



**SGCE**

## **SISTEMA DE GESTIÓ I CONTROL D'EQUIPS**

Memoria del Proyecto Fin de  
Carrera  
de Ingeniería en Informática  
realizado por  
**Alberto Bermejo Mena**  
y dirigido por  
**Javier Serrano García**

Bellaterra, 1 de juny de 2007



**En memoria de mi abuelo.**

**A Cristina por su apoyo y paciencia.  
A Moa por su cariño.**

**A toda mi familia.**

**A Javier por asignarme este proyecto  
y guiarme en su desarrollo cumpliendo  
los objetivos que se habían propuesto.**

# Índice

<b>Capítulo 1 Introducción</b>	<b>1</b>
1.1 Presentación	1
1.2 ¿Por qué realizar este proyecto?	2
1.3 Motivación personal	2
1.4 Objetivos generales	3
1.5 Estructura de la memoria	3
1.6 Planificación del proyecto	4
<b>Capítulo 2 El CatSalut y Tecnologías aplicadas</b>	<b>6</b>
2.1 Estructura organizativa del CatSalut	6
2.1.1 Organización del SAPLH	7
2.1.2 Infraestructura del SAPLH	9
2.1.3 Informatización de los centros	9
2.2 Tecnologías aplicadas	13
2.2.1 Visual Studio 2005	13
2.2.2 Microsoft .NET Framework	13
2.2.2.1 ASP.NET	16
2.2.2.2 ADO.NET	17
<b>Capítulo 3 Análisis de especificaciones</b>	<b>18</b>
3.1 Especificaciones funcionales	18
3.1.1 Especificaciones de los usuarios del sistema	18
3.1.2 Especificaciones de la seguridad en el sistema	19
3.1.3 Detalles de las funcionalidades	19
3.2 Especificaciones no funcionales	22

**Capítulo 4 Diseño** **23**

4.1 Metodología de desarrollo	23
4.2 Diseño funcional	25
4.2.1 Diagramas de flujo	25
4.2.2 Detalle de los usuarios del sistema	32
4.2.2.1: Detalle basado en el rol	32
4.2.2.2: Detalle de seguridad	33
4.3 La base de datos	35
4.3.1 Diagrama entidad-relación	35
4.3.2 Diseño de las tablas	36
4.4. Diseño de la Intefície Gráfica de Usuario	51
4.4.1 Características de la aplicación Web	51
4.4.2 Maqueta de las GUI	52
4.4.3 Aplicación de las características a las páginas Web	53

**Capítulo 5 Implementación y prueba** **58**

5.1 Implementación de la aplicación	58
5.1.1 Estructura del sitio Web	58
5.1.2 Implementación de las clases de la lógica de negocio	59
5.1.3 La implementación de la clase de la capa de acceso a datos	68
5.2 Prueba	74

**Capítulo 6 Manual de usuario** **77**

6.1 Introducción	77
6.2 Entrada a la aplicación	78
6.3 Localizar un equipo	80
6.4 Gestión de una incidencia	86
6.5 Visualización de datos	89
6.6 Administración del centro	91
6.7 Cambio de Àmbito, SAP o Centro	92

**Capítulo 7 Conclusiones** **94**

7.1 Conclusiones	94
7.2 Líneas de futuro	95

**Bibliografía** **96**



# Capítulo 1

## INTRODUCCIÓN

### 1.1 Presentación.

La Sociedad de la Información establece continuamente nuevos métodos de trabajo y cambios en los ya existentes. Estas nuevas formas de trabajo vienen marcadas por cambios culturales, tecnológicos, socio-económicos,... que se verifican en la sociedad. Nos encontramos inmersos dentro del fenómeno de la globalización en el que todos los sectores atraviesan transformaciones en sus estructuras al compás de estos cambios.

El CatSalut, dentro del sector sanitario, quiere ir más allá de la simple atención sanitaria, para ello promueve la participación de los profesionales, dota de instrumentos organizativos que potencien la calidad, la profesionalidad y la responsabilidad en el trabajo con el objetivo de garantizar una atención sanitaria integral y de calidad para todos los ciudadanos y ciudadanas de Cataluña.

Durante el año 2002 se inició el Plan de sistemas de información que consistía en definir el conjunto de instrumentos organizativos y tecnológicos necesarios para conseguir los objetivos estratégicos del CatSalut, algunos de los objetivos estratégicos que se iniciaron fueron el despliegue de una nueva red de telecomunicaciones, la creación de la estación de trabajo clínico de atención primaria (e-CAP) y de atención hospitalaria (e-CHOS), la creación de la plataforma de servicios básicos para los profesionales del ICS,...

A mediados de 2004 el departamento de Sistemas de Información del Servicio de Atención Primaria de l'Hospitalet (SAPLH en adelante) empezó a implantar el proyecto

e-CAP, dotar a los centros de la estación clínica de trabajo de atención primaria, además también se llevó a cabo una renovación del equipo informático obsoleto conforme al nuevo proyecto.

## 1.2 ¿Por qué realizar este proyecto?

Inicialmente el SAPLH tenía informatizado los puestos de trabajo del personal de la unidad de atención al usuario (UAAU, en adelante) y las consultas médicas de algunas áreas básicas de salud de los centros de atención primaria. Este número de equipos informáticos, representativo al principio, vería incrementado su número debido a la creación de nuevos puestos informatizados. Además, en cada centro se consolidó la estación clínica de trabajo, un puesto trabajo informatizado por cada consulta médica (las consultas de medicina, pediatría, enfermería, asistente social, psicología, psiquiatría,...). Algunos de esos centros, además de ofrecer asistencia médica también ofrecían otros servicios como el servicio de Rehabilitación, el de Urgencias, el de Radiología,...

El proyecto e-CAP ha incrementado notablemente el número de equipos informáticos existente y ha acentuado la necesidad de gestión. De esta necesidad surge el *Sistema de Gestió y Control d'Equips*, SGCE. SGCE deberá proporcionar al departamento de Sistemas de Información la solución para la gestión del equipo informático del SAPLH.

Mi incorporación al departamento de Sistemas de Información era reciente cuando se inició el proyecto, apenas unos meses. A medida que se iban informatizando centros la idea de realizar el sistema iba, a la vez, cogiendo consistencia.

## 1.3 Motivación personal.

Cuando decidí a llevar a cabo este proyecto realizaba funciones de administración de equipos y asistenciales a los usuarios del SAPLH. Por las funciones que desarrollaba en del departamento no se contempló la idea de realizarlo como proyecto externo a la Universidad.

A pesar de no realizar el proyecto como externo, existía la posibilidad de implantarlo en el departamento y decidí que este podría ser mi proyecto de final de carrera. Por mis labores en el departamento dispondría de toda la información necesaria para llevarlo a cabo, tendría acceso a la información de los centros, a la ubicación de los puestos de trabajo, a la información del equipo informático (modelos, fabricantes, características hardware, software instalado), como se gestionan las averías, qué información es relevante y necesaria para el usuario de un centro,...

Por otro lado este proyecto se presentaba como un reto ya que debía asumir el perfil de analista, programador y “*project manager*” a la vez. Al ejercer estas tres funciones me permitirían elegir detalles del diseño, de la implementación, de las tecnologías a utilizar, marcar el “*planning*” de desarrollo,...

## 1.4 Objetivos generales.

Las ventajas que se obtendrían de la utilización del sistema serían las siguientes:

- Desarrollar una herramienta única para llevar a cabo tanto la administración de los equipos como la gestión de las incidencias.
- Orientado al personal administrativo del centro (CAP o ABS).
- Supervisado por los informáticos de los SAP.
- Visualización de la información de un equipo informático.
- Control y mantenimiento del equipo informático del que se dispone.
- Facilidad para gestionar una incidencia de un equipo desde su inicio hasta su conclusión.

## 1.5 Estructura de la memoria.

La memoria se ha estructurado de la siguiente forma: se ha presentado una primera parte introductoria donde se explican los objetivos y las motivaciones de este proyecto, a continuación, se presenta un capítulo centrado en el Catsalut, su estructura organizativa y dónde se ubica el departamento de sistemas de información, asimismo se detallan las tecnologías aplicadas para la generación de la solución. Seguidamente se detallan los requisitos mediante el análisis de especificaciones de la aplicación. Los capítulos siguientes están centrados en el diseño y la implementación de la solución. A continuación se presenta el manual de usuario detallando los pasos a seguir de cada funcionalidad implementada. Como parte final, se exponen las conclusiones extraídas del trabajo realizado y las líneas futuras del sistema.

## 1.6 Planificación del proyecto.

El proyecto se dividió en tres fases:

- Fase 1: Análisis de la organización, en ella se estudiaría la infraestructura informática de la organización:
  - Hardware y Software.
  - Sistemas de Información.
  - Evolución de la organización.
- Fase 2: Análisis de requerimientos y de elección de las herramientas de desarrollo.

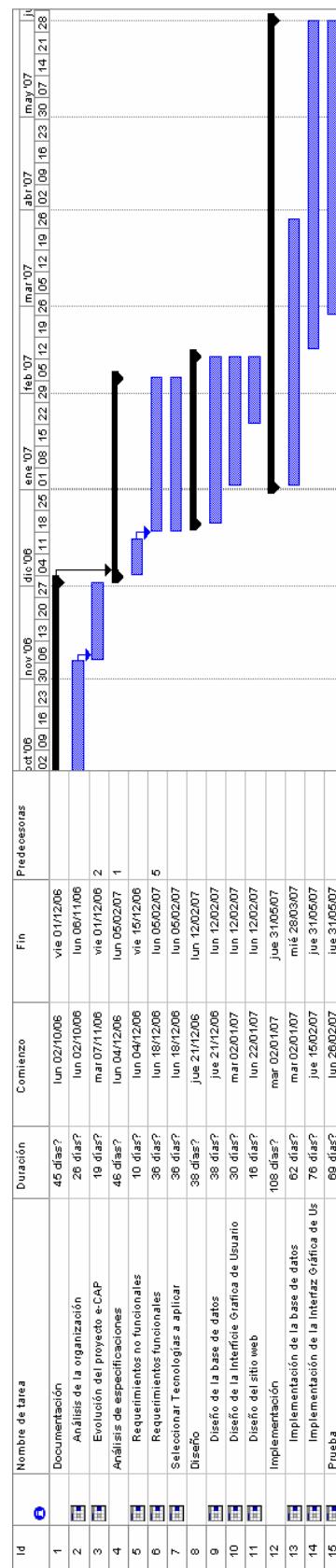
Esta fase ha servido para:

- Detallar el alcance del proyecto. Definir la funcionalidad de la aplicación.
  - Evaluar las herramientas y software para el desarrollo.
  - Seleccionar las herramientas.
- 
- Fase 3: De diseño e implementación.

Está formada por cinco etapas:

- Diseño de la base de datos
- Diseño de la Interfície Gráfica de Usuario
- Diseño del sitio web
- Implementación del sistema
- Prueba de forma individual y conjunta de los módulos integrantes.

En la siguiente figura se presenta el cronograma del proyecto, en él se detallan las fases de las que consta y el tiempo dedicado a cada fase respectivamente. Se estableció su inicio el 2 de octubre de 2006 y su finalización para el 31 de mayo de 2007, un total de 237 días de los cuales, 163 laborables, con una dedicación de dos horas diarias al desarrollo del proyecto dan como resultado un total de 326 horas.



## Capítulo 2

# EL CATSALUT Y TECNOLOGÍAS APLICADAS

En este capítulo se presenta en primer lugar la disposición organizativa del CatSalut, dónde se ubica el SAPLH dentro de este contexto y a continuación se detalla la evolución del proceso de implantación del proyecto e-CAP.

