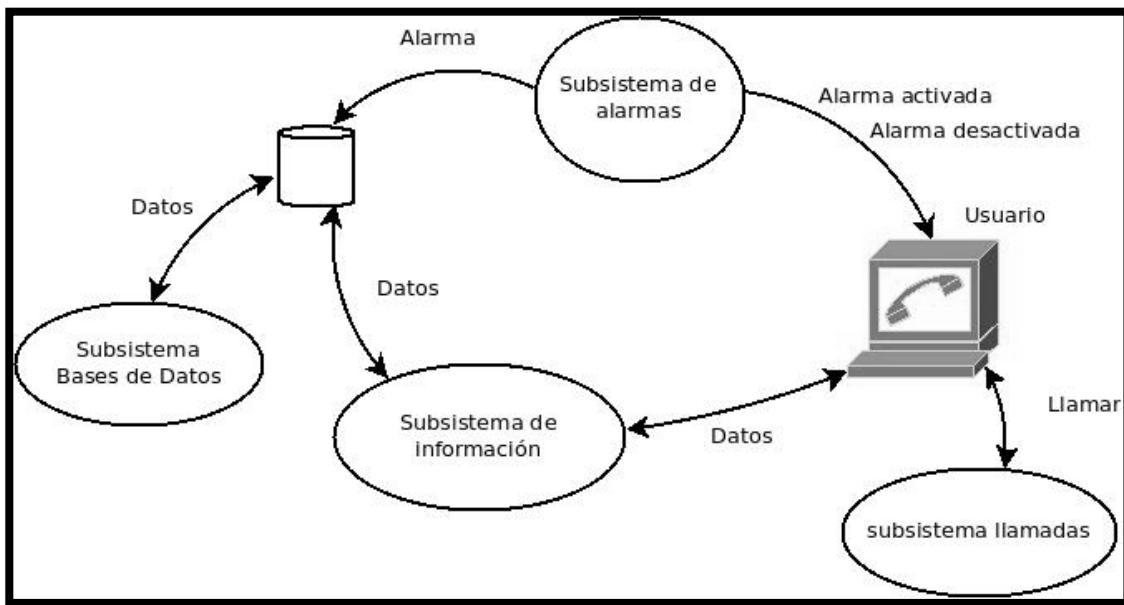
---

## 4.4 Los subsistemas

AsisT es el sistema a desarrollar. Los subsistemas son las partes dentro de AsisT que se han diferenciado según la funcionalidad.

- Subsistema de base de datos: consiste en la realización y mantenimiento de la base de datos necesaria para almacenar toda la información que necesita AsisT en un gestor de base de datos adecuado.
- Subsistema de información: tiene toda la interfaz e interacción con la base de datos. Se han distinguido distintos módulos en función de la información a gestionar:
  - o Módulo de autentificación: gestiona el acceso a la aplicación.
  - o Módulo Expediente-Cliente: información vivienda, instalación, conocidos, etc.
  - o Módulo Clientes: dar de alta un nuevo cliente, listados de clientes, etc.
  - o Módulo Usuarios: dar de alta un nuevo usuario a la aplicación AsisT, listado de usuarios, permisos, etc.
  - o Módulo Alarmas: ver información de alarmas realizadas y gestionar las activas por un operador.
  - o Módulo Llamadas: ver información de las llamadas recibidas y realizadas por un operador.
- Subsistema *softphone*: hace el vínculo entre el subsistema de información y la centralita IP. Se encarga de utilizar el protocolo adecuado para recibir y enviar las llamadas que realice el usuario.
- Subsistema Aviso alarmas: vínculo entre la centralita IP y los usuarios para la gestión de alarmas.

En la Figura 4-1 se muestra la conexión entre los diferentes subsistemas.



**Figura 4-1 Subsistemas**

A continuación, en la página 31 de la memoria, explica lo que se ha implementado de cada uno de los subsistemas: las soluciones empleadas, la funcionalidad implementada y el resultado final.

## 5. DISEÑO DE LA INFORMACIÓN

---

### 5.1 Resumen de la actividad del diseño

El diseño completo de AsisT está reflejado el ANEXO C.

En esta fase del desarrollo se han realizado las siguientes actividades siguiendo la metodología de Métrica 3 [30]:

- Establecimiento de una serie de normas:
  - o Nombrado de la base de datos.
  - o Nombrado de los ficheros de la aplicación.
  - o Nombrado de las excepciones de error.
- Diseño de los Escenarios de la aplicación
  - o Confección de los diagramas de secuencia de los escenarios.
  - o Confección de los diagramas de actividad.
- Identificación de los patrones de diseño:
  - o Patrones de diseño ya existentes: VO, DAO, DAOFactory:
    - Confección del diagrama DAOFactory.
  - o Patrones de diseño localizados por repetición de actividad:
    - Confección del diagrama del patrón.
- Confección del diagrama de clases orientado a objetos.

---

### 5.2 Patrones de diseño

La importancia de los patrones de diseño se debe al ahorro en tiempo de implementación que supone aplicar métodos que facilitan el mantenimiento y la escalabilidad. Permite que el software sea reusable, progrese y sea estructurado.

En este apartado se explica el patrón repetido por toda la aplicación. Para ver más detalle véase en los Anexos.

#### 5.2.1 Interfaz - gestión

AsisT está formada por una información destinada a ser almacenada y gestionada. La información está almacenada en la base de datos y siempre que se gestiona, se utilizan los objetos como contenedores para posteriormente interactuar con la base de datos. El patrón que se repite es:

- Mostrar por pantalla una tabla/lista/paneles con información.
- En algunos casos cada fila se puede añadir/modificar/eliminar mediante una ficha.

Para abordar el flujo de información entre la interfaz y la base de datos se han recurrido a los patrones de diseño VO (Value Object) – DAO (Data Access Object).

- Clase VO: contenedor para la información.
- Clase DAO: funciones propias para interactuar entre la aplicación y el gestor de base de datos.

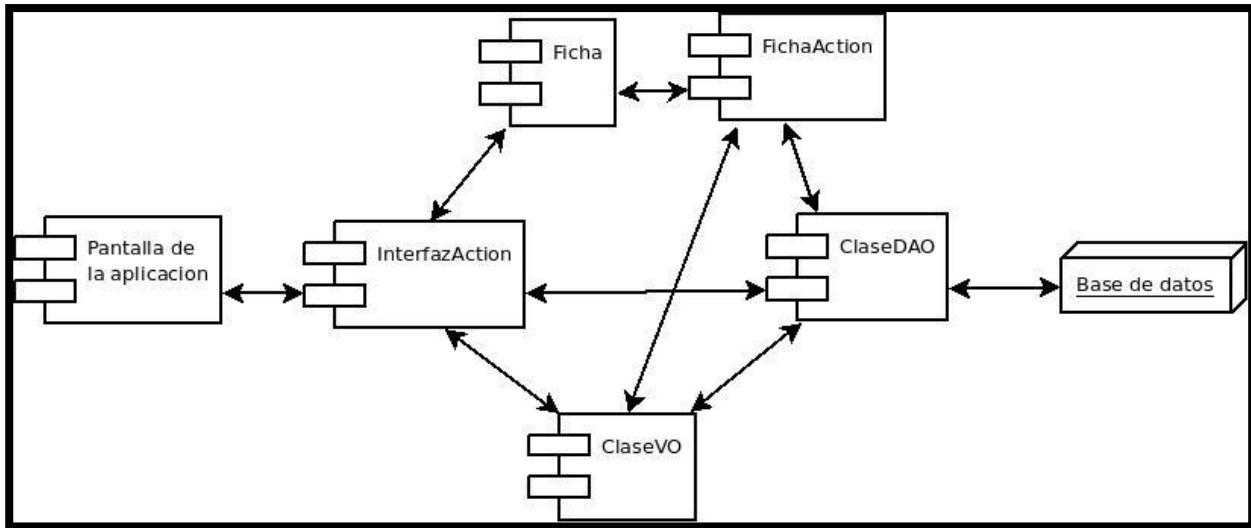
Siempre que se interactúa con la base de datos, es necesario construir un objeto VO de forma que la clase DAO se encargue de gestionarlo con la base de datos.

La funcionalidad se ha aislado entre capas:

- Interfaz: el panel que se muestra por pantalla, personalizado para cada módulo.

- InterfazAction: la lógica de negocio de la información que se obtiene por pantalla.
- ClaseDAO: contiene las búsquedas, inserción y actualización de datos para las clases VO.

La Figura 5-1, ilustra cada una de las clases y su interacción. Desde la lógica de negocio del *InterfazAction* se accede a otra interfaz, *Ficha*, cuya pantalla son los campos para introducir nuevos datos.



**Figura 5-1 Interfaz - gestión**

Por ejemplo, el gestor clientes de teleasistencia se ha implementado con dicho patrón en el que:

- Pantalla de la aplicación: interfaz de una tabla con botones de añadir y modificar.
- InterfazAction:
  - o Carga la información de los clientes en la tabla (instancia *ClaseDAO* para obtener el listado)
  - o Controla la lógica de botones para introducir o modificar clientes (instancia *ClientesFicha*)
- Ficha: interfaz de una ficha con los campos de un cliente.
- FichaAction:
  - o Ficha nueva para crear un nuevo cliente o modificarlo.
  - o La lógica de guardar o modificar datos (instancia *ClaseDAO* para utilizar las funciones como insertar, actualizar, eliminar, etc).
- ClaseDAO: *clienteDAO* tiene todas las funciones de obtención de datos referente a un cliente.
- ClaseVO: *clienteVO* tiene los métodos para obtener y dar valor a sus atributos.

### 5.3 Elección del tipo de aplicación

Una decisión importante del diseño es la elección del tipo de aplicación, que puede ser web o de escritorio. A continuación se analizan los factores que determinan la elección.

Es de vital importancia recibir una alarma en tiempo real. Esto significa que en el momento que la centralita ha recibido una alarma, todos los operadores deben ser avisados al instante para procesarla. Además, en ese mismo momento, deben acceder a información de diferente índole: del cliente que ha activado la alarma, como de anteriores alarmas que han atendido, etc.

Una aplicación de escritorio es síncrona, por otro lado, una aplicación web puede ser síncrona si se utilizan herramientas que simulan una aplicación de escritorio, dentro de la propia web. Aun así, las soluciones existentes no garantizan a nivel de aplicación de escritorio, el tiempo real.

El *softphone* abre una conexión a bajo nivel al puerto del puesto del trabajo. Existen complicaciones de abrir un socket en la aplicación web, a no ser que se utilice Active X [53], un aplicación de Flash [51] o una aplicación de Java (aplicación en contexto web o *applet*) [52]. La implementación se complica con estos lenguajes que finalmente simulan una aplicación de escritorio.

Por último, en caso de que la arquitectura se ofrezca en un servidor externo a la empresa de teleasistencia, si no hay conexión de Internet, el servicio queda inactivo.

Una aplicación de escritorio consume más recursos y su mantenimiento en caso de actualizaciones es más costoso que una aplicación web. No obstante, estos factores son intrascendentes comparado con la gran importancia de los anteriores.

La aplicación de escritorio es la solución idónea al problema AsisT.

---

## 5.4 Elección del lenguaje de programación de AsisT

Los requisitos que debe cumplir el lenguaje de programación son:

- Aplicación en tiempo real: necesario que reciba la información en el mismo momento que se haya transmitido desde el servidor.
- Multiplataforma: la aplicación funcione independientemente del sistema operativo.
- Disponibilidad de APIs: facilite la programación dado a la existencia de librerías. En concreto con varios gestores de bases de datos populares, cola de mensajería y *softphone*.

.Net [47] y Java [6] son los lenguajes de programación más populares para aplicaciones de escritorio. Ambos disponen entornos de programación para ayudar a crear las aplicaciones.

La diferencia entre ambos es que .Net no es multiplataforma, sólo está soportado por el sistema operativo de Windows. Java es multiplataforma y posee el API para la implementación del *softphone* y para diferentes colas de mensajería (ambos aspectos de abordan en la sección 6).

Por tanto, Java es el lenguaje elegido para realizar la aplicación.

---

## 5.5 Elección del lenguaje de programación que integra la centralita IP con AsisT

El escenario de la integración es que la centralita recibe la alarma. En este punto se necesita un pequeño programa que ejecute la centralita por medio de un interfaz que dispone (AGI [54]). El programa tiene que introducir los datos de la alarma en la base de datos y comunicarla con la cola de mensajería para que pase a los operadores.

Las características que debe cumplir son:

- Rápido en el arranque y ejecución
- El lenguaje sea actual para facilitar el mantenimiento y la ampliación
- Compatible con el interfaz AGI.

## Diseño de la información

---

Java [6] es un lenguaje que penaliza en tiempo de inicialización debido a la máquina virtual.

C también es un lenguaje que puede ser solución ya que no tiene la penalización que ocurre con Java.

No obstante, PHP [32] es un lenguaje interpretado que evitaría también penalización en arranque. Además dispone de las librerías de mensajería y base de datos de modo que facilita la programación en comparación con C.

Por lo tanto este programa será realizado en PHP.

## 6. SISTEMA ASIST

Tal y como aparece en el apartado de análisis, AsisT se ha dividido en subsistemas.

En este apartado se explica cada uno de ellos: funcionalidad, soluciones adoptadas e integración con el resto de componentes.

Los detalles de la implementación se reflejan en el ANEXO D y aspectos de funcionalidad e interfaz en el “Manual de usuario” en el ANEXO E.

---

### 6.1 Subsistema de bases de datos

#### 6.1.1 Elección gestor de bases de datos

AsisT necesita un gestor de base de datos apropiado cuyas características sean:

- Compatible con Java [6] por medio de un API ya elaborado.
- Compatible con la plataforma en la que se programa: Linux [8].
- Soporte de claves foráneas.
- Soporte de transacciones.
- Rápido, seguro y robusto.
- Software Libre: aporte las ventajas de ya comentadas.

Se han escogido para el estudio los gestores de MySQL [9] y PostgreSQL [10]. Prácticamente son muy similares: aceptan distintos tipos de datos, creación de funciones, transacciones, claves foráneas, disponen de API de Java, etc. No obstante PostgreSQL consume más recursos que MySQL.

Debido a la importancia de que la aplicación sea rápida, El gestor elegido para implementar la base de datos de AsisT ha sido MySQL.

El gestor de almacenamiento en el que se ha construido es InnoDB [27], ya que es condición indispensable la existencia de claves foráneas.

#### 6.1.2 Implantación base de datos

La base de datos implantada es la analizada en el análisis del sistema, el esquema entidad relación puede verse en el ANEXO B, “Actividad ASI 6”.

Las características de la base de datos son:

- Está normalizada hasta 3FN.
- Analizada para ser escalable: la base de datos está diseñada para que pueda crecer en información sin tener que rediseñarse.
- Todas las tablas contienen 6 campos imprescindibles para el cumplimiento de la Ley de protección de datos [12].
  - o Fechas de: creación, modificación y eliminación de la información
  - o Persona que ha: creado, modificado y eliminado la información

#### 6.1.3 Integración centralita IP con base de datos

Llega una alarma a la centralita IP. Tras decodificarla según el protocolo del terminal que proviene, se dispone de la siguiente información:

- CIU del cliente: identificador único del terminal que ha producido la alarma.
- Tipo de alarma: la centralita obtiene la información del tipo de alarma.

Todos los dispositivos de una casa están conectados al terminal y siempre será éste el que emita la señal de alarma.

Para interpretar el multiprotocolo, la base de datos tiene una tabla con todos los protocolos de fabricantes y otra tabla que conecta el protocolo con el tipo de alarma que corresponde. Se ha implementado un script en PHP [32] que introduce la alarma en la base de datos. Este programa ha sido realizado con la herramienta Aptana [14].

Este script parte como entrada del código de identificación del usuario (CIU), y del tipo de alarma:

- Crea la alarma con los datos de fechas, estado y cliente.
- Inserta en la base de datos utilizando el API de MySQL disponible para PHP.

El script es ejecutado por la centralita IP, por medio del interfaz AGI [54]. Ésta dispone de las librerías y mecanismos para ejecutarlo dentro del servidor. No es necesario por tanto, un servidor web para su ejecución como se acostumbra a relacionar PHP.

---

## 6.2 Subsistema de atención de alarmas

En primer lugar se explica cómo AsisT soluciona e integra el problema de las alarmas. Los operadores están conectados a la red local del centro de teleasistencia. Todos los operadores están conectados a un servidor local formado por:

- Centralita IP
- Base de datos

Los operadores necesitan ser avisados de dos tipos de eventos:

- Llegada de una alarma nueva al centro de teleasistencia: se informa por pantalla a todos los operadores.
- Quitar el aviso de la alarma ya que un operador la ha capturado: todos los operadores deben dejar de recibir el aviso.

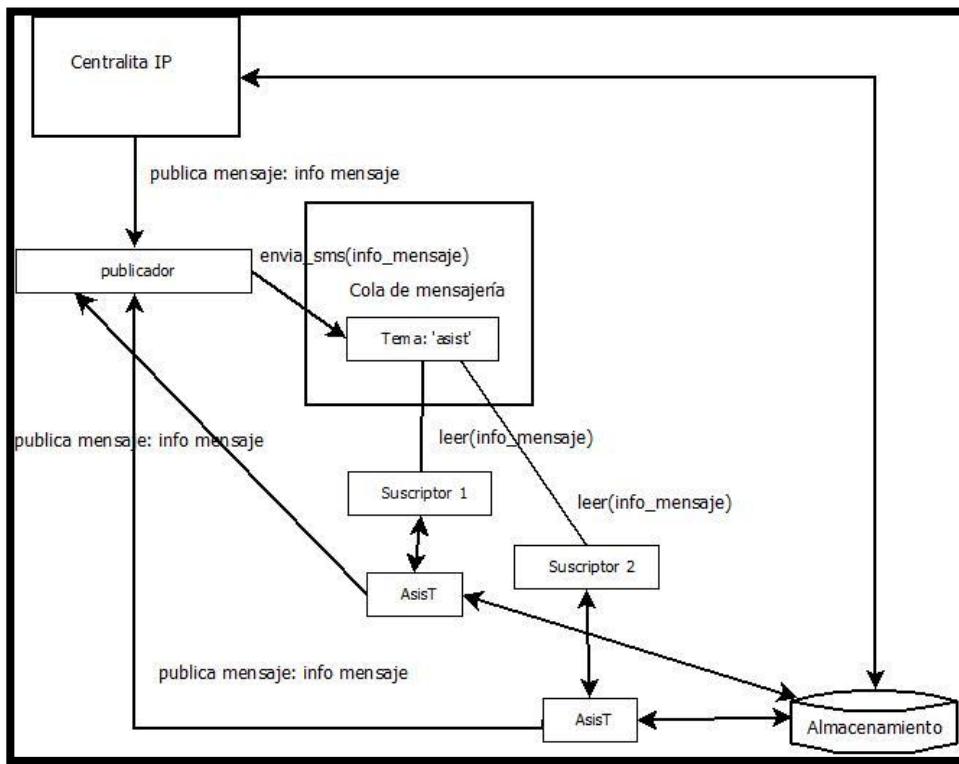
### 6.2.1 Solución al aviso de alarmas

La solución para el aviso de alarmas es la utilización de una cola de mensajes (*Message Broker*) de forma que los usuarios que estén conectados a la red local y que hayan inicializado la aplicación AsisT, puedan dar parte de la llegada de un evento y actuar en consecuencia.

El modelo empleado es publicador/suscriptor y los eventos son de tipo síncronos. Cuando se quiera notificar un evento al resto de suscriptores, se instancia la clase publicador, para que escriba un mensaje en un tema de la cola de mensajería.

La cola de mensajería es un servidor que se ha instalado localmente junto con la base de datos y la centralita IP. El objetivo: conectar la centralita IP – alarmas – operadores.

AsisT puede ser publicador o suscriptor según el tipo de evento que desea notificar. Como puede observarse en la Figura 6-1. Al iniciar la aplicación, se convierten en suscriptores del tema ‘asist’, y a partir de este momento leen los mensajes publicados. Por el contrario, la centralita IP sólo notifica eventos. En la sección 6.2.3 se detallan los eventos y actuaciones ante los avisos.



**Figura 6-1 Mensajería modelo publicador / suscriptor**

### 6.2.2 Elección del servidor de mensajería

Para elegir un servidor de mensajería acorde con las características de la aplicación se destacan las siguientes necesidades del sistema:

- Compatible con java (*Java message service, JMS* [18]): de forma que AsisT interaccione con el servidor.
- Compatible con PHP [32]: script PHP que ejecuta la centralita IP para publicar un mensaje.
- Software libre.

De los numerosos servidores de colas de mensajes disponibles, se destacaron los siguientes por popularidad y reciente actividad: ActiveMQ [24], OpenJMS [25] y RabbitMQ [26].

OpenJMS ha sido descartado debido a la incompatibilidad con PHP. ActiveMQ y RabbitMQ son muy similares: ofrecen la posibilidad de ser multiplataforma y varios servicios para facilitar la gestión como un gestor web para publicar mensajes.

Finalmente se eligió ActiveMQ debido al soporte por parte de la comunidad Apache. La comunidad Apache ofrece confianza en mantenimiento y seguridad.