Una vez vista la situación organizativa, se pasará a evaluar las herramientas, el software para el desarrollo y las tecnologías que se aplicarán posteriormente en la fase de implementación del sistema.

### 2.1 Estructura organizativa del CatSalut

El CatSalut es el mayor proveedor de servicios sanitarios de Catalunya. Está adscrito a Departament de Salut de la Generalitat de Catalunya y su función es la de garantizar la atención sanitaria de calidad para todos los ciudadanos y ciudadanas de Catalunya adecuando los servicios sanitarios en función de las necesidades de la población y evaluando la satisfacción ciudadana de los servicios prestados.

Para poder realizar su misión efectúa un despliegue territorial que permite un mejor conocimiento de las necesidades de salud de la población y una mejor aproximación a los proveedores de servicios en cada una de las regiones sanitarias de las que se compone.

Catalunya está dividida en diferentes regiones sanitarias (ver figura 2.1). Cada región sanitaria cuenta con una dotación adecuada de recursos sanitarios de atención

primaria y de atención especializada para atender las necesidades de la población; cada región se divide, a su vez, en sectores sanitarios.

Los sectores sanitarios son los ámbitos donde se desarrollan y se coordinan las diferentes actividades de salud pública y la asistencia socio-sanitaria a nivel de atención primaria y de las especialidades médicas.

Cada ámbito consta a su vez de uno o más servicios de atención primaria (SAP). Estos SAP están formados por la agrupación de áreas básicas de salud.

Las áreas básicas de salud (ABS) se definen como la unidad territorial elemental. A través de ellas se organizan los servicios de atención primaria de salud. Si estas unidades se encuentran en áreas urbanas pueden estar formadas por barrios o distritos y por uno o varios municipios en ámbito rural.



**Figura 2.1: Regiones sanitarias de Cataluña.**

### 2.1.1 Organización del SAPLH

En el marco definido anteriormente se encuentra el Servicio de Atención Primaria de l'Hospitalet del Llobregat (SAPLH) que cubre la población de l'Hospitalet del Llobregat. El SAPLH pertenece a la región sanitaria de Barcelona y depende del Àmbit Costa de Ponent.

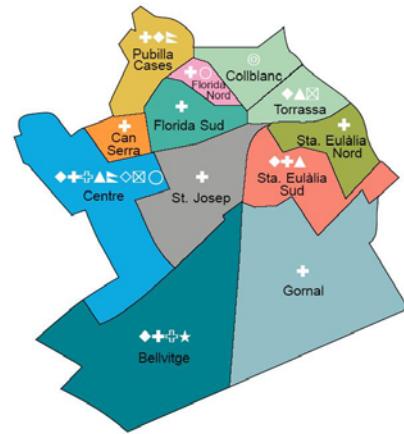
El SAPLH está formado por los siguientes centros de atención primaria:

- CAP Bellvitge
- CAP Bòbila
- CAP Gornal
- CAP Florida
- CAP Rambla Just Oliveras
- CAP Sanfeliu
- CAP Amadeu Torner

- CAP Mossèn Cinto Verdaguer
- CAP Rambla Marina
- CAP Can Serra

Estos centros albergan las siguientes áreas básicas de salud:

- ABS Bellvitge
- ABS Pubilla Casas
- ABS Gornal
- ABS Florida Nord
- ABS Florida Sud
- ABS Centre
- ABS SanJosep
- ABS SanFeliu
- ABS Santa Eulalia Sud
- ABS Santa Eulalia Nord
- ABS Rambla Marina
- ABS Can Serra



**Figura 2.2: ABS y CAPs del SAPLH.**

En la figura 2.2 aparece Collblanch y Torrassa, ambas áreas básicas pertenecen al Consorci Sanitari Integral (CSI), un proveedor de servicios sanitarios del CatSalut.

El SAPLH está formado además por servicios centrales en el ámbito asistencial, ámbito técnico y ámbito administrativo.

En el ámbito técnico tenemos:

- **Informática:** administra los sistemas centrales y las redes informáticas, instala y configura el software y el hardware, realiza la atención y soporte a los usuarios así como el mantenimiento de los equipos.
- **Mantenimiento:** realiza el mantenimiento preventivo, el seguimiento y la reparación de las averías en los edificios. Interviene en las nuevas instalaciones, modificaciones y reformas en general.

En el ámbito administrativo:

- **Administración y logística:** administra las finanzas del SAPLH, los gastos, presupuestos, facturas,... Y la parte de logística se encarga de las compras, la recogida, el almacenamiento y la distribución del material.

- **RRHH:** Tramita el alta y la baja de los profesionales, gestiona y controla los contratos, los permisos,...

En el ámbito asistencial:

- **Unidad de atención al usuario:** Tramita y gestiona documentación de los usuarios para proveedores de salud.
- **Unidad de Farmacia:** Almacena, controla y prepara la dosificación de medicamentos, vacunas,...
- **Unidad de Formación y Procesos de Enfermería:** Se encarga de perfilar un calendario anual de formación continua para el personal del SAPLH y de todo lo relacionado con Enfermería.
- **Unidad Asistencial:** Se encarga de proporcionar atención a los mismos usuarios del SAPLH en temas relacionados con programas específicos de fabricantes externos e información administrativa-socio-sanitaria.

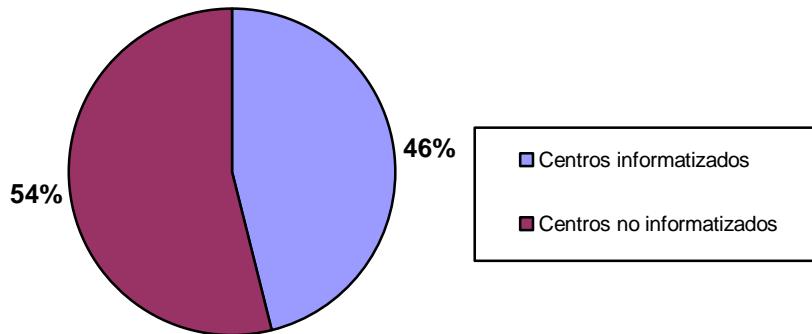
### **2.1.2 Infraestructura del SAPLH**

El SAPLH está formado por diez centros de atención primaria donde se atiende a la población de las diferentes áreas básicas de salud. No todos los centros disponen del mismo número de consultas médicas, servicios asistenciales, servicios especializados,... Sin embargo, la infraestructura de la que dispone cada centro es similar, es decir, cada centro tiene su propia red local con su rango de direcciones IP propio y están unidos al Nus Sanitari, una red de telecomunicaciones creada para unir los centros del CatSalut.

### **2.1.3 Informatización de los centros.**

El SAPLH fue pionero en Catalunya al introducir en la consulta un equipo informático, un primer esbozo de la futura estación clínica de trabajo de atención primaria y utilizar la historia clínica informatizada como soporte para pasar consulta a los pacientes, dejando a un lado la tradicional historia de papel.

De los trece centros pertenecientes al SAPLH, seis se informatizaron, esta cifra representa el 46%. Este porcentaje indica una fuerte apuesta por parte del SAPLH por las nuevas tecnologías de la información.



**Figura 2.3: Gráfico que muestra el porcentaje de centros informatizados.**

La estación clínica se define como el equipo informático de una consulta, estaba formado por un PC y una impresora de inyección de tinta, y requería conexión a la intranet del SAPLH para poder acceder a la información de cada paciente que se encontraba centralizada en la unidad de Sistemas de Información.

Debido a la ubicación de los centros informatizados se mejoró la infraestructura de telecomunicaciones existente.

El programa que utilizaba el personal sanitario en las consultas era el SIAP-WIN. Esta aplicación ofrecía a los médicos y enfermeras las funcionalidades necesarias para hacer un seguimiento de la salud de los pacientes como la prescripción de medicamentos, el control de factores de riesgo, seguimiento clínico,... Durante los años que esta aplicación se mantuvo fue revisada y renovada, añadiéndole nuevas funcionalidades requeridas por el personal sanitario.

Dentro de la misma región sanitaria existía otro programa con similares funcionalidades, el OMIAP.

El CatSalut no tenía claro qué aplicación instalar en todo el territorio, pero las experiencias y los conocimientos que extraía de los diferentes profesionales que trabajaban con estas herramientas demostraron que la línea a seguir era aplicar las nuevas tecnologías al servicio de todos los profesionales de la sanidad y no únicamente centrarla en la consulta médica.

En su Plan de Sistemas de Información, el CatSalut asentó las bases de la estación clínica de trabajo de atención primaria como fruto de los conocimientos y experiencias adquiridas durante los años de trabajo con las aplicaciones SIAP-WIN y OMIAP.

Además de informatizar las consultas de atención primaria, el CatSalut también contemplaba la informatización de los diferentes servicios asistenciales que complementan la medicina familiar, algunos de estos servicios son el de Urgencias, el de Odontología, de Salut Mental,... de manera que todos los profesionales del CatSalut utilizaran e-CAP como herramienta de trabajo. Por razones que quedan fuera de este proyecto no se comentará por qué se implantó e-CAP, en el CatSalut.

En el 2004 el SAPLH empezó la migración de los centros con SIAP-WIN hacia e-CAP y la informatización del resto de centros. La migración no fue sencilla, en los centros informatizados con SIAP-WIN se planificó un calendario, se coordinaron equipos de migración de base de datos, de infraestructura de telecomunicaciones,... En los centros de nueva informatización, se debió realizar un trabajo previo de adecuación del edificio a la nueva infraestructura de telecomunicaciones, así como dotar de un sistema eléctrico apropiado y propio para el equipo informático que se iba a instalar.

El SAPLH se planteó dos objetivos bien definidos, el primero consistía en informatizar el servicio de atención primaria de todos los centros y el segundo, la implantación de e-CAP como estación clínica de trabajo de atención primaria siguiendo las directrices del CatSalut.

En la figura 2.4 vemos de forma cronológica cómo se realizó la implantación de la aplicación e-CAP en el SAPLH. Podemos observar como a partir de julio del 2005 ya se había conseguido el primer objetivo y un año más tarde se completó el segundo.

Siguiendo las directrices del CatSalut, una vez informatizada la atención primaria con la aplicación e-CAP, el SAPLH inició a partir de julio del 2006 la fase de informatización de los servicios de atención especializada.