### 6.2.3 Integración mensajería con AsisT

La centralita ejecuta un script en PHP [32] en el que:

- Introduce la alarma en la base de datos.
- Se conecta a la cola de mensajería ActiveMQ [24] y publica un mensaje anunciando el evento: "alarma".

El mensaje tiene la siguiente estructura:

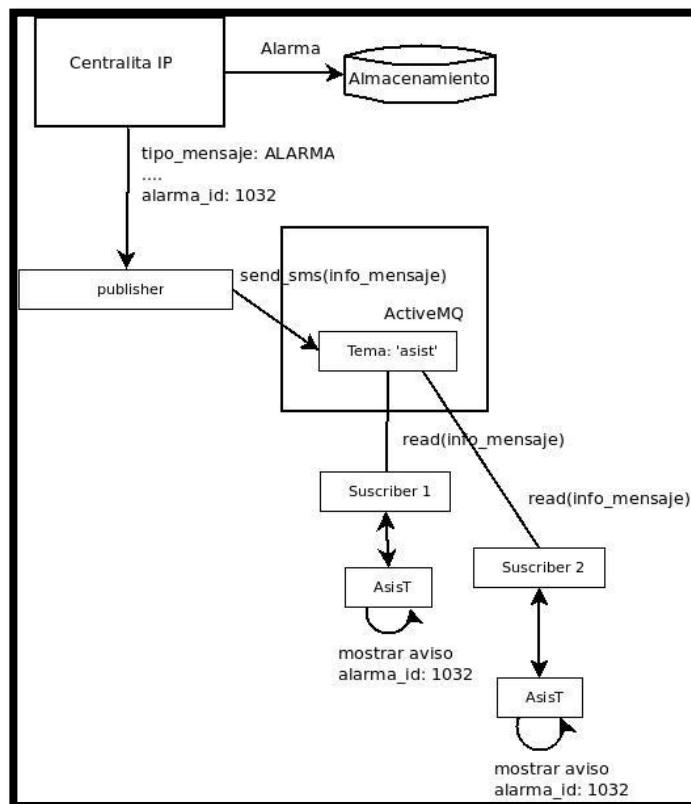
```
message_type=ALARMA
message_source=ASTERISK
message_created=(FECHA_HORA_ACTUAL)
alarma_id=1201
```

Para integrar la mensajería con Java se ha hecho uso del API JMS [18]. Al iniciar la aplicación, se suscribe a un tema: ‘asist’ en el proveedor de mensajería ActiveMQ.

Desde este momento, el sistema está a la escucha de los mensajes publicados en tema. Cuando la aplicación detecta que se ha publicado un mensaje, según el tipo de evento, se interpreta para actuar en consecuencia: mostrar o quitar aviso de alarma.

### Llegada de alarma

Figura 6-2 ilustra cómo el sistema AsisT es notificado ante el evento de una alarma nueva. El mensaje escrito por la centralita es de tipo alarma, los suscriptores leen el mensaje y muestran el aviso de la alarma.



**Figura 6-2 Aviso de alarma**

AsisT inicia una ventana emergente, véase Figura 6-3 , en la esquina inferior derecha en todas las aplicaciones de los usuarios que están inicializadas. Dicha ventana tiene información de la alarma del cliente al que pertenece, el tipo de alarma, la fecha y la hora.



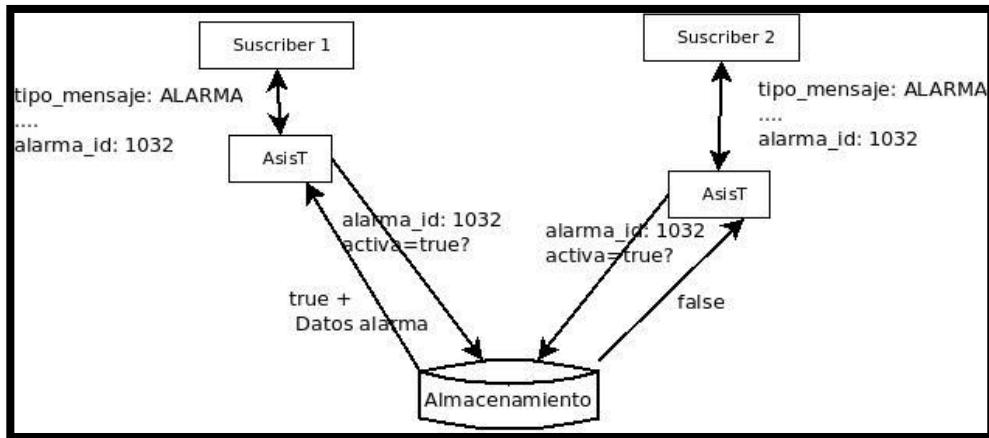
**Figura 6-3 Ventana aviso de alarma**

A medida que se reciban nuevas alarmas, estarán listas para ser capturadas por el usuario. En esta primera versión, cuando un usuario esté atendiendo una alarma, no recibirá nuevos avisos.

### Capturar la alarma

La Figura 6-4 ilustra cómo AsisT captura una alarma. En la ventana emergente de “aviso de alarma”, los operadores pueden seleccionar “capturar alarma” e iniciar la gestión de la misma.

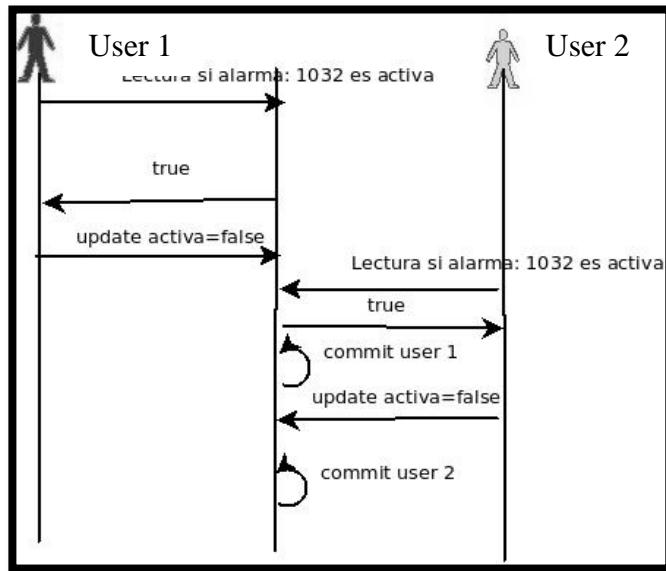
El proceso de que un operador obtenga la alarma, comienza en comprobar si no ha sido ya capturada por otro operador. Para ello, se hace una lectura en la base de datos y se comprueba el estado. La Figura 6-4, muestra la interacción entre los componentes. El operador ganador es el que ha leído la alarma que está desactivada. Una vez cambiado el estado a activada, los operadores no podrán capturarla.



**Figura 6-4 Caputura de alarma**

Una alarma sólo puede ser atendida por una persona, dos usuarios nunca pueden atender una alarma simultáneamente. Esto provocaría que los datos almacenados no sean los correctos.

La Figura 6-5 ilustra la situación de error. Los dos usuarios hacen la consulta en la base de datos. El primero en llegar, ha modificado la alarma indicando que ya no está activa. En un periodo pequeño de tiempo, los cambios no se han guardado de forma permanente en la base de datos (no ha realizado *commit*). El problema surge cuando el segundo operador, lee que está libre, accediendo a la misma información y gestionándola a la vez.

**Figura 6-5 Error en captura**

La base de datos está implementada en el gestor de almacenamiento InnoDB [27] que permite transacciones. El modelo de transacciones permite lecturas en modo de bloqueo. La función “select... for update” bloquea la captura de la alarma, de esta manera, cuando la aplicación realiza la consulta, sólo un usuario accede en exclusividad [48]. Aunque se pulse casi simultáneamente para capturar la alarma, sólo uno la obtiene y el perdedor, es notificado por pantalla.

### **Cancelar alarma para el resto de suscriptores**

Un usuario ha aceptado la alarma, por lo tanto debe desaparecer la ventana emergente de aviso de alarma para todos los usuarios que están conectados a la aplicación.

En este momento, AsisT se comporta como publicador y publica un mensaje en ActiveMQ con la siguiente información:

```

message_type=ALARMA_CAPTURADA
message_source=usuario_id
message_created=(FECHA_HORA_ACTUAL)
alarma_id=1201
  
```

Donde “usuario\_id” es el operador que la ha capturado y “alarma\_id” es el identificador de la alarma.

Todos los usuarios leen dicho mensaje en el Tema “asist” del Active MQ, incluido el usuario que la ha capturado. La aplicación actúa eliminando el aviso de alarma por pantalla.

La Figura 6-6 ilustra que el “suscriptor 1” captura la alarma e informa al resto de aplicaciones activas por medio de ActiveMQ para quitar el aviso.

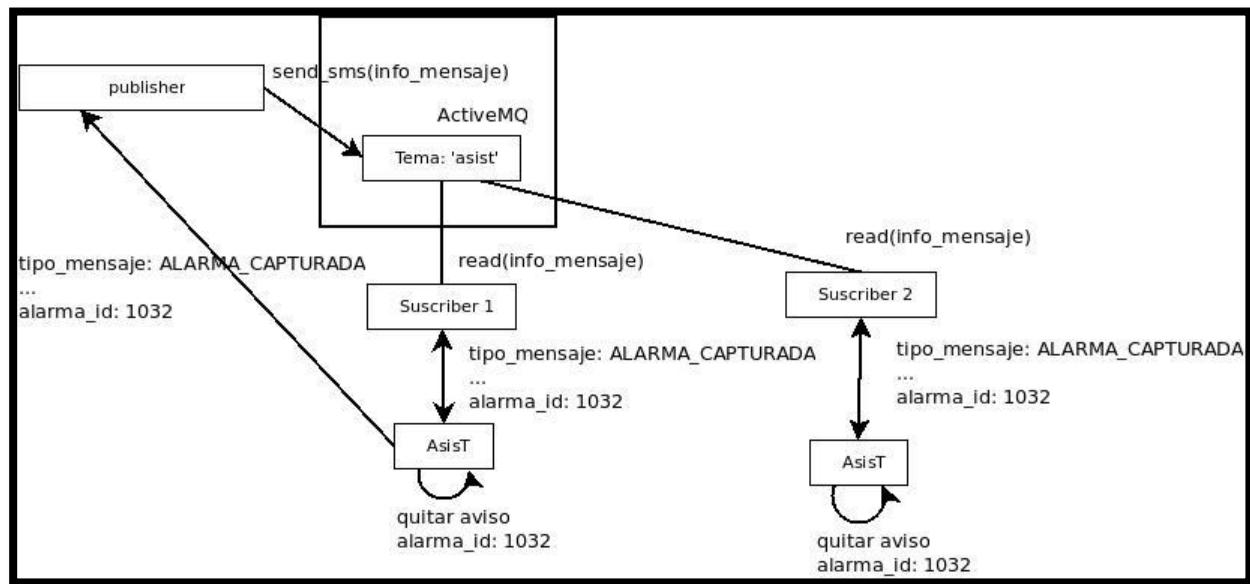


Figura 6-6 Alarma capturada

### 6.3 Subsistema teléfono

El *softphone* es el teléfono integrado en la aplicación. En otras palabras, es la implementación de las funcionalidades típicas de un teléfono y de la gestión de los canales de voz y datos con la centralita IP.

Para que una llamada desde la aplicación pueda salir a la red convencional o una llamada desde el exterior pueda entrar a la red local, se necesita una centralita IP configurada a medida.

Un teléfono IP puede ser un dispositivo externo cuya diferencia con un teléfono convencional es que está conectado a la red local por la que recibe los datos de voz IP.

La primera versión de AsisT está configurada de modo que el teléfono está integrado en la aplicación. Para ello, el puesto de los operadores necesita una tarjeta de sonido integrada para la entrada de altavoces y micrófono.

Para que un teléfono sea operativo en la aplicación necesita:

- Conectarse satisfactoriamente con la centralita IP.
- Realizar llamadas y aceptarlas a partir de la aplicación AsisT.
- Señalizar con la centralita la información siguiendo el protocolo de la misma.

#### 6.3.1 Centralita IP

La centralita IP está instalada y configurada en un servidor por medio del software libre Asterisk [22]. Los detalles programados en Asterisk quedan fuera del alcance del proyecto.

El protocolo IP utilizado por la centralita para comunicarse en la red, es SIP (*Session Initiation Protocol*) [2]. Una vez que llega la red de telefonía básica (RTB) al centro de teleasistencia, la pasarela (*gateway*), la conecta con SIP para poder comunicarse con la centralita y el resto de dispositivos conectados en red. Entre otros dispositivos, están todos los puestos de los operadores con la aplicación AsisT operativa.

En la Figura 6-7 se observa la plataforma IP donde los elementos conectados en la red local son el puesto de trabajo, teléfono IP, servidor, y la pasarela. El servidor contiene la centralitaIP, la base de datos y la cola de mensajería.

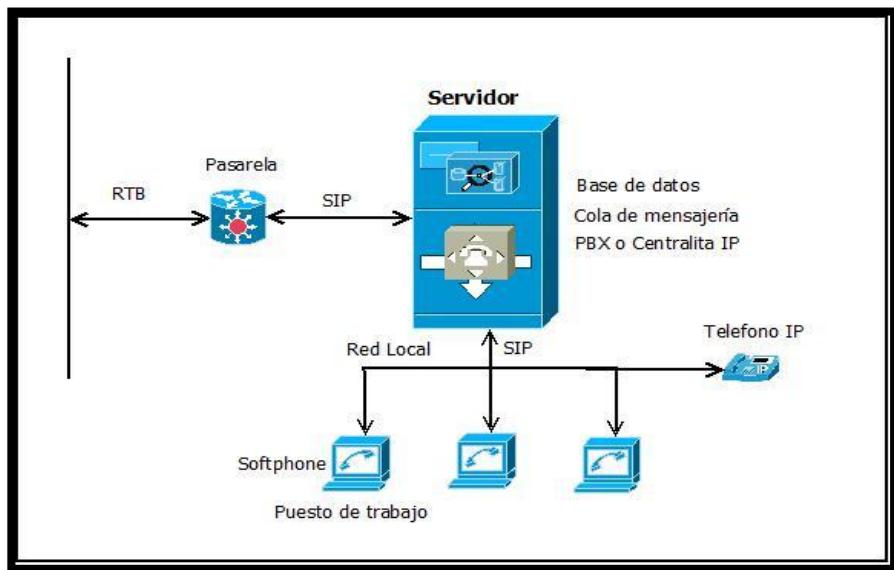


Figura 6-7 Plataforma IP

### 6.3.2 Solución adoptada

Para implementar la conexión y comunicación con la centralita se ha hecho uso de la librería Jain [1], que es un API disponible para Java [6] que ofrece las funciones para construir mensajes SIP en bajo nivel.

El subsistema está formado por tres capas de forma que abstrae la funcionalidad de la capa inferior:

- 1<sup>er</sup> Nivel: funciones que utiliza el subsistema de información de AsisT para establecer conexión, realizar una llamada y terminarla.
- 2º Nivel: funciones para construir paquetes de SIP acordes con la funcionalidad y configuración deseada.
- 3º Nivel: librería Jain: aporta funciones para la construcción de la pila de SIP.

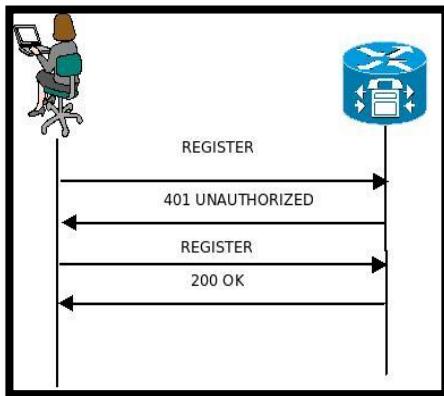
Se ha creado una nueva tabla en la base de datos, que almacena la configuración del teléfono de la aplicación. Cuando AsisT inicia la aplicación, establece la conexión con Asterisk. Si la configuración es la correcta, el sistema queda a la escucha de recibir eventos.

El protocolo SIP se basa en el envío de mensajes. La aplicación envía mensajes: peticiones y respuestas. Una petición es cualquier mensaje con un tipo en concreto: *invite*, *request*, *ok*, *ack*, *bye*, etc. Una respuesta es una petición con un tipo, a consecuencia de recibir una petición por parte de la centralita. [2]

Para más detalle de las funciones implementadas, véase la sección D.8, en el ANEXO D.

### 6.3.3 Conexión con la centralita

La Figura 6-8 detalla la secuencia implementada para establecer la conexión con Asterisk [22]. El sistema se conecta a Asterisk por identificación por “paso de testigo” (*token passing*). El primer mensaje de AsisT siempre es sin éxito y el servidor responde con un mensaje con un identificador propio para asegurar la conexión. AsisT responde reenviando el mensaje con los datos de autenticación del puesto de trabajo. Si son correctos, la conexión se ha realizado con éxito.

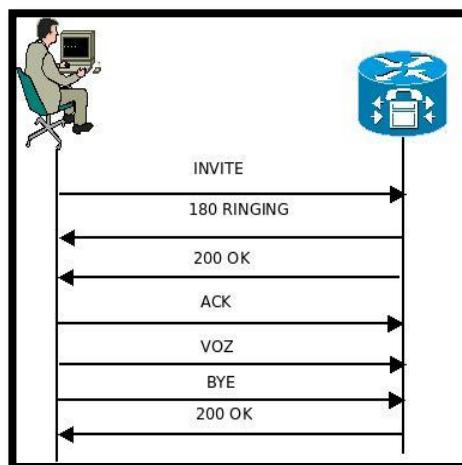


**Figura 6-8 Conexión con la centralita**

### 6.3.4 Establecer una llamada

Una llamada se inicia con una petición de tipo *invite*. La Figura 6-9 muestra la secuencia de mensajes SIP. AsisT realiza:

- Cuando pulse el botón de inicio de llamada, se manda la información del teléfono configurado en ese puesto.
- Se realiza el paso de mensajes SIP entre los dos dispositivos de forma que confirmen ambos la conexión.
- Se abre el canal de audio.
- Finaliza la llamada (*bye*).



**Figura 6-9 Realizar una llamada**

Todas llamadas salientes permitidas por la aplicación son a consecuencia de una alarma. Cuando se indica la actuación más acorde con la incidencia, aparece un listado de números de teléfono que cuando confirme, establece directamente la llamada. El operador, no introduce por su cuenta ningún otro número de teléfono para realizar una actuación, para evitar el mal uso de la aplicación.

### 6.3.5 Aceptar una llamada entrante

Una llamada entrante llega a la centralita a consecuencia de una alarma:

- Asterisk identifica que la alarma que ha recibido es de tipo llamada.
- Asterisk mantiene la llamada en espera hasta que un operador la capture.
- Cuando es capturada, AsisT responde con una llamada a Asterisk.
- Se realiza el paso de mensajes SIP hasta finalmente se abre el canal de voz. Véase la Figura 6-9 de la sección anterior.

- El operador habla con el cliente hasta que él finaliza la llamada (siguiendo las especificaciones del convenio-marco [5]).

### 6.3.6 Interfaz del teléfono

El estado del teléfono está siempre visible en la esquina superior derecha. Los mensajes de estado son:

- “Esperando llamadas”: una vez que el teléfono se ha registrado correctamente y se ha terminado una llamada.
- “Registrando teléfono”: ha lanzado la petición para conectarse y espera confirmación.
- “Recibiendo llamadas”: el *softphone* tiene una llamada entrante.
- “Marcando <número\_al\_que\_marca>”: ha iniciado la llamada pero el teléfono del destinatario, aún no está sonando.
- “Teléfono no registrado”: cuando los datos de configuración del usuario no son correctos.
- “Hablando...”: el canal de audio ha sido abierto.
- “Estableciendo conexión”: cuando esté conectado con la centralita IP.
- “Llamada <número de teléfono>”: en el caso que esté realizando una llamada o recibiendo. Aparece la información del número de teléfono y del tiempo actualizado de la llamada.
- “Sonando”: cuando tiene una llamada entrante.
- “Denegado”: cuando la centralita deniega establecer una llamada con dicho número.
- “Prohibido”: cuando no está permitido llamar a dicho número.

En la Figura 3-2, se observa el *softphone* en diferentes estados. Cuando no hay una llamada activa, el *softphone* tiene el botón “llamar”. Por el contrario, cuando está hablando, el botón “colgar”.



Figura 6-10 *Softphone* sonando - hablando – en espera

### 6.3.7 Integración con la base de datos

Cuando se realiza o recibe una llamada, la centralita o Asterisk se encarga de introducir la información. Se ha realizado un programa en PHP, similar a introducir una alarma que introduce y modifica los datos de la llamada en la base de datos, de forma que puedan ser visualizados en el expediente del cliente y en los historiales del operador.

### 6.3.8 Chequeo de un equipo

Chequear un terminal consiste en conocer en tiempo real desde la aplicación AsisT, el estado del dispositivo: “activado” o “desconectado”.

Desde el expediente del cliente, en la pestaña instalación, se gestionan los equipos que dispone el cliente en su hogar. Cuando se añade un nuevo equipo su estado es “pendiente”. El botón “chequear” comprueba el estado del dispositivo.

Un chequeo se reduce a realizar una llamada hacia el dispositivo, en la que no abre el canal de audio. Si responde correctamente, el estado es “activado”.