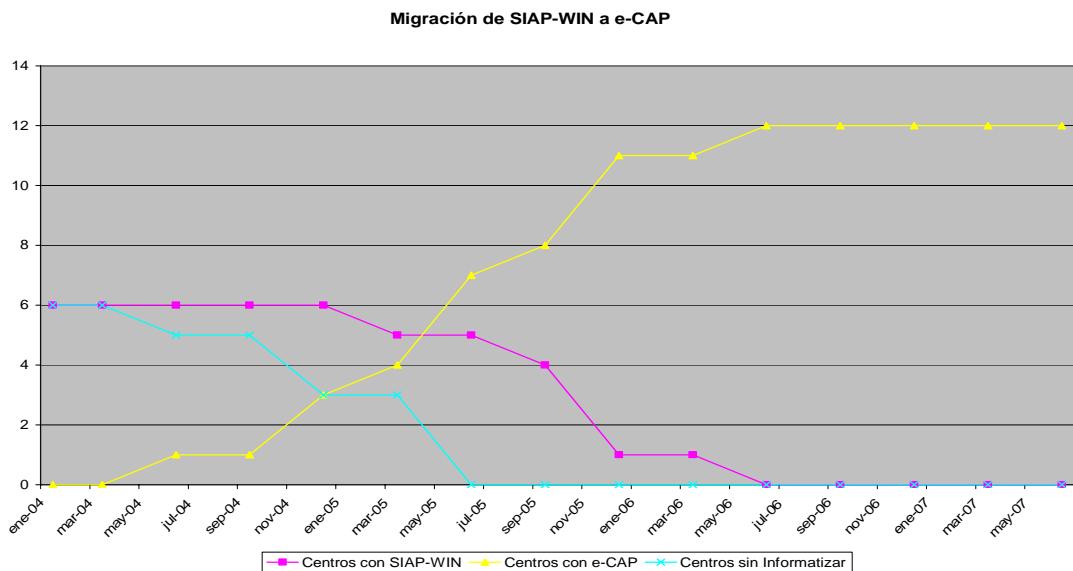
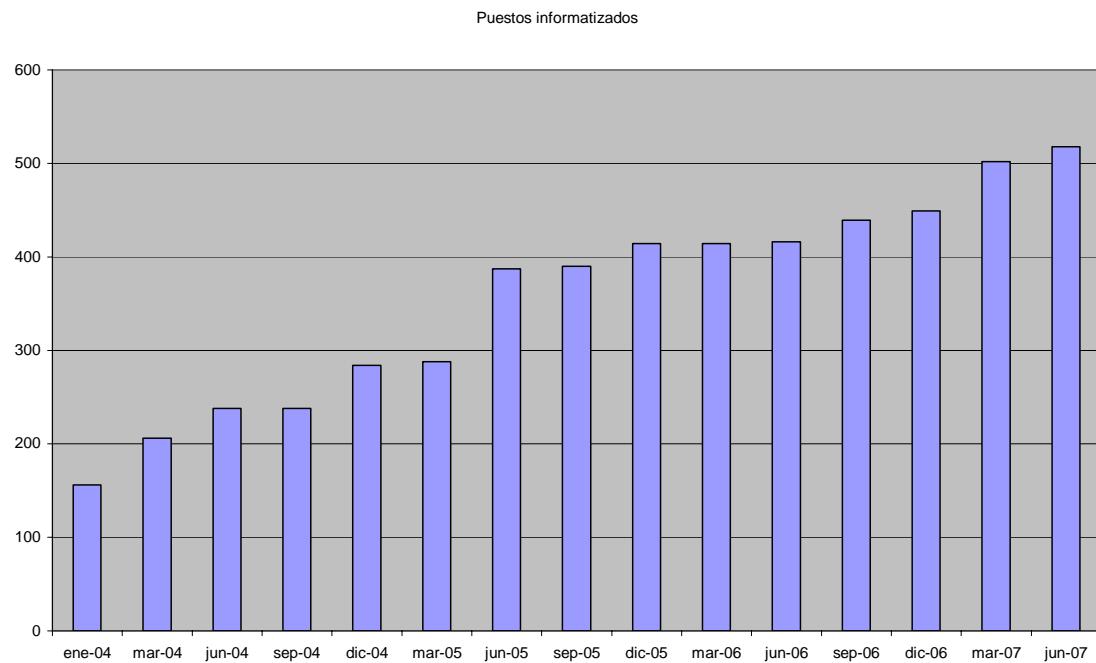
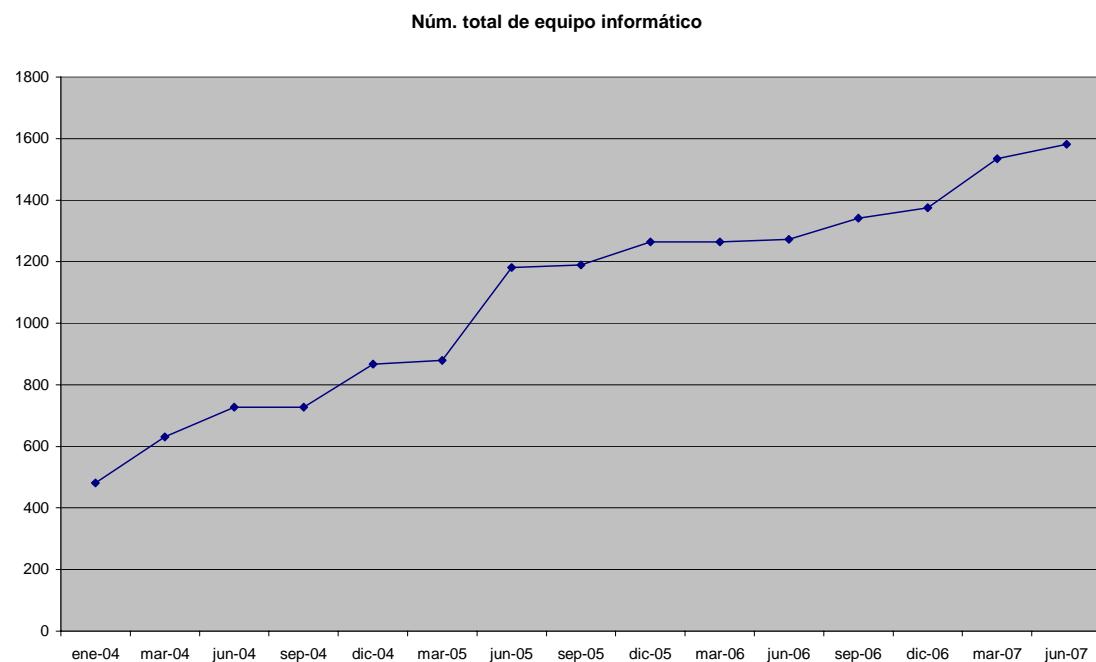


Figura 2.4: Gráfico de evolución de la migración e informatización de centros.

En la figura 2.5 se puede observar el crecimiento del número de puestos informatizados como resultado de la progresiva informatización y de la migración de los centros a e-CAP.

**Figura 2.5:** Gráfico que muestra el creciente número de puestos informáticos.

En el siguiente gráfico de la figura 2.6 vemos el aumento del número de equipos informáticos.

**Figura 2.6:** Gráfico que muestra el aumento del número del equipo informático.

## 2.2 Tecnologías aplicadas

La Web está en continuo avance, debemos dotar a nuestras aplicaciones, servicios e infraestructuras de la tecnología necesaria para hacer realidad este avance.

El desarrollo Web no es una tarea simple ya que mientras que un modelo de programación para aplicaciones de uso común está muy bien establecido y soportado por un gran número de lenguajes, herramientas de desarrollo, la programación Web es una mezcla de varios lenguajes de etiquetas, un gran uso de scripting y plataformas de servidor.

Entre estas tecnologías se encuentra Microsoft .NET Framework. .NET es una plataforma que proporciona la infraestructura y los medios necesarios para que el software que reside en plataformas heterogéneas pueda coexistir y colaborar sin problemas, es un modelo revolucionario para el desarrollo de una nueva generación de software.

El funcionamiento de las aplicaciones y las soluciones de software basadas en esta tecnología supondrán un nuevo reto para los profesionales de TI.

### 2.2.1 Visual Studio 2005

La mayor parte del tiempo que empleamos al desarrollar una aplicación está centrado en los archivos que forman ese proyecto, en organizar código fuente que una vez compilado dará lugar a una librería o a una aplicación. En el ámbito del desarrollo podemos ir más allá y añadir el concepto de *solución* que podemos definir como la agrupación de una serie de proyectos bajo una única estructura, este conjunto de proyectos forman nuestra *solución de negocio*.

Visual Studio es un IDE (Integrated Developement Environment, Entorno Integrado de Desarrollo) muy popular por su rendimiento en cualquier plataforma, facilitando de manera notable el desarrollo de .NET. Mediante el uso de este IDE el desarrollador de software incrementa notablemente la productividad en poco tiempo.

Visual Studio 2005 es la última evolución del IDE de Microsoft para desarrollar soluciones en la nueva plataforma de desarrollo .NET. Esta nueva versión ofrece a los desarrolladores un gran número de ventanas, barras de herramientas, menús, submenús, menús contextuales, cuadros de diálogo... con el fin de escribir código de forma más rápida, eficiente, ampliar la funcionalidad de las aplicaciones, gestionar sus proyectos y soluciones, gestionar referencias de ensamblado, así como discernir los formatos de los archivos del proyecto o la solución en cuestión.

### 2.2.2 Microsoft .NET Framework

Microsoft .NET es una plataforma de desarrollo y ejecución de aplicaciones. Mediante esta plataforma, tenemos a nuestro alcance todas las herramientas y servicios que se necesitan para desarrollar modernas aplicaciones empresariales y de misión

crítica, así como proporcionarnos mecanismos robustos, seguros y eficientes para asegurar que la ejecución de las mismas sea óptima.

Los componentes principales de la plataforma .NET son:

- Un entorno de ejecución de aplicaciones, también llamado “Runtime”, que es un componente de software cuya función es la de ejecutar las aplicaciones .NET e interactuar con el sistema operativo ofreciendo sus servicios y recursos.
- Un conjunto de bibliotecas de funcionalidades y controles reutilizables.
- Un conjunto de lenguajes de programación de alto nivel, junto con sus compiladores y linkers, que permitirán el desarrollo de aplicaciones sobre la plataforma .NET.
- Un conjunto de utilitarios y herramientas de desarrollo para simplificar las tareas más comunes del proceso de desarrollo de aplicaciones
- Documentación y guías de arquitectura.

Por otra parte, .NET representa la evolución COM (Component Object Model), la plataforma de desarrollo de Microsoft anterior a .NET y sobre la cual se basaba el desarrollo de aplicaciones Visual Basic 6 (entre otros tantos lenguajes y versiones).

.NET es una plataforma de ejecución intermedia, las aplicaciones .NET no son ejecutadas directamente por el sistema operativo, como ocurre en el modelo tradicional de desarrollo. En su lugar, las aplicaciones .NET están diseñadas para ser ejecutadas contra un componente de software llamado Entorno de Ejecución (muchas veces también conocido como “Runtime”, o, “Máquina Virtual”). Este componente es el encargado de manejar el ciclo de vida de cualquier aplicación .NET. Definiéndose como código manejado aquel que se ejecuta contra este “Runtime”.

La plataforma Microsoft .NET está completamente basada en el paradigma de Orientación a Objetos.

En la figura 2.7 se pueden apreciar las distintas partes que componen el .NET Framework, incluidas el entorno de ejecución de aplicaciones (CLR, en verde), el conjunto de bibliotecas de funcionalidad reutilizable (.NET Framework Class Library, en azul) y los compiladores y herramientas de desarrollo para los lenguajes .NET (en rojo). Todos estos componentes se montan por encima de la familia de sistemas operativos Windows.

Dentro del conjunto de la .NET Framework Class Library se distinguen 4 subcomponentes principales:

- La Base Class Library (BCL - Biblioteca de Clases Base), que contiene la funcionalidad más comúnmente utilizada para el desarrollo de todo tipo de aplicaciones.

- ADO.NET, que contiene un conjunto de clases que permiten interactuar con bases de datos relacionales y documentos XML.
- ASP.NET, que constituye la tecnología dentro del .NET Framework para construir aplicaciones con interfaz de usuario Web.
- Windows Forms (o simplemente WinForms), que constituye la tecnología dentro del .NET Framework que permite crear aplicaciones con interfaz de usuario basada en formularios y ventanas Windows.