## 6.4 Subsistema información

Este subsistema se caracteriza por contener toda la interfaz y procesamiento de información de la aplicación. Se ha construido siguiendo el patrón para el acceso a base de datos VO-DAO y para la interfaz como View – Action.

El orden en el que se explican no es el de implementación. Se trata de explicar lo que se ha implementado en cada uno y la funcionalidad que aporta al sistema.

### 6.4.1 Contexto del funcionamiento

AsisT es una aplicación de escritorio. Para entrar en la aplicación hace falta autentificarse con nombre y contraseña. Según el perfil y permisos de usuario la aplicación tiene vistas distintas. Los administradores, observarán el “módulo de administración”, mientras que los operadores el “módulo de tele-asistencia”. En los ejemplos descritos durante la memoria, se considerara el perfil de “super-administrador” sin restricción de acceso.

Una vez que los datos son correctos, se accede a la pantalla principal. Está formada por un menú a la izquierda, que siempre está visible y el resto es el centro de operaciones para la gestión de la información.

Cuando arranca la aplicación se suscribe a la cola de mensajes, se inicializa el *softphone* y se establece conexión con la base de datos.

La Figura 6-11 muestra el inicio de la aplicación AsisT: barra de estado superior, menú y centro de operaciones. En el centro de operaciones siempre está visible el “Inicio” o cuadro de mandos, cuyos detalles se explican en el apartado 6.4.2.

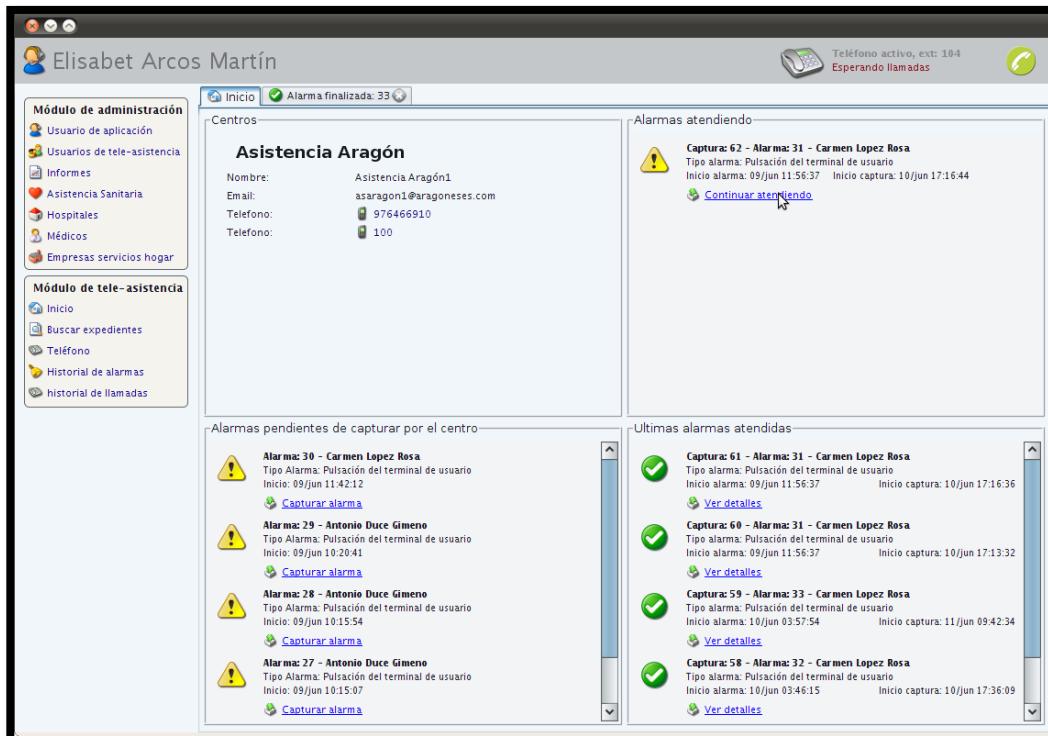
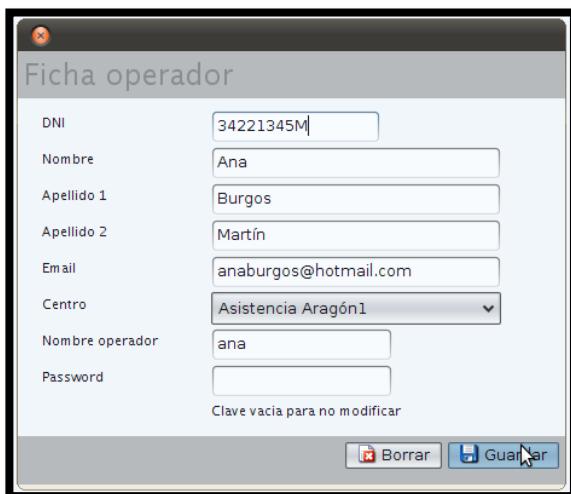


Figura 6-11 Inicio aplicación AsisT

Cuando se pulsa un botón del menú, se abre una pestaña en el panel central con dicha información. Se pueden abrir tantas pestañas como el usuario necesite, de la misma forma que se pueden cerrar. La aplicación no duplica una pestaña si ya está abierta.

Para crear o modificar datos se inicia una ventana emergente en forma de formulario o ficha, como se muestra en la Figura 6-12.



**Figura 6-12 Ficha operador**

#### 6.4.2 Cuadro de mandos

El cuadro de mandos o inicio es la pantalla que siempre está visible en el centro de operaciones de la aplicación. El objetivo de esta pantalla es ofrecer al usuario acceso rápido a información relevante: las alarmas. La información que se muestra es:

- Información del centro: toda llamada saliente necesitará una previa identificación. Dicha información es necesario que esté visible en todo momento.
- Alarmas que está atendiendo el operador: en esta primera versión se ha decidido que un usuario atienda una sola alarma. En caso de cerrarla por error en la gestión de la misma, rápidamente accede a ella en este panel.
- Alarmas activas en el centro: las ventanas emergentes se superponen en la esquina superior derecha. La pila de alarmas completa está disponible en el cuadro de mandos. Estas alarmas aún no han sido capturadas por ningún operador y sólo puede acceder a ellas si no está atendiendo ninguna alarma.
- Últimas alarmas atendidas: en ocasiones puede haberse cerrado una alarma erróneamente. Podrá reactivarla accediendo a los datos de la alarma finalizada.

El cuadro de mandos se actualiza cada vez que se pulsa sobre la pestaña a excepción del panel de alarmas activas en el centro, que sufre un refresco cada 5 segundos si el usuario está en esta visualizando la pestaña, para garantizar la coherencia con el resto de operadores.

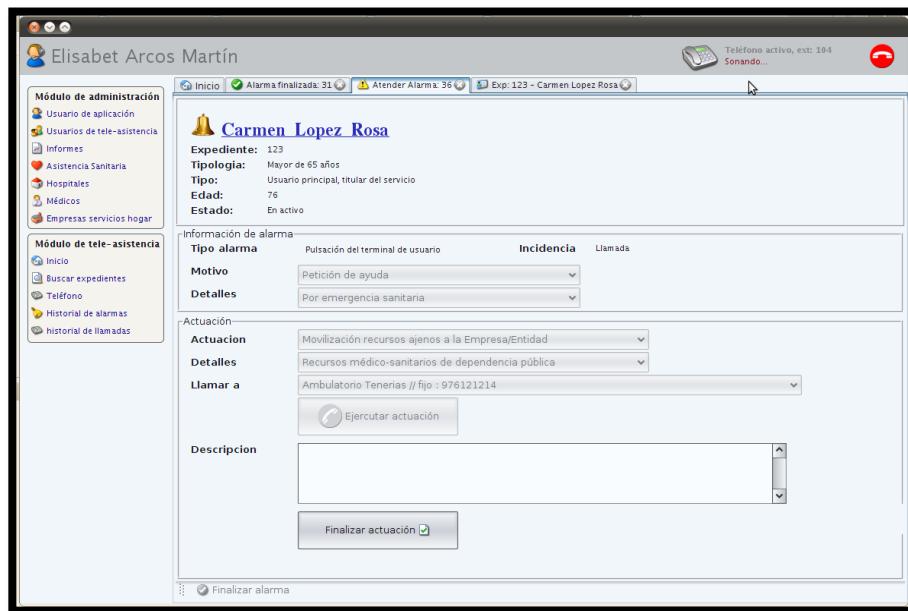
La Figura 6-11, de la sección anterior, ilustra la pestaña de inicio.

#### 6.4.3 Gestión Alarmas

La gestión de Alarmas lleva consigo tres operaciones:

1. Completar datos del tipo de alarma que se ha generado
2. Elegir la actuación pertinente
3. Inicializar la llamada al número de teléfono seleccionado de dicha actuación.

La Figura 6-13 representa la solución que presenta la aplicación AsisT para la gestión de las operaciones. En la parte superior se ofrece información del cliente del terminal que ha generado la alarma. El operador podrá consultar o ayudarse para dar un buen servicio, del expediente del cliente, accediendo directamente por un enlace en el propio nombre de la persona.



**Figura 6-13 Gestión de alarmas**

Un usuario acepta una alarma desde la ventana emergente o desde el cuadro de mandos. AsisT lee de la base de datos la información de la alarma. En caso de ser una alarma técnica, no hay llamada, y se conoce el tipo de alarma (lo aporta la centralita IP), por lo tanto solo hay que tomar actuaciones.

Si es una alarma con llamada, una vez capturada se inicia conversación de voz por medio del subsistema *softphone* y el operador selecciona los motivos por los cuales el cliente ha activado la alarma.

### **Seleccionar y ejecutar actuaciones**

La selección de actuaciones se presenta en la pantalla de gestión de alarmas y es la referida a la acción de que un operador ofrezca asistencia desde el centro para el cliente.

Para ello, AsisT le propone según el convenio-marco[5] una serie de actuaciones predefinidas, que a su vez está formadas por actuaciones más específicas.

Una vez elegida la actuación específica, AsisT le propone una serie de números de teléfono pertenecientes al cliente acorde con la actuación y con los datos disponibles del cliente.

Si procede a realizar la actuación, el sistema almacena la información de la alarma con las modificaciones: actuaciones, datos de la activación y datos de la llamada.

Si la actuación conlleva una llamada, AsisT establece la llamada sin tener que marcar el número. Una vez que libere la llamada podrá seleccionar más actuaciones.

Para terminar la atención, debe haberse indicado al menos una actuación. En caso contrario, no permite “finalizar alarma”. Si la alarma no ha tenido ninguna actuación por el centro o familiares, hay una actuación específica para ello y es necesario indicarlo para futuros informes.