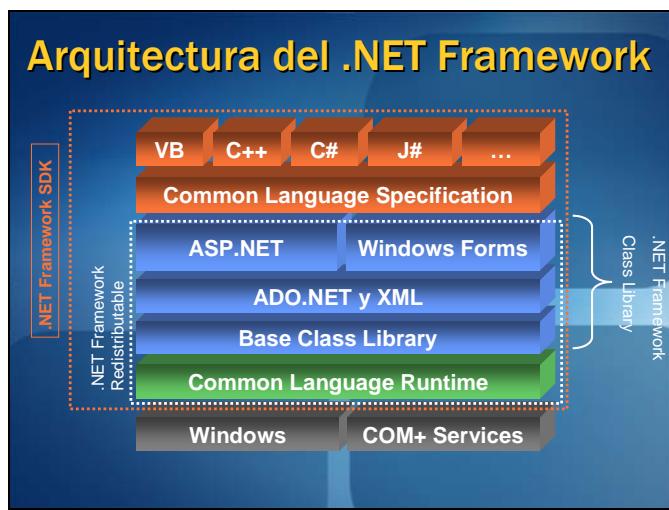


Figura 2.7: Arquitectura de la plataforma .NET

Las ventajas y la innovación aportadas por esta tecnología han hecho que empresas y grupos de desarrollo la trasladen a diferentes plataformas como Linux, Solaris, MacOS,..., este es el caso de un grupo de desarrollo que bajo el nombre de proyecto Mono están haciendo posible que la comunidad GNU/Linux, entre otras, la disfruten y la desarrollen.

El proyecto Mono es una iniciativa, con Novell como espónsor, para desarrollar código abierto en Unix, el Mono Framework, de la plataforma Microsoft .NET, incluyendo tanto las herramientas como la infraestructura necesaria para ejecutar aplicaciones .NET en clientes y en servidores.

Los Arquitecturas y Sistemas Operativos que soportan Mono:

Arquitecturas soportadas	Runtime	Sistema Operativo
<u>s390, s390x (32 and 64 bits)</u>	JIT	Linux
<u>SPARC (32)</u>	JIT	Solaris, Linux
<u>PowerPC</u>	JIT	Linux, Mac OSX
<u>x86</u>	JIT	Linux, FreeBSD, OpenBSD, NetBSD, Microsoft Windows, Solaris, OS X
<u>x86-64: AMD64 and EM64T (64 bit)</u>	JIT	Linux, Solaris
<u>IA64 Itanium2 (64 bit)</u>	JIT	Linux
<u>ARM: little and big endian</u>	JIT	Linux (both the old and the new ABI)
<u>Alpha</u>	JIT	Linux
<u>MIPS</u>	JIT	Linux

Figura 2.8: Tabla que relaciona Arquitecturas y Sistemas Operativos soportados.

La mayoría de distribuciones de Linux incorporan el Mono Framework Runtime y éste puede ser actualizado a través de los respectivos repositorios mediante el gestor de paquetes de cada distribución. A medida que los desarrolladores portan partes de .NET Framework de Microsoft a Mono se actualizan los repositorios, este proceso de actualización es constante lo que conlleva una liberación de una nueva versión del Runtime actualizado a las nuevas funcionalidades.

En el momento de escribir esta memoria el .NET Framework 1.1 había sido completamente portado en cambio no todo el Microsoft .NET Framework 2.0 y 3.0 está soportado.

### 2.2.2.1 ASP.NET

Active Server Pages .NET (ASP.NET) es la evolución de la tecnología existente conocida como Active Server Pages (ASP) introducida por Microsoft hace unos 10 años. ASP.NET no solo es una sucesión, es un nuevo enfoque, un nuevo concepto, un conjunto de tecnologías para el desarrollo de aplicaciones y servicios Web. Está integrado en el .NET Framework y el objetivo que persigue es el de resolver las limitaciones de ASP, su antecesor, y posibilitar la creación de software como servicio.

ASP.NET permite desarrollar y ejecutar tanto aplicaciones Web como servicios Web. De la misma forma que sucedía con ASP, ASP.NET se ejecuta en el servidor HTTP de Microsoft, el Internet Information Server (IIS). El desarrollo de aplicaciones

utilizando esta tecnología consiste en utilizar formularios Web que están diseñados para hacer la creación de aplicaciones Web de forma más sencilla sin tener que mezclar el código de nuestra aplicación con las etiquetas HTML, manipulando en el servidor los elementos de nuestra página Web,...

La programación en ASP.NET, al igual que sucede en cualquier entorno de programación visual, está basada en el uso de controles y eventos, implementando su funcionalidad en fragmentos de código que se ejecutan como respuesta a eventos asociados a los controles de la interfaz con los que puede interactuar el usuario. Mediante esta forma de funcionar le proporciona a ASP.NET un mayor nivel de abstracción, requiere menos código y permite crear aplicaciones más modulares, legibles y mantenibles.

### **2.2.2.2 ADO.NET**

Se llama ADO.NET a todas las clases, interfaces y delegados que se encuentran dentro del espacio de nombres del .NET Framework y representa un eslabón más en el modelo de acceso a datos desarrollado por Microsoft. En la actualidad se ha liberado la versión ADO.NET 2.0, esta versión es la evolución del modelo acceso a datos ideado por Microsoft (ADO, ActiveX Data Object) y utilizado por miles de programadores tanto en aplicaciones Win32 como ASP.

ADO.NET es notablemente más sencillo que ADO debido a que su modelo de datos es más sencillo y a que tiene un campo de aplicación más estrecho, es bastante superior a ADO en términos de prestaciones y escalabilidad pudiendo definir escenarios en modo conectado y en modo desconectado.

ADO.NET se basa en proveedores .NET. Un proveedor de datos .NET es un conjunto de clases que permiten conectarse a una base de datos, ejecutar un comando sobre ella y tener acceso a los resultados de su ejecución, tanto de forma conectada como desconectada sin padecer penalización de prestaciones que se derivan de cambiar entre código administrado y no administrado.

## Capítulo 3

# ANÁLISIS DE ESPECIFICACIONES

Una vez visto el análisis de la organización y los recursos tecnológicos con los que se cuentan para realizar este proyecto, nos centraremos, a continuación, en modelar el comportamiento que debe tener el sistema extrayendo los requerimientos tanto funcionales como no funcionales.

### 3.1 Especificaciones funcionales.

El sistema debe gestionar el equipo informático y administrar los usuarios que accedan a él, para ello, debe permitir o denegar acceso de un usuario al sistema, visualizar la información de un equipo de un centro, debe dar de alta una incidencia de un equipo en el sistema, concluir una incidencia de un equipo cuando esté resuelta,...

#### 3.1.1 Especificaciones de los usuarios del sistema.

Después de analizar la estructura organizativa podemos definir los cuatro tipos de usuarios que harán uso del sistema, cada tipo de usuario corresponde con un grupo organizativo.

**Tipos de usuario:**

- **UAAU:** corresponde al personal de UAAU de un centro CAP o ABS.
- **Coordinador:** se corresponde con la persona que dirige el centro.
- **Informático:** se corresponde con el personal del departamento de informática de una SAP.
- **Master:** este usuario será también del departamento de informática pero con más competencias dentro del sistema que el resto de informáticos.

**3.1.2 Especificaciones de la seguridad en el sistema.**

Para poder acceder al sistema y evitar intrusos, cada usuario debe de tener:

- un login
- un password.

Mediante esta pareja login y password un usuario se autentica ante el sistema y este le autoriza o le deniega el inicio de sesión.

**3.1.3 Detalles de las funcionalidades**

El sistema debe proporcionar las funcionalidades siguientes para asegurar la gestión y administración del equipo informático así como incluir material didáctico para su utilización por los diferentes tipos de usuario. Se definen estas seis:

- Equipo.
- Avería.
- Herramientas.
- Ayuda.
- Administrativa.
- Mantenimiento.

La funcionalidad Equipo debe permitir localizar un equipo informático dentro del sistema. Una vez localizado el equipo se debe mostrar la siguiente información:

- Número de Identificación de Equipo (NIE)
- Número de Serie (NSE)
- Modelo

- Tipo de hardware
- Nombre del equipo (si tiene)
- Función que realiza
- Configuración software y hardware
- Fecha de instalación
- ABS al que pertenece
- Centro al que pertenece
- Ámbito al que pertenece
- Proyecto al que pertenece

La funcionalidad Avería debe llevar a cabo la gestión de una incidencia de un equipo, hay que diferenciar una incidencia nueva en el sistema de una resolución de una incidencia ya abierta.

Si es una incidencia nueva es necesario mostrar los siguientes datos de un equipo:

- Número de Identificación de Equipo (NIE)
- Número de Serie (NSE)
- Modelo
- Tipo de hardware
- Nombre del equipo (si tiene)
- Ubicación

Además se debe proporcionar los siguientes datos administrativos:

- Centro dónde está ubicado el equipo
- Dirección del centro
- Persona de contacto
- Teléfono de contacto

Para poder dar de alta la incidencia en el sistema es necesario además proporcionar la siguiente información:

- El número de avería de la empresa de mantenimiento
- El código / descripción de avería del equipo

Si se trata de una resolución, además de los datos anteriores se debe proporcionar:

- La fecha fin de la incidencia
- El código / resolución de la avería.

La funcionalidad Herramientas debe ofrecer las siguientes funcionalidades:

- Cambiar de contraseña.
- Cambiar de centro
- Visualizar información de las averías
- Visualizar información de los equipos

La funcionalidad Administrativa debe permitir administrar los usuarios de un centro:

- Permitir / denegar el acceso de un usuario al sistema.
- Visualizar el personal UAU del centro.

La funcionalidad Mantenimiento debe permitir el mantenimiento del sistema mediante las funcionalidades:

- Mantenimiento Usuarios
- Mantenimiento Equipos

La funcionalidad Ayuda debe proporcionar al usuario del sistema la información necesaria y detallada de cómo realizar las siguientes tareas:

- Localizar un equipo dentro del sistema
- Gestionar una incidencia
- Cambiar de centro
- Administrar los usuarios del centro
- Cambiar la contraseña

### **3.2 Especificaciones no funcionales.**

Se definen en este apartado los requerimientos no funcionales que vienen impuestos por la empresa.

- El software a desarrollar será para la plataforma Microsoft Windows, la plataforma con la que se trabaja en el CatSalut.
- El sistema operativo
  - del servidor:
    - Microsoft Windows 2003 Server
  - del cliente:
    - Microsoft Windows 2000 SP4 Professional
    - Microsoft Windows XP SP2 Professional.
- Los equipos tienen arquitectura Intel x86 (el procesador más pequeño es un Pentium III a 1 GHZ.)
- Resolución de la pantalla: 1024 x 768 píxeles.

Explorador es el Microsoft Internet Explorer v. 6.0.2900.2180

## Capítulo 4

# DISEÑO

El sistema a desarrollar deberá estar diseñado en una arquitectura cliente / servidor de tres capas, un sistema distribuido en el que se han separado los distintos servicios que componen el sistema.

La división que normalmente se sigue para estos sistemas es la definición de tres capas lógicas (que posteriormente se convertirán en diferentes capas físicas) de la siguiente manera: En la capa más inferior se encuentra la **capa de datos**, en esta capa se encuentra la base de datos; la siguiente capa que se define contiene la **lógica de negocios** que contiene la definición de las operaciones que son necesarias para que nuestro sistema haga su trabajo. Por último tenemos la **capa de presentación** estableciendo la interfaz de comunicación entre los usuarios y el sistema.

La presentación que se hace de estas tres capas en este capítulo es la siguiente: Primeramente se explica la lógica de negocios mediante la representación de los diferentes diagramas de flujo, a continuación la capa de datos donde podremos ver el diseño de la base de datos y por último veremos la interfaz gráfica o capa de presentación.

### 4.1 Metodología de desarrollo

Se entiende por *ciclo de vida de un proyecto software* a todas las etapas por las que pasa un proyecto desde la concepción de la idea que hace surgir la necesidad de diseñar un sistema software pasando por el análisis, desarrollo, implantación y mantenimiento hasta que finalmente muere por ser sustituido por otro sistema.

En este capítulo y en el siguiente centrado en el diseño, implementación y prueba el proceso iterativo e incremental define el modo de trabajo desarrollado. Iterativo porque involucra la gestión del flujo de ejecutables del sistema y cada nueva

versión corrige los defectos de la anterior e incorpora nueva funcionalidad de esta forma se reducen los riesgos más significativos para el éxito del proyecto.

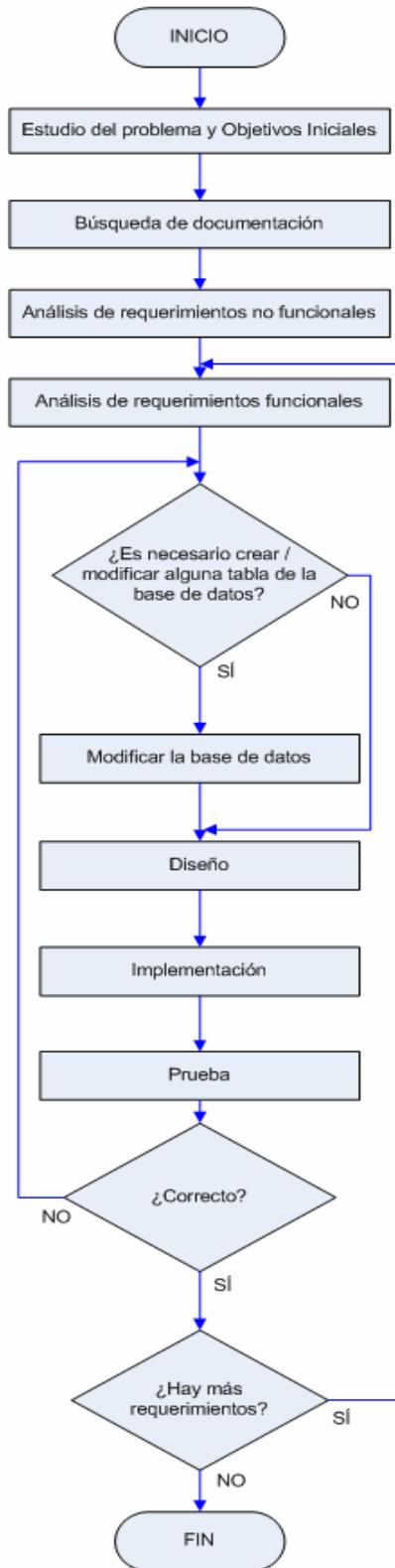


Figura 4.1 Diagrama de flujo de las fases del proyecto.

## 4.2 Diseño funcional.

En esta sección se presentan los diagramas de flujo de cada una de las funcionalidades del sistema recogidas en la fase de análisis de requerimientos. Estos diagramas forman parte de la lógica de negocios y definen la secuencia de acciones a realizar, la información a introducir, los datos a visualizar,... para cada funcionalidad representada en ellos.

### 4.2.1 Diagramas de flujo.

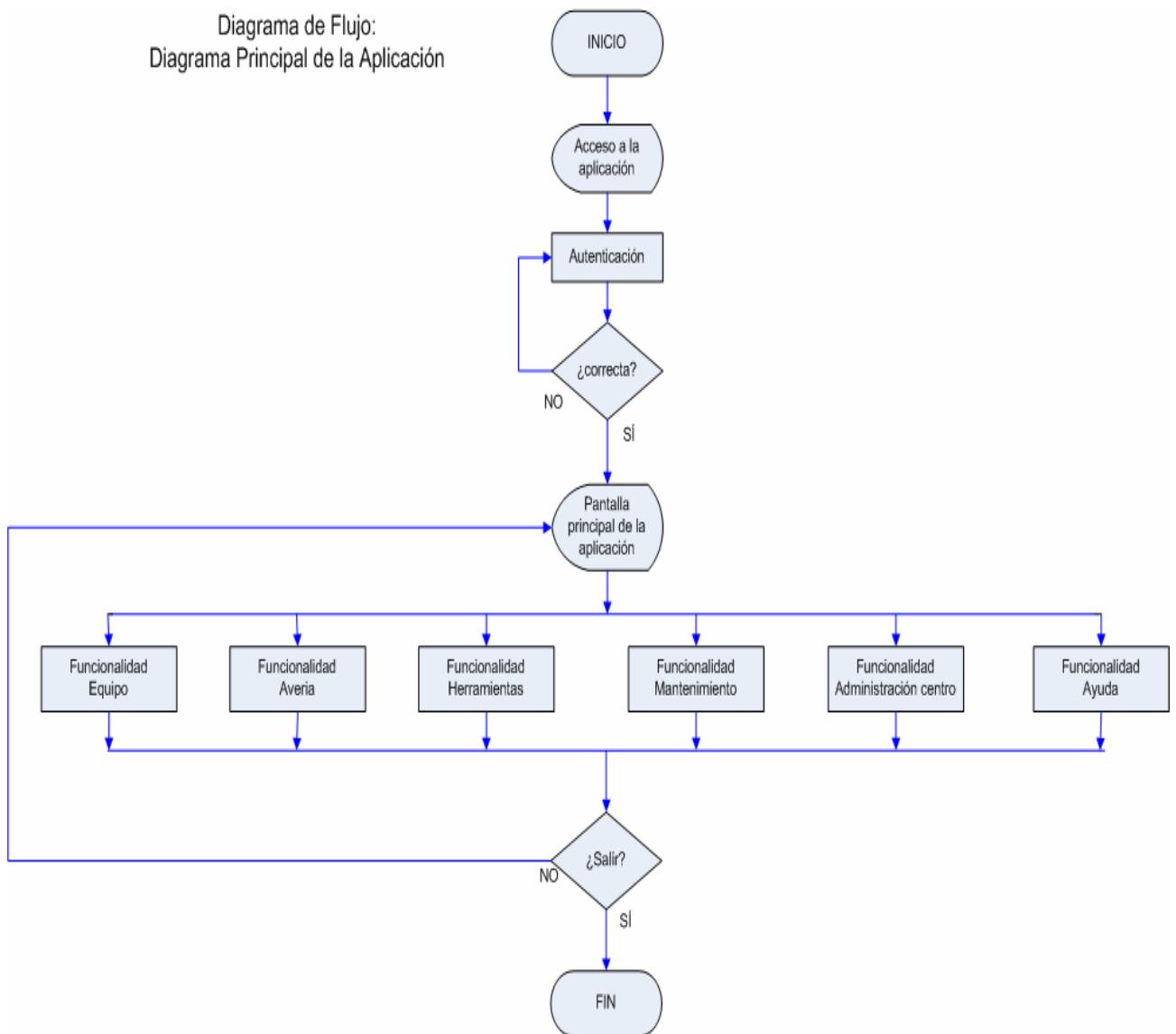


Figura 4.2 Diagrama de flujo principal de la aplicación.

**Diagrama de Flujo:**  
**Diagrama Funcionalidad Equipo**

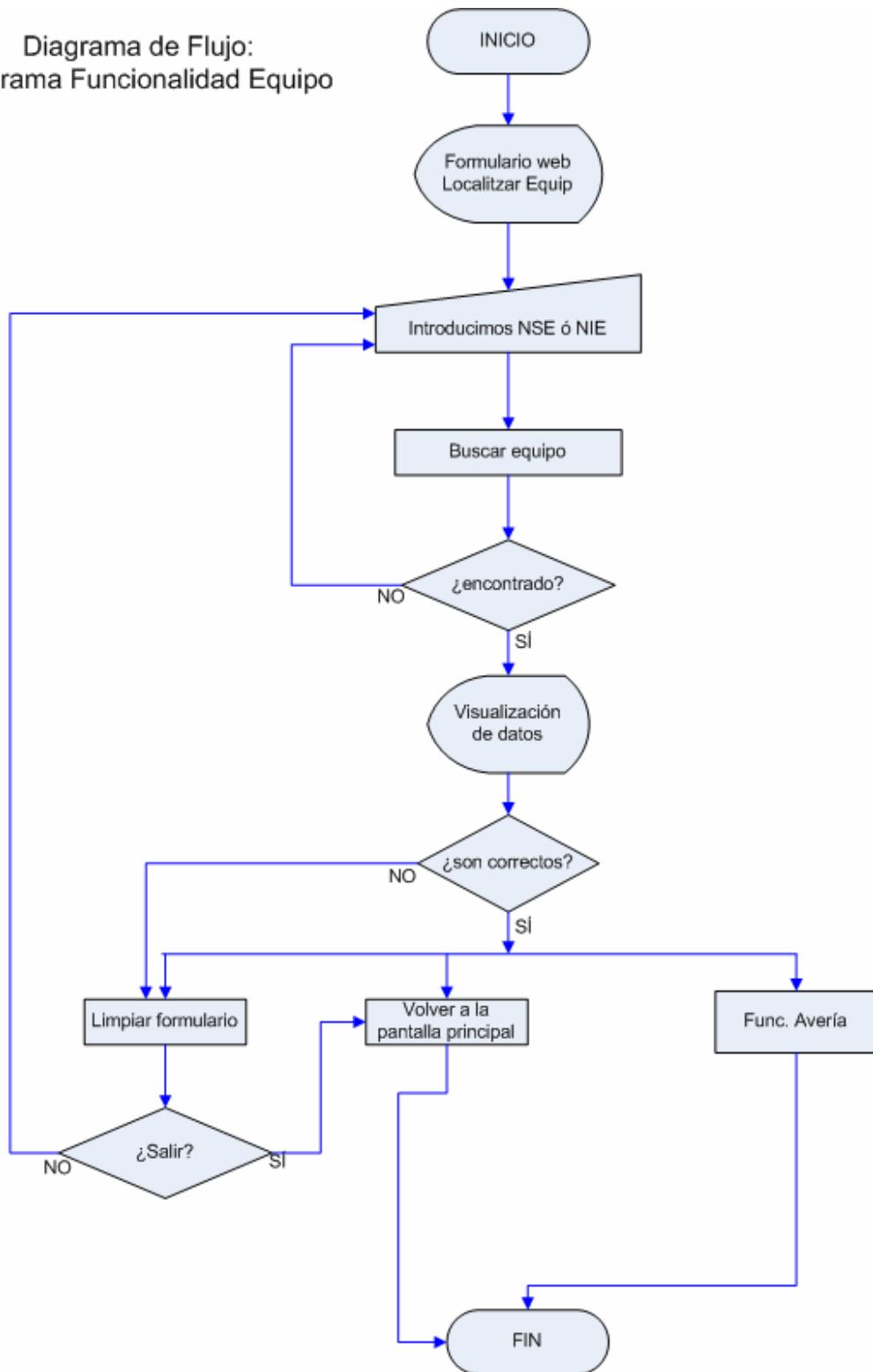


Figura 4.3: Diagrama de flujo de la funcionalidad Localitzar Equip.