Al pulsar “finalizar alarma” la pestaña se cierra para mostrar los detalles de la alarma finalizada.

### **Visualizar alarma finalizada**

Esta pestaña se inicializa tras pulsar “finalizar alarma”. Ofrece información del cliente, las capturas sufridas por la alarma y actuaciones tomadas por cada captura. En esta pantalla es donde podrá reactivar la alarma. También podrá acceder al expediente del cliente.

Uno de los problemas que tiene el software existente, es que una vez que una alarma ha sido finalizada, no se puede modificar los datos. AsisT permite reactivar la alarma. Esto facilita que pasado un tiempo, pueda ponerse en contacto con el usuario a causa de la alarma y que quede registrado. Por ejemplo: la actuación ha sido llamar a los bomberos y el usuario tras realizar la llamada, ha marcado por error que la alarma ha finalizado. En este caso reactiva la alarma para llamar al cliente y asegurarse de que ha recibido asistencia.

### **6.4.4 Gestión de la autentificación**

Este gestor contiene la interfaz y lógica para autenticar un usuario y dar acceso a la aplicación.

El usuario introduce su nombre y contraseña y la aplicación no permite observar la contraseña introducida. Los datos de usuario y contraseña son únicos en la base de datos.

Se ha utilizado el algoritmo de cifrado MD5 para almacenar la contraseña. Cuando se guarda un usuario se transforma la contraseña en una cadena de 32 caracteres que no es posible descifrar. La contraseña es correcta, si al transforma la cadena mediante MD5, es la misma que la cifrada en la base de datos.

Los datos se cifran antes de ser enviados por la red. De esta forma se garantiza su seguridad.

### **6.4.5 Gestión de los usuarios de la aplicación**

El gestor se encarga de dar de alta a o modificar los datos de los usuarios de la aplicación.

Permite asignar perfiles a un operador, de forma que tenga acceso a la funcionalidad de la aplicación según los actores definidos. Se utiliza cifrado en contraseñas desde la ficha insertar nuevo usuario.

### **6.4.6 Gestión de los clientes de la aplicación**

Es la parte encargada de la gestión de los clientes de la aplicación: dar de alta / baja y ver el listado de clientes realizando una búsqueda exhaustiva: expediente, CIU, nombre, apellidos, municipio, provincia, sexo, tipo de cliente, etc.

Cuando se da de alta a un cliente, su estado por defecto es “sin iniciar actividad”. Será necesario cumplimentar sus datos para proseguir. La primera vivienda añadida es por defecto la que tiene la instalación, ya que es el caso más común de un cliente.

Por el contrario, cuando se da de baja a un cliente, aparece la ficha de estado para forzar a que el usuario cumplimente el motivo de la baja.

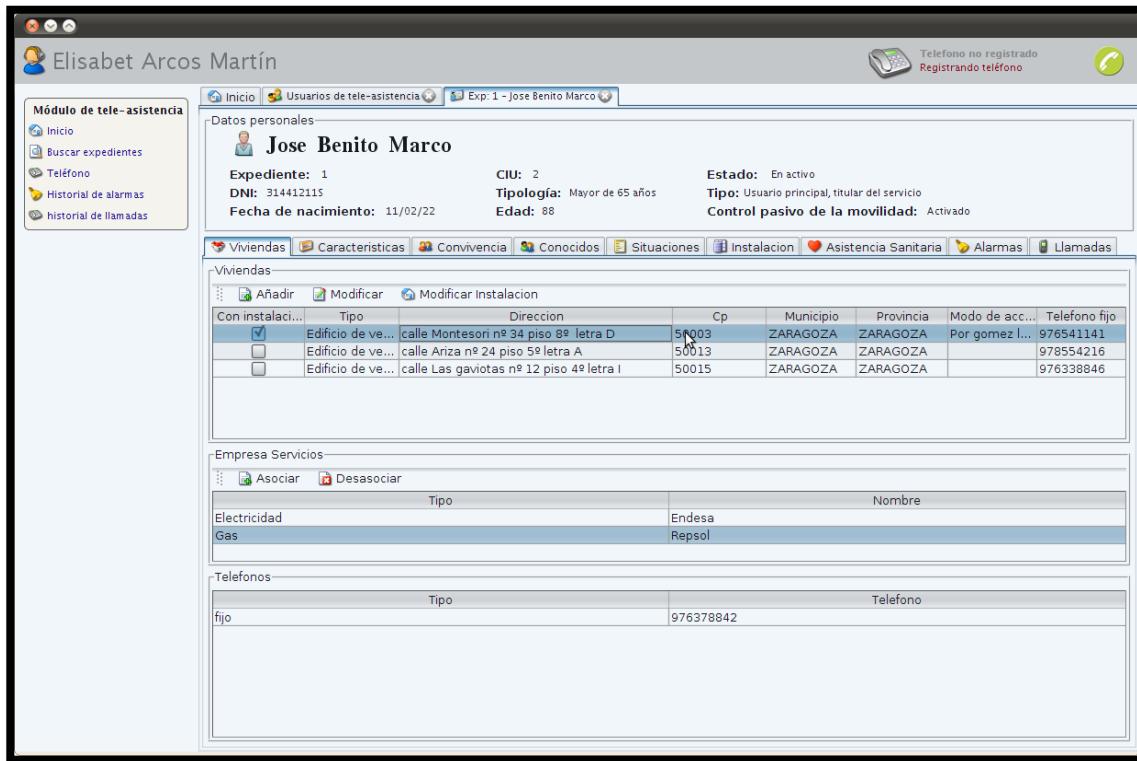
### **6.4.7 Buscar expedientes**

La pantalla buscar expedientes forma parte del módulo de tele-asistencia. Difiere de la de gestión, en los permisos: no se da de alta o modifica un cliente.

Desde esta pantalla, se accede al expediente de un cliente, siguiendo la misma funcionalidad que en la “gestión clientes”.

## 6.4.8 Gestión de un expediente de un cliente

Contiene toda la información de los clientes. Esta información está gestionada en distintos módulos con su correspondiente manipulación si es permitida. La Figura 6-14 muestra la primera pantalla del expediente. Cada pestaña, contiene la gestión de un módulo.



**Figura 6-14 Expediente cliente**

Al expediente se puede acceder desde:

- La pantalla de gestión de clientes.
- La pantalla de buscar expedientes.
- La gestión de una alarma activa.
- La pantalla de alarma finalizada.

Cada pestaña, tiene implementada la interfaz y gestión de información de los siguientes módulos:

- Módulo vivienda: gestiona las viviendas de un cliente, destacando la que tiene la instalación.
  - o Agregar/modificar/eliminar una vivienda
    - La vivienda principal será por defecto la primera vivienda insertada.
    - Control de no eliminar la vivienda que tenga la instalación principal.
  - o Modificar la vivienda donde tiene la instalación
  - o Asociar/desasociar una empresa que ofrece un servicio a la vivienda.
- Módulo características: gestiona la información personal del cliente respecto a riesgos, patologías, etc.
- Módulo personas de convivencia: gestiona la información de las personas que conviven con el cliente:
  - o Agregar/modificar/eliminar una persona con la que convive
  - o Agregar/modificar/eliminar un número de teléfono de dicha persona
- Módulo personas conocidas: gestiona la información de las personas a recurrir:
  - o Agregar/modificar/eliminar una persona a recurrir

- Agregar/modificar/eliminar un número de teléfono de dicha persona
- Módulo situaciones: gestiona el historial de los estados de la persona.
  - Permite modificar el estado de un usuario. Al dar de alta un nuevo estado, el estado anterior se finaliza en tiempo.
- Módulo instalación: gestiona la información de la instalación y equipos instalados junto con el estado.
  - Agregar/modificar/eliminar un equipo de los existentes en el mercado
  - Ver los últimos chequeos realizados al dispositivo para comprobar el estado.
- Módulo asistencia sanitaria: gestiona la información de los centros sanitarios asignados, hospitales y médicos.
  - Asociar/desasociar una vivienda de las del cliente
  - Agregar/modificar el tiempo de una vivienda a los centros de asistencia.
  - Asociar/desasociar al cliente un centro de asistencia realizando una búsqueda por zona
  - Asociar/desasociar al cliente un hospital, médico a partir de la asistencia sanitaria
  - Ver la información de los médicos y hospitales que tiene asignado por cada asistencia sanitaria.
- Módulo llamadas-cliente: muestra las últimas llamadas enviadas y recibidas para un cliente
- Módulo alarma-clientes: muestra las últimas alarmas generadas por el cliente.
  - Al realizar doble “click” sobre una fila, si la alarma está:
    - Desactivada: se abre el panel de alarma finalizada para ver las actuaciones.
    - Activada y capturada por el mismo usuario que está examinando el expediente: se abre el panel de gestión de alarma.
    - Activada y capturada por otro usuario: deniega el acceso informando que ha sido capturada por otra persona.
    - Activada y sin capturar: avisa que la alarma está pendiente de capturar. Puede aceptarla desde la esquina inferior derecha.

#### **6.4.9 Gestión Historiales**

La aplicación ofrece dos historiales diferenciados en dos módulos y accesibles desde el menú de la aplicación:

- Módulo alarmas-operador: muestra las últimas alarmas capturadas de un operador. La operativa es similar al módulo alarmas-clientes, pero en este caso, todas las alarmas están capturadas. Si se hace doble “click” sobre una fila, según el estado de la alarma:
  - Alarma desactivada: se muestra la información de la alarma finalizada.
  - Alarma activada: esta activada y siempre capturada por el operador, por lo tanto aparece la ventana de gestión de la alarma.
- Módulo llamadas-operador: muestra las últimas llamadas entrantes y salientes de un operador.

El objetivo de los historiales es servir de utilidad a la empresa de teleasistencia, en la realización de informes y al propio operador para acceder rápidamente a la información reciente. Se pretende en posteriores versiones integrar los historiales con ficheros de tipo PDF, o Excel o dispositivos como la impresora.

## 6.5 Funcionalidades adicionales de la aplicación

En este apartado se trata la implementación que no es crítica para el funcionamiento del sistema, sino que lo hacen escalable y modular, de forma que facilite el mantenimiento.

### 6.5.1 Complementos

Un complemento es una extensión de funcionalidad o característica de una aplicación que se añade a una aplicación a posterior de su creación.

Se ha implementado en AsisT un cargador de módulos en caliente. Al arrancar AsisT carga dinámicamente los módulos que se configuran en un fichero XML [49]. Esto permite crear configuraciones distintas para empresas (cargar una selección de módulos a medida) y actualizar la aplicación fácilmente, ya que sólo es necesario añadir nuevos módulos y modificar el fichero de configuración.

De esta forma, una vez obtenida la primera versión de AsisT, se puede añadir funcionalidad, garantizando el futuro y mantenimiento de la aplicación.