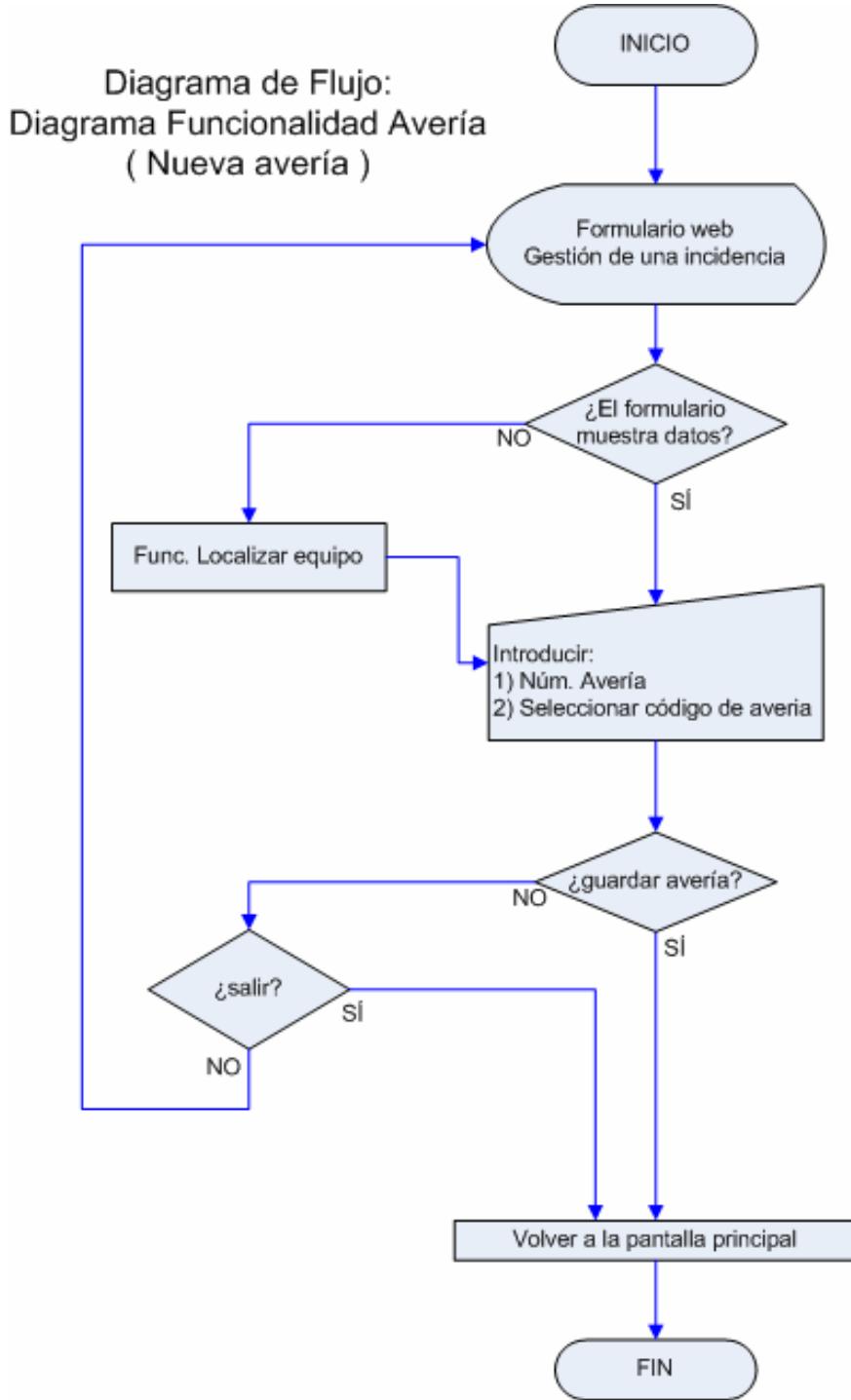


Figura 4.4: Diagrama de flujo de la funcionalidad Avería, el alta de una incidencia de un equipo en el sistema.

**Diagrama de Flujo:  
Diagrama Funcionalidad Avería  
( Resolución de avería )**

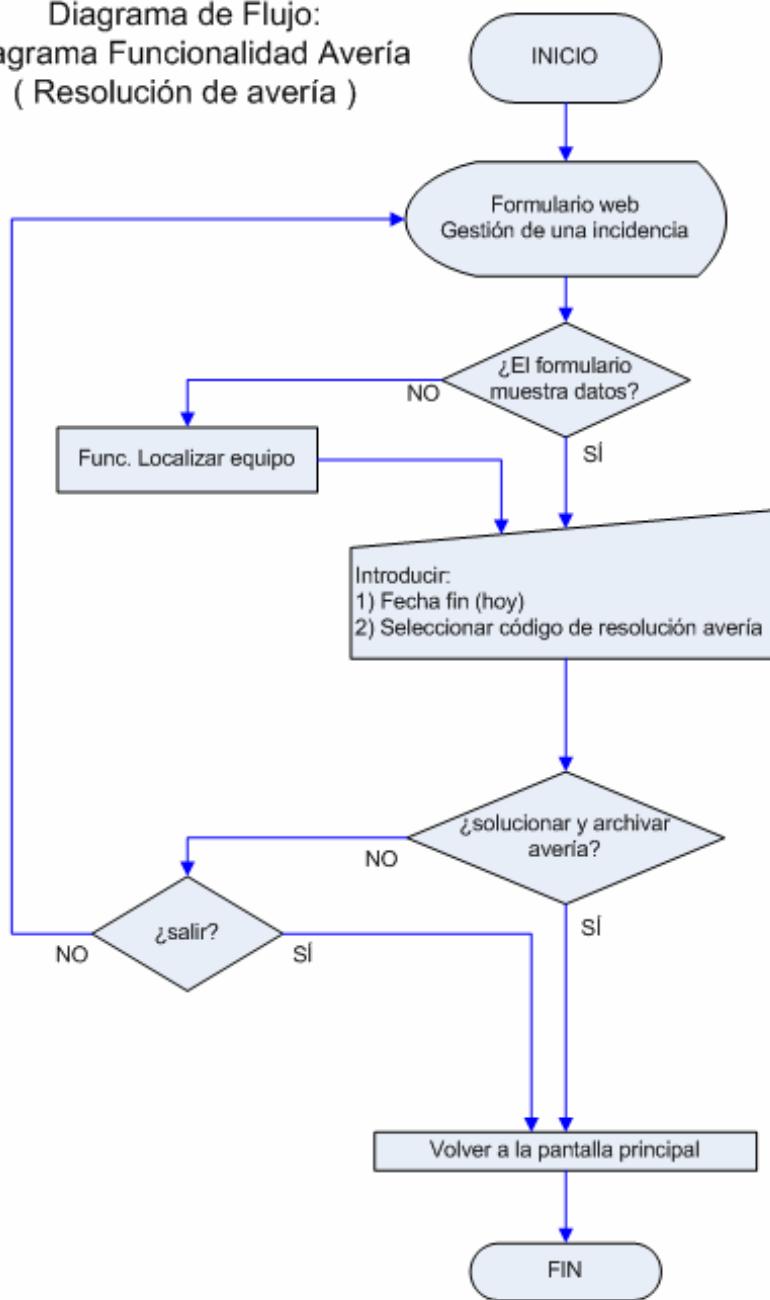


Figura 4.5: Diagrama de flujo de la funcionalidad Avería, la resolución de una incidencia en el sistema.

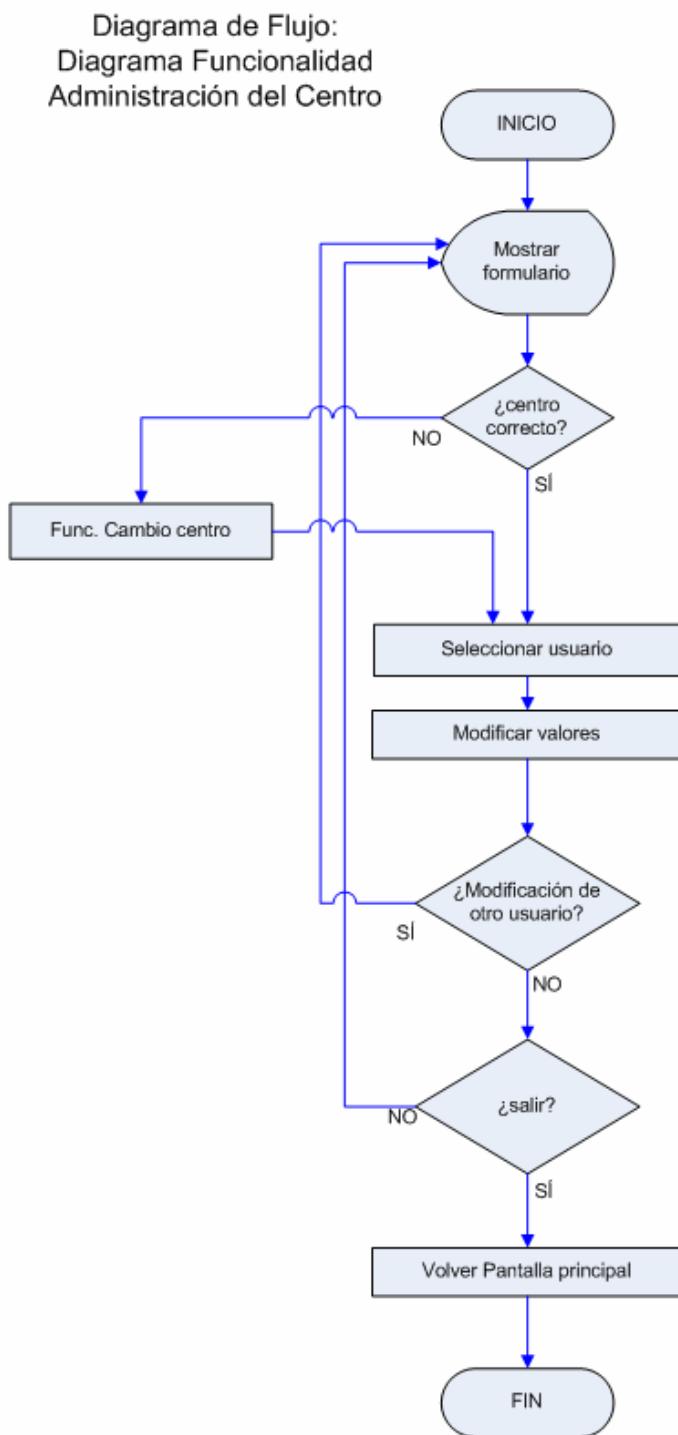


Figura 4.6: Diagrama de flujo de la funcionalidad Administración del centro.

Diagrama de Flujo:  
Diagrama Funcionalidad  
Cambio de Centro

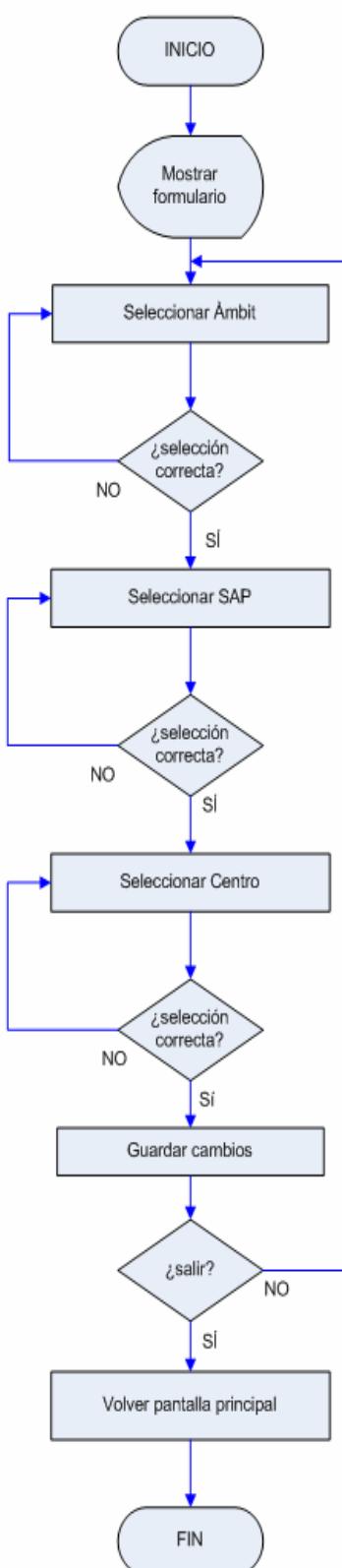


Figura 4.7: Diagrama de flujo de la funcionalidad Cambio de centro.

Diagrama de Flujo:  
Diagrama Funcionalidad  
Cambio de Contraseña

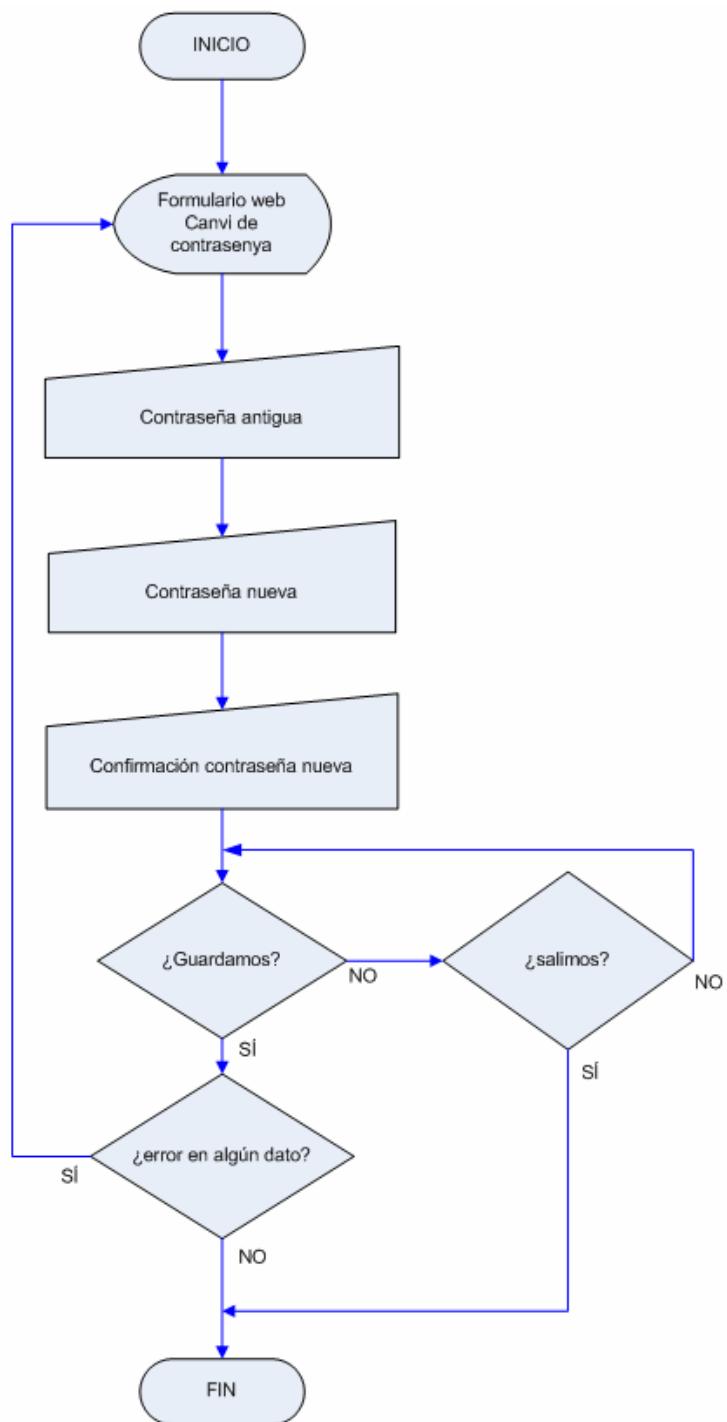


Figura 4.8: Diagrama de flujo de la funcionalidad cambio de contraseña.

#### 4.2.2 Detalle de los usuarios del sistema.

Una vez definidos los diversos usuarios que van a hacer uso del sistema, se especifican cuatro tipos de rol, este rol es el que permitirá a un usuario de la aplicación realizar un conjunto de funcionalidades dentro del sistema:

- Master.
- Informático.
- Usuario avanzado.
- Usuario.

El usuario con rol *master* será el usuario encargado de la gestión del sistema, este usuario será un informático de una SAP o de un Ámbito al que se le haya dado esta responsabilidad. Cabe decir que este rol es el máximo que el sistema puede otorgar.

El usuario con rol *informático* realizará un seguimiento y mantenimiento de los usuarios de la aplicación. Este rol recae sobre los usuarios del departamento de informática del SAP.

Un usuario con rol *usuario avanzado* responde a la figura de un coordinador o de un adjunto de un centro, esta persona es la responsable de gestionar los usuarios del centro que dirige, que a su vez, serán los encargados de gestionar las incidencias del equipo informático de ese centro.

Por último, la persona con rol *usuario* se corresponde con el personal de UAAU de un centro, será la persona encargada de gestionar las incidencias del equipo informático.

Existe una clara relación entre los roles del sistema y el departamento al que pertenece un usuario.

##### 4.2.2.1: Detalle basado en el rol.

A continuación se definen las funcionalidades permitidas agrupadas por rol.

###### Rol Usuario:

1. No tiene acceso a la funcionalidad Administrativa del centro y sí tiene acceso a la funcionalidad Localizar Equipo, Avería, Herramientas y Ayuda.
2. Puede dar de alta y concluir una incidencia de un equipo en el sistema.
3. No puede modificar una incidencia ya concluida ni dar de alta una incidencia de un equipo sin concluir la incidencia anterior.
4. Puede buscar un equipo en el sistema mediante la introducción de cualquiera de los siguientes datos Número de serie del equipo (NSE) ó Número de Identificación de equipo (NIE).
5. Puede buscar una incidencia abierta y concluida de un equipo en el sistema.
6. Puede visualizar el conjunto de incidencias abiertas y concluidas en su centro.

**Rol Usuario avanzado:**

1. Puede acceder a las mismas funcionalidades que accede el rol usuario y a la funcionalidad Administrativa para la gestión de usuarios de su centro.
2. No tiene acceso a la funcionalidad Mantenimiento.
3. Puede realizar todas las operaciones de un usuario.
4. Puede habilitar a una persona de su centro a ser usuario del sistema y poder así gestionar las incidencias del centro.
5. Puede visualizar mediante una lista una serie personas y asignar el rol de usuario del sistema.
6. Puede deshabilitar a una persona para que deje de ser usuario del sistema.

**Rol Informático:**

1. Puede acceder a todas las funcionalidades del sistema a las que accedía el usuario avanzado y realizar todas las operaciones de un usuario avanzado.
2. No tiene restricciones de centro, puede gestionar cualquier centro.
3. No tiene acceso al menú Mantenimiento.

**Rol Master:**

1. Puede acceder a toda la funcionalidad del sistema, no tiene ningún tipo de restricción.

**4.2.2.2: Detalle de seguridad**

A continuación se detallan las reglas que definen la autenticación y la autorización de los usuarios en el sistema:

1. Un usuario del sistema debe tener un login y un password para poder autenticarse.
2. Un usuario del sistema debe autenticarse para realizar funcionalidades definidas por su rol en el sistema.
3. El password de un usuario tiene una vigencia dependiendo del rol asignado y del tiempo de inactividad.
4. Un usuario del sistema pertenece a un rol de manera única.
5. Un rol determina qué funcionalidades puede realizar un usuario en el sistema.
6. La vigencia de un password queda determinada por las siguientes relaciones descritas en la tabla 4.9.
7. Un usuario pertenece a un único centro.
8. Un usuario avanzado pertenece a un único centro.
9. Un informático pertenece a un centro y tiene la funcionalidad de cambiar de centro.
10. El master pertenece a un centro y tiene la funcionalidad de cambiar de centro.

Rol	Caducidad	Cuenta
Master	NUNCA	SIEMPRE HABILITADA
Informático	45 días.	Más de 60 días desde la última validación la cuenta queda deshabilitada.
Usuario avanzado	45 días.	Más de 60 días desde la última validación la cuenta queda deshabilitada.
Usuario	30 días.	Más de 60 días desde la última validación la cuenta queda deshabilitada.

**Tabla 4.9: Definición de la caducidad de los roles y del tipo de cuenta.**

Un usuario master tienen siempre habilitada la cuenta y nunca le caduca el password porque es el usuario responsable de la gestión y administración del propio sistema.

11. Un usuario sólo puede realizar las funcionalidades del sistema restringidas a su rol y al centro al que pertenece.
12. Un usuario avanzado sólo puede realizar las funcionalidades del sistema restringidas a su rol y al centro al que pertenece.
13. Un informático puede realizar las funcionalidades del sistema sin restricción del centro, tiene acceso a todos los centros.
14. El master puede realizar las funcionalidades del sistema sin restricción del centro, tiene acceso a todos los centros.

## 4.3 La base de datos.

En esta sección se definirá la capa de datos del sistema, la base de datos de la que la aplicación obtendrá y añadirá los datos. La base de datos utilizada está implementada en PostgreSQL, no es la base de datos utilizada por la empresa (que es ORACLE), pero se decidió su uso por varias razones, la primera al no disponer de acceso a la base de datos de la empresa, la segunda al ser de libre distribución y haber abundante documentación, la tercera al ser de fácil aprendizaje, además, el diseño del sistema permitirá una migración no traumática hacia ORACLE.

### 4.3.1 Diagrama entidad-relación.

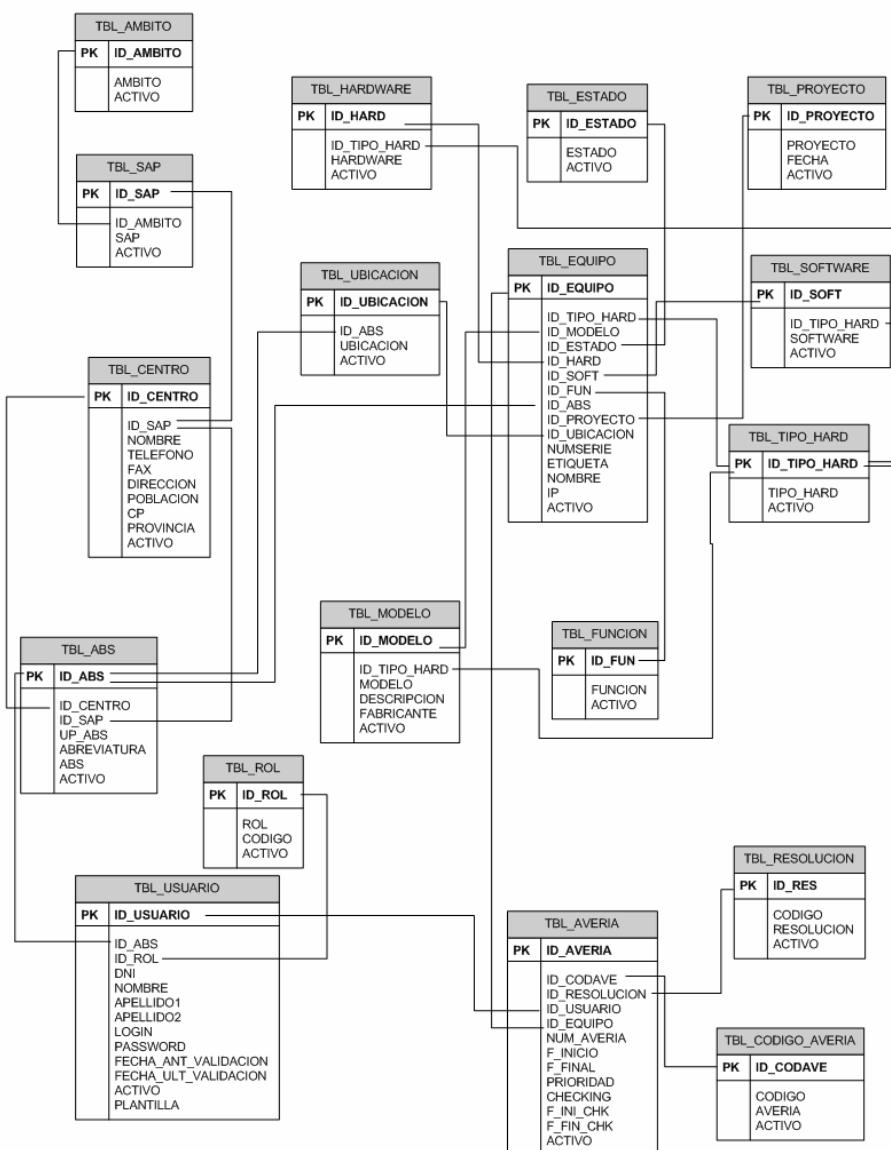


Figura 4.10: Diagrama Entidad-Relación de las tablas de la aplicación.