### 6.5.2 Traducciones

Se ha creado un fichero properties [23] que contiene todas palabras o cadenas de la aplicación identificadas por una clave. Para que la interfaz de cabida a dicha clave, se ha implementado una librería que extiende a los componentes de *Swing* de forma que todo texto sea leído del properties dado un identificador de tipo cadena.

Al arrancar AsisT, inicia por defecto el fichero de traducciones en castellano.

### 6.5.3 Soporte de distintos gestores de bases de datos

Se ha decidido por defecto utilizar el gestor de bases de datos de MySQL [9]. No obstante, si una empresa desea utilizar otro gestor de bases de datos, AsisT está diseñado para esta circunstancia.

AsisT se ha diseñado utilizando el patrón de diseño DAOFactory de forma que organiza y distribuye las clases de acceso a la base de datos. En el **¡Error! No se encuentra el origen de la referencia.** se explica la utilización del patrón.

De esta forma, el cambio a una base de datos (nótese que ésta tiene que estar soportada por Java) supondría los cambios de las clases DAO (ficheros de acceso a la base de datos), pero no de las clases de los objetos VO, ni de las clases de la interfaz y gestión de la interfaz.

### 6.5.4 Permisos de usuarios

Una empresa de teleasistencia puede tener distintos empleados con distinta funcionalidad. Por ejemplo: una empresa le interesa que los operadores, puedan modificar los datos de los clientes, mientras que otra no, puesto que dispone de otros empleados que se encargan exclusivamente de esa tarea.

Para ello se ha implementado que la base de datos junto con la configuración de la interfaz, soporten:

- Asignar permisos a usuarios:
  - o Ejemplo: super-administrador, administrador, operador.
- Que cada perfil, tenga distintos permisos:
  - o Ejemplo: modificar usuarios, dar de alta usuarios, modificar clientes, etc

De esta forma, una empresa puede gestionar los perfiles y permisos que tienen los empleados, para que sean configurados en la gestión de usuarios.

### **6.5.5 Control de errores**

Se ha implementado la gestión de errores por pantalla de forma que avise a un usuario de la introducción de datos erróneos o acciones que no puede llevar a cabo, como por ejemplo, obtener una alarma que ha capturado otro operador.

Este gestor permite guardar la información con el fin de que en un futuro y por petición del usuario, un error sea enviado a un servidor web, fichero de un servidor o un correo electrónico, para ser resuelto por un equipo de mantenimiento del software.

## 7. CONCLUSIONES

En este apartado, se detallan las conclusiones llegadas tras la realización de este proyecto. Se analizan tres aspectos:

- Cumplimiento de los objetivos
  - Futuro de AsisT
  - Valoración personal
- 

### 7.1 Cumplimiento de los objetivos iniciales

Volviendo la vista atrás, los objetivos que inicialmente se establecieron se han cumplido:

- Se ha desarrollado un sistema de gestión de alarmas en tiempo real, capaz de recibir alarmas desde una centralita IP.
- Se ha construido un sistema de información capaz de gestionar la información de los clientes de teleasistencia.
- Se ha desarrollado un sistema *softphone* capaz de establecer llamada con el exterior a través de una centralita IP.
- Se ha construido la lógica de gestión para que una empresa de alta a los usuarios de la aplicación y los clientes.
- El sistema es robusto, escalable, multiusuario y cumple la ley de protección de datos.

Además, dos aspectos importantes se han tenido en cuenta:

- En el desarrollo del sistema se ha utilizado la metodología del software de Métrica 3: aspecto que aporta consistencia al desarrollo.
- Todas las herramientas utilizadas han sido software libre, lo que permite superar las limitaciones del software privado.

Aunque los objetivos hayan sido alcanzados, se puede afirmar que el sistema no está completo. Un sistema de información de clientes de teleasistencia puede crecer exponencialmente en funcionalidad. Además se han descubierto nuevas líneas que podrían ser incorporadas. Este aspecto se detalla a continuación.

---

### 7.2 Futuro de AsisT

AsisT es un sistema innovador en el mercado. Actualmente, no existe ningún software de gestión de un centro de teleasistencia, que integre un *softphone* en la aplicación y/o que interactúe con una centralita IP.

La telefonía IP está ganando nuevos adeptos en las empresas, dejando obsoletas las centralitas analógicas por ser poco escalables y costosas.

La teleasistencia basada en una plataforma IP es el siguiente paso al que se enfrentan las empresas de teleasistencia. Este proyecto da solución a este problema. A partir de aquí, la programación aporta nueva funcionalidad para mejorar la gestión de la aplicación y la gestión de la teleasistencia.

En cuanto a la gestión de la aplicación:

- Gestión asistencia sanitaria, hospitales y médicos: crear / modificar /eliminar dichos datos (actualmente en desarrollo).

## Conclusiones

---

- Gestión de empresas de servicios del hogar: crear/modificar/eliminar empresas de servicios (actualmente en desarrollo).
- Gestión de grupos de operadores en un centro de teleasistencia (actualmente en desarrollo).

En cuanto a la gestión de teleasistencia:

- Gestión de informes: sistema completo de informes para la empresa (actualmente en desarrollo).
  - o De las sesiones de un operador en un periodo de tiempo.
  - o En formato “.pdf” y correo electrónico.
  - o Sesiones directamente a la impresora.
- Teléfono integrado en la aplicación:
  - o Llamada en espera: dejar en espera al cliente mientras se realizan las actuaciones.
  - o Grabación de llamadas entrantes y salientes.
  - o Videoconferencias.
  - o “Conversación a 3”.
  - o Control de volumen desde la aplicación.
- Configuración para la personalización de AsisT:
  - o Fijar filtros según criterios de aviso: solo reciban avisos de alarma si no están atendiendo, atender una o varias alarmas, etc.
  - o Detalles de aviso configurables con sonido.
  - o Modificación datos del cliente durante la sesión.
- Alarmas:
  - o Integrar aviso de alarmas con mensajería móvil.
- Tele-programación de los equipos desde la aplicación: ejemplo, modificar la fecha y hora, modificar el volumen, etc.
- Agenda: gestionar las tareas para un operador a beneficio de los clientes:
  - o Planificación de citas o recordatorios de los clientes.
- Teleasistencia móvil:
  - o Integrar alarmas desde un dispositivo móvil provenientes de GPS [55].

Es mucha la funcionalidad que podrían incorporarse: la programación y las herramientas existentes en la actualidad, lo hacen posible.

Lo importante, es que lo más peliagudo: los conocimientos, la plataforma, la primera versión de gestión, está realizada y que dadas las características con las que se ha realizado el sistema, puede ser escalable en funcionalidad e información.

Se espera que AsisT pueda tener una larga vida y un gran futuro, en el que espero seguir trabajando en ello.

---

## 7.3 Valoración Personal

Mi valoración del proyecto es muy positiva. La motivación inicial que me llevó aceptar el proyecto ha impulsado mi gran interés y confianza en un proyecto innovador, basado en nuevas tecnologías y con un futuro prometedor.

El proyecto ha indagado en los campos de *softphone*, cola de mensajes, programación en java, programación en PHP, servidores, redes, tecnología IP, etc. Gracias a ello, he conseguido adquirir nuevos conocimientos acerca de nuevas herramientas que desconocía y disfrutar de los logros alcanzados.

## Conclusiones

---

El hecho de realizar el proyecto en una empresa ha sido una experiencia muy gratificante. Tener el apoyo de la empresa, obtener los conocimientos de los compañeros y cometer errores y aprender de ellos, me ha ayudado para formarme en conocimientos, profesionalmente y como persona.







## ACRÓNIMOS

API	Application Programming Interface
BD	Base de Datos
CIU	Código de Identificación de Usuario
CPD	Centro de Proceso de Datos
CTI	Computer Telephony Integration
DAO	Data Access Object
DFD	Diagrama de Flujo de Datos
DTMF	Dual-Tone Multi-Frecuency
FEMP	Federación Española de Municipios y Provincias
IMSERSO	Instituto de Mayores y Servicios Sociales
GPS	Global Position System
IP	Internet Protocol
JDBC	Java Database Connectivity
JMS	Java Messaging Service
PBX	Private Branch Exchange
PDF	Portable Document Format
RTB	Red de Telefonía Básica
SAAS	Sotfware As A Service
SDP	Session Description Protocol
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
UDP	User Data Protocol
UCR	Unidad de Control Remoto
URI	Uniform Resource Identifier
VO	Value Object
XML	Extensible Markup Language

# BIBLIOGRAFÍA

Referencia	Título
[1]	JAIN <a href="http://java.sun.com/products/jain/">http://java.sun.com/products/jain/</a>
[2]	SIP: RFC 3261 <a href="http://www.ietf.org/rfc/rfc3261.txt">http://www.ietf.org/rfc/rfc3261.txt</a>
[3]	IMSERSO <a href="http://www.imserso.es/imserso_01/index.htm">http://www.imserso.es/imserso_01/index.htm</a>
[4]	FEMP <a href="http://www.femp.es/">http://www.femp.es/</a>
[5]	Normativa de teleasistencia aportada por IMSERSO y FEMP: <a href="http://www.imsersomayores.csic.es/documentos/documentos/imserso-programatelesistencia-01.pdf">http://www.imsersomayores.csic.es/documentos/documentos/imserso-programatelesistencia-01.pdf</a>
[6]	Java : <a href="http://www.java.com/en/">http://www.java.com/en/</a>
[7]	Comunidad Open Source: <a href="http://www.opensource.org/">http://www.opensource.org/</a>
[8]	Sistema operativo Linux Ubuntu: <a href="http://www.ubuntu-es.org/">http://www.ubuntu-es.org/</a>
[9]	MySQL: <a href="http://www.mysql.com/">http://www.mysql.com/</a>
[10]	Postree SQL <a href="http://www.postgresql.org/">http://www.postgresql.org/</a>
[11]	Oracle: Gestor de bases de datos <a href="http://www.oracle.com/global/es/index.html">http://www.oracle.com/global/es/index.html</a>
[12]	Ley jurídica de protección de datos: <a href="http://noticias.juridicas.com/base_datos/Admin/lo15-1999.html">http://noticias.juridicas.com/base_datos/Admin/lo15-1999.html</a>
[13]	NetBeans: Herramienta de programación <a href="http://netbeans.org/">http://netbeans.org/</a>
[14]	Aptana: Herramienta de programación utilizada para PHP <a href="http://www.aptana.org/">http://www.aptana.org/</a>
[15]	JDBC Driver para MySQL <a href="http://dev.mysql.com/downloads/connector/j/3.0.html">http://dev.mysql.com/downloads/connector/j/3.0.html</a>
[16]	Substance: <a href="https://substance.dev.java.net/">https://substance.dev.java.net/</a>
[17]	Herramienta de debugging para java: <a href="http://logging.apache.org/log4j/1.2/download.html">http://logging.apache.org/log4j/1.2/download.html</a>
[18]	Librería JMS para paso de mensajes: <a href="http://java.sun.com/products/jms/docs.html">http://java.sun.com/products/jms/docs.html</a> Java message service / Richard Monson-Haefel, David A. Chappell
[19]	Librería JUnit pruebas del sistema: <a href="http://www.junit.org/">http://www.junit.org/</a>
[20]	Design Patterns: Elements of Reusable Object-Oriented Software Escrito por Gamma
[21]	Fundamentos de voz sobre IP / Jonathan Davidson, James Peters
[22]	AsterisK: Software de gestión de la centralita IP <a href="http://www.asterisk.org/">http://www.asterisk.org/</a>
[23]	Properties clasas: <a href="http://java.sun.com/j2se/1.5.0/docs/api/">http://java.sun.com/j2se/1.5.0/docs/api/</a>
[24]	Cola de mensajería Active MQ: <a href="http://activemq.apache.org/">http://activemq.apache.org/</a>
[25]	Cola de mensajería OpenJMS <a href="http://openjms.sourceforge.net/">http://openjms.sourceforge.net/</a>
[26]	Cola de mensajería Rabbit MQ <a href="http://www.rabbitmq.com/">http://www.rabbitmq.com/</a>
[27]	Gestor de almacenamiento InnoDB <a href="http://dev.mysql.com/doc/refman/5.0/en/innodb.html">http://dev.mysql.com/doc/refman/5.0/en/innodb.html</a>
[28]	Análisis y diseño orientado a objetos de sistemas usando UML / Simon Bennet, Steve McRobb, Ray Farmer
[29]	Object-oriented analysis / Peter Coad and Edward Yourdon
[30]	Métrica 3:Metodología de Software <a href="http://www.csi.map.es/csi/metrica3/index.html">http://www.csi.map.es/csi/metrica3/index.html</a>
[31]	Pencil v1.0: herramienta de modelado <a href="http://www.evolus.vn/pencil/Downloads.html">http://www.evolus.vn/pencil/Downloads.html</a>
[32]	PHP <a href="http://php.net/index.php">http://php.net/index.php</a>

## Bibliografía

---

Referencia	Título
[33]	DIA : Herramienta de diagramas <a href="http://live.gnome.org/Dia">http://live.gnome.org/Dia</a>
[34]	GIMP: Herramienta de edición de imágenes <a href="http://www.gimp.org/">http://www.gimp.org/</a>
[35]	Umbrello: Herramienta de modelado UML <a href="http://uml.sf.net">http://uml.sf.net,</a>
[36]	OpenOffice: Procesador de texto y diagramas <a href="http://es.openoffice.org/">http://es.openoffice.org/</a>
[37]	Subversion <a href="http://subversion.tigris.org/">http://subversion.tigris.org/</a>
[38]	Label: Clase Label de Java <a href="http://java.sun.com/j2se/1.4.2/docs/api/java.awt/Label.html">http://java.sun.com/j2se/1.4.2/docs/api/java.awt/Label.html</a>
[39]	Clase JOptionPane <a href="http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/JOptionPane.html">http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/JOptionPane.html</a>
[40]	Event Listener <a href="http://java.sun.com/docs/books/tutorial/uiswing/events/index.html">http://java.sun.com/docs/books/tutorial/uiswing/events/index.html</a>
[41]	Timer Task <a href="http://java.sun.com/j2se/1.4.2/docs/api/java/util/TimerTask.html">http://java.sun.com/j2se/1.4.2/docs/api/java/util/TimerTask.html</a>
[42]	Mozilla Firefox: navegador web <a href="http://www.mozilla.com/en-US/">http://www.mozilla.com/en-US/</a>
[43]	ActiveMQ Web console <a href="http://activemq.apache.org/web-console.html">http://activemq.apache.org/web-console.html</a>
[44]	MySQL Query Browser <a href="http://dev.mysql.com/doc/query-browser/en/index.html">http://dev.mysql.com/doc/query-browser/en/index.html</a>
[45]	Empresa Verklizan: Servidor UMO <a href="http://www.verklizan.org/exec/verklizanweb.exe?lang=SP&amp;page=Pagina/UMO52.html">http://www.verklizan.org/exec/verklizanweb.exe?lang=SP&amp;page=Pagina/UMO52.html</a>
[46]	Empresa Sabia: <a href="http://bioingenieria.es/Teleasistencia/Teleasistencia.htm">http://bioingenieria.es/Teleasistencia/Teleasistencia.htm</a>
[47]	.Net <a href="http://www.microsoft.com/net/">http://www.microsoft.com/net/</a>
[48]	Select for update <a href="http://dev.mysql.com/doc/refman/5.0/en/innodb-locking-reads.html">http://dev.mysql.com/doc/refman/5.0/en/innodb-locking-reads.html</a>
[49]	XML <a href="http://www.w3.org/XML/">http://www.w3.org/XML/</a>
[50]	Swing <a href="http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/package-summary.html">http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/package-summary.html</a>
[51]	Aplicación Flash <a href="http://www.adobe.com/es/">http://www.adobe.com/es/</a>
[52]	Applet Java <a href="http://java.sun.com/applets/">http://java.sun.com/applets/</a>
[53]	Active X <a href="http://msdn.microsoft.com/en-us/library/ms537508.aspx">http://msdn.microsoft.com/en-us/library/ms537508.aspx</a>
[54]	AGI <a href="http://www.voip-info.org/wiki/view/Asterisk+AGI">http://www.voip-info.org/wiki/view/Asterisk+AGI</a>
[55]	GPS <a href="http://www.gps.gov/">http://www.gps.gov/</a>
[56]	API JCalendar <a href="https://jcalendar.dev.java.net/">https://jcalendar.dev.java.net/</a>
[57]	Wikipedia <a href="http://www.wikipedia.org">http://www.wikipedia.org</a>

## GLOSARIO

**Teleasistencia domiciliaria:** la teleasistencia domiciliaria es un servicio que permite a las personas mayores y/o personas discapacitadas, recibir asistencia en un centro, atendido por personal específicamente preparado para dar respuesta adecuada a la necesidad. El servicio es prestado a través de la línea telefónica y con un equipamiento de comunicaciones e informático específico, ubicado en un centro de atención y en el domicilio de los usuarios. Los clientes con sólo accionar el dispositivo que llevan constantemente puesto o el terminal de su hogar, entran en contacto verbal, "manos libres", durante las 24 horas del día y los 365 días del año. La empresa que ofrece el servicio presta atención, bien movilizando los recursos propios que dispone o movilizando otros recursos humanos o materiales, propios del usuario o existentes en la comunidad [5].

**Centro de atención:** también llamado centro de operadores o "call-center" es el lugar en el que los empleados de una empresa de teleasistencia dan servicio a los clientes [5].

**Clientes:** son personas mayores o discapacitadas, que contratan el servicio de teleasistencia a una empresa [5].

**El titular del servicio de teleasistencia domiciliaria:** persona que reúne todos los requisitos para ser cliente y el terminal de la instalación pertenece a su nombre [5].

**Servicio de teleasistencia domiciliaria sin unidad móvil:** el servicio básico de teleasistencia domiciliaria se presta exclusivamente desde el centro de atención, tratando las distintas situaciones que se presentan [5].

**Servicio de teleasistencia domiciliaria con unidad móvil:** equipo propio de la empresa de teleasistencia, encargado de incrementar la eficacia del servicio, por medio de una atención presencial en los domicilios de los usuarios. Esta actividad, será realizada por el personal de las unidades móviles [5].

**Entidad local:** una entidad local es una zona que puede ser definida como: municipio, código postal, barrio, distrito, etc. que solicita como miembro público del estado, realizar las tareas de empresa de teleasistencia, bajo su jurisdicción [5].

**Usuarios - operadores:** son los encargados de recibir, en primera instancia, las alarmas y llamadas, interviniendo según instrucciones y protocolos establecidos. Pueden solicitar la intervención directa del responsable del centro cuando por su complejidad no puedan o no sepan resolver la situación por sí mismos.

**Terminales de usuario:** emisores de alarmas con sus correspondientes unidades de control remoto (UCR). Dispondrá de un botón bien diferenciado del resto, si es que los hubiese, cuya sola pulsación permita la activación del sistema y puesta en contacto, en modo conversación "manos libres" para el usuario, con el centro receptor. Integrará un altavoz y un micrófono controlables, en caso de alarma, desde la central receptora. Será capaz de comunicarse con el centro receptor utilizando más de un protocolo de comunicaciones [5].

**La unidad de control remoto (UCR):** dispositivo asociado al terminal de usuario, el cual dispone de un botón cuya sola pulsación desencadene la activación del sistema y puesta en contacto, en modo conversación "manos libres" para el usuario, con el centro receptor [5].

**Sistema de atención de alarmas:** sistema destinado a la recepción y gestión de las llamadas de alarma recibidas por el centro receptor de alarmas [5].

**Discapacitado:** persona con minusvalías permanentes o temporales, así como a quienes padeczan cualquier tipo de limitación física, psíquica o sensorial [5].

**Control pasivo de la movilidad:** tipo de servicio disponible sólo para usuarios de alto riesgo. Son todos aquellos que requieran de un seguimiento más intenso [5].

**Entidad de asistencia sanitaria:** establecimiento médico dependiente del sistema de sanidad pública o privada, en el que se presta asistencia médica y farmacéutica a pacientes sin ingresarlos en él [5].

**Estados de un usuario:** dependiendo de su situación respecto al proyecto, pueden generarse las siguientes situaciones:

- En activo: recibe actualmente el servicio.
- Ausencia domiciliaria inferior a 24 horas: estado en activo. El usuario ha salido del domicilio y lo comunica.
- Suspensión temporal: habiendo estado en activo, deja provisionalmente de recibir el servicio por periodo superior a 24 horas.
- Baja definitiva: habiendo estado en activo, causa baja definitivamente en el servicio.
- Sin iniciar actividades: servicio solicitado y asignado, aún no ha causado alta.
- Baja definitiva de solicitantes que nunca estuvieron en activo [5].

**Solicitud:** es la previa autorización de los servicios sociales, familiares o del propio usuario, explicando de forma clara las consecuencias de su utilización. Cada usuario sólo podrá presentar una solicitud [5].

**Empresa de servicios:** unidad de organización dedicada a actividades del hogar como Gas, Electricidad, etc [5].

**Empresa de Terminales:** unidad de organización dedicada a la fabricación de terminales de teleasistencia [5].

**Recursos de una entidad:** son servicios de urgencia propios de una entidad [5].

**Organismos:** son servicios públicos que ofrecen servicios de urgencia, como bomberos, ambulancias, etc [5].