#### 4.3.2 Diseño de las tablas.

Para proporcionar escalabilidad al sistema se ha diseñado una base de datos con dieciocho tablas que se han distribuido de la siguiente manera: cuatro entidades principales y catorce entidades secundarias. Las entidades secundarias tienen como función albergar información mientras que las entidades principales están formadas por información procedente de estas entidades secundarias y tienen como objetivo representar los objetos comerciales que se usarán en la capa de negocios.

Las tablas secundarias son:

- TBL\_ABS
- TBL\_AMBITO
- TBL\_CODIGO\_AVERIA
- TBL\_ESTADO
- TBL\_FUNCION
- TBL\_HARDWARE
- TBL\_MODELO
- TBL\_PROYECTO
- TBL\_RESOLUCION
- TBL\_ROL
- TBL\_SAP
- TBL\_SOFTWARE
- TBL\_TIPO\_HARD
- TBL\_UBICACION

Las tablas principales son:

- TBL\_AVERIA
- TBL\_CENTRO
- TBL\_EQUIPO
- TBL\_USUARIO

## TBL\_ABS

### Descripción de la tabla:

Esta tabla nos permitirá guardar los datos de los ABS.

TIPO	CAMPO	DESCRIPCION
PK	<b>ID_ABS</b>	Identificador de ABS.
FK	<b>ID_CENTRO</b>	Identificador de Centro
FK	<b>ID_SAP</b>	Identificador de SAP
	<b>UP_ABS</b>	UP
	<b>ABREVIATURA</b>	Abreviatura del ABS
	<b>ABS</b>	Nombre del ABS
	<b>ACTIVO</b>	Activo / No activo.

### Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
<b>ID_ABS</b>	SERIAL
<b>ID_CENTRO</b>	INTEGER NOT NULL
<b>ID_SAP</b>	INTEGER NOT NULL
<b>UP_ABS</b>	NUMÉRICO NOT NULL
<b>ABREVIATURA</b>	VARCHAR(10) NOT NULL
<b>ABS</b>	VARCHAR(80) NOT NULL
<b>ACTIVO</b>	BOOL NOT NULL

### Script que genera la tabla

```
CREATE TABLE TBL_ABS
(
    ID_ABS           SERIAL          ,
    ID_CENTRO        INTEGER        NOT NULL,
    ID_SAP           INTEGER        NOT NULL,
    UP_ABS           NUMERIC(5,0)   NOT NULL,
    ABREVIATURA     VARCHAR(10)    NOT NULL,
    ABS              VARCHAR(80)    NOT NULL,
    ACTIVO          BOOL           NOT NULL,
    CONSTRAINT ID_ABS_PK PRIMARY KEY(ID_ABS),
    CONSTRAINT ID_CENTRO_FK FOREIGN KEY(ID_CENTRO) REFERENCES
    TBL_CENTRO(ID_CENTRO),
    CONSTRAINT ID_SAP_FK FOREIGN KEY(ID_SAP) REFERENCES
    TBL_SAP(ID_SAP)
);
```

## TBL\_AMBITO

### Descripción de la tabla:

Esta tabla nos permitirá guardar los datos referentes a los Ámbitos.

TIPO	CAMPO	DESCRIPCION
PK	<b>ID_AMBITO</b>	Identificador de ámbito.
	<b>AMBITO</b>	Nombre del Ámbito
	<b>ACTIVO</b>	Activo / No activo.

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
ID_AMBITO	SERIAL
AMBITO	VARCHAR(60) NOT NULL
ACTIVO	BOOL NOT NULL

Script de generación de la tabla:

```
CREATE TABLE TBL_AMBITO
(
    ID_AMBITO          SERIAL           ,
    AMBITO              VARCHAR(60)      NOT NULL,
    ACTIVO              BOOL             NOT NULL,
    CONSTRAINT          ID_AMBITO_PK PRIMARY KEY(ID_AMBITO)
);

;
```

**TBL\_AVERIA:**Descripción de la tabla.

En esta tabla guardamos los datos de una avería.

TIPO	CAMPO	DESCRIPCION
PK	ID_AVERIA	Identificador de avería
FK	ID_CODAVE	Identificador de código de avería
FK	ID_RESOLUCION	Identificador de resolución
FK	ID_USUARIO	Identificador de usuario
FK	ID_EQUIPO	Identificador de equipo
	NUM_AVERIA	Código proporcionado por la empresa externa.
	F_INICIO	Fecha de inicio
	F_FINAL	Fecha final
	PRIORIDAD	Prioridad de la avería
	CHECKING	Reabrimos la misma avería
	F_INI_CHK	Fecha inicio checking
	F_FIN_CHK	Fecha final checking
	ACTIVO	Activo / No activo

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
ID_AVERIA	SERIAL
ID_CODAVE	INTEGER NOT NULL
ID_RESOLUCION	INTEGER NOT NULL
ID_USUARIO	INTEGER NOT NULL
ID_EQUIPO	INTEGER NOT NULL
NUM_AVERIA	VARCHAR(10)
F_INICIO	DATE
F_FINAL	DATE
PRIORIDAD	NUMERIC(3,0)
CHECKING	BOOL
F_INI_CHK	DATE
F_FIN_CHK	DATE
ACTIVO	BOOL NOT NULL

Script de generación de la tabla:

```

CREATE TABLE TBL_AVERIA
(
    ID_AVERIA      SERIAL          ,
    ID_CODAVE      INTEGER        NOT NULL,
    ID_RESOLUCION  INTEGER        NOT NULL,
    ID_USUARIO     INTEGER        NOT NULL,
    ID_EQUIPO      INTEGER        NOT NULL,
    NUM_AVERIA     VARCHAR(10)   ,
    F_INICIO       DATE           ,
    F_FINAL        DATE           ,
    PRIORIDAD     NUMERIC(3,0)   ,
    CHECKING       BOOL           ,
    F_INICIO_CHK   DATE           ,
    F_FINAL_CHK    DATE           ,
    ACTIVO         BOOL           NOT NULL,
    CONSTRAINT ID_AVERIA_PK PRIMARY KEY(ID_AVERIA),
    CONSTRAINT ID_CODAVE_FK FOREIGN KEY(ID_CODAVE) REFERENCES
        TBL_CODIGO_AVERIA(ID_CODAVE),
    CONSTRAINT ID_USUARIO_FK FOREIGN KEY(ID_USUARIO)
        REFERENCES TBL_USUARIO(ID_USUARIO),
    CONSTRAINT ID_RES_FK FOREIGN KEY(ID_RES) REFERENCES
        TBL_RESOLUCION(ID_RES),
    CONSTRAINT ID_EQUIPO_FK FOREIGN KEY(ID_EQUIPO) REFERENCES
        TBL_EQUIPO(ID_EQUIPO)
);

```

**TBL\_CENTRO:**Descripción de la tabla.

Esta tabla se encargará de almacenar los datos referentes a un ABS o CAP.

<b>TIPO</b>	<b>CAMPO</b>	<b>DESCRIPCION</b>
PK	<b>ID_CENTRO</b>	Identificador del centro.
FK	<b>ID_SAP</b>	SAP al que pertenece.
	<b>NOMBRE</b>	Nombre del centro.
	<b>TELEFONO</b>	Teléfono del centro.
	<b>FAX</b>	Fax del centro.
	<b>DIRECCION</b>	Dirección del centro.
	<b>POBLACION</b>	Población a la que pertenece el centro.
	<b>CP</b>	Código postal del centro.
	<b>PROVINCIA</b>	Provincia a la que pertenece el centro.
	<b>ACTIVO</b>	El centro está activo o no.

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO	
ID_CENTRO	SERIAL	
ID_SAP	INTEGER	NOT NULL
NOMBRE	VARCHAR(20)	NOT NULL
TELEFONO	VARCHAR(9)	NOT NULL
FAX	VARCHAR(9)	NOT NULL
DIRECCION	VARCHAR(20)	NOT NULL
POBLACION	VARCHAR(20)	NOT NULL
CP	NUMERIC(5,0)	NOT NULL
PROVINCIA	VARCHAR(20)	NOT NULL
ACTIVO	BOOL	NOT NULL,

Script de generación de la tabla:

```
CREATE TABLE TBL_CENTRO
(
    ID_CENTRO      SERIAL          ,
    ID_SAP         INTEGER        NOT NULL,
    NOMBRE         VARCHAR(20)   NOT NULL,
    TELEFONO       VARCHAR(9)    NOT NULL,
    FAX            VARCHAR(9)    NOT NULL,
    DIRECCION      VARCHAR(20)   NOT NULL,
    POBLACION      VARCHAR(20)   NOT NULL,
    CP              NUMERIC(5,0) NOT NULL,
    PROVINCIA      VARCHAR(20)   NOT NULL,
    ACTIVO          BOOL           NOT NULL,
    CONSTRAINT ID_CENTRO_PK PRIMARY KEY(ID_CENTRO)
    CONSTRAINT ID_SAP_FK FOREIGN KEY(ID_SAP) REFERENCES
    TBL_SAP(ID_SAP)
);
```

**TBL\_CODIGO\_AVERIA:**Descripción de la tabla.

Tabla donde registramos los posibles codigos de averías

TIPO	CAMPO	DESCRIPCION
PK	ID_CODAVE	Id código avería
	CODIGO	Numero que identifica el código
	AVERIA	Descripción de la avería
	ACTIVO	Activo / No activo

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO	
ID_CODAVE	SERIAL	
CODIGO	NUMERIC(3,0)	NOT NULL
AVERIA	VARCHAR(80)	
ACTIVO	BOOL	NOT NULL

Script de generación de la tabla:

```

CREATE TABLE TBL_CODIGO_AVERIA
(
    ID_CODAVE      SERIAL ,
    CODIGO         NUMERIC(3,0)      NOT NULL,
    AVERIA         VARCHAR(80)       ,
    ACTIVO         BOOL            NOT NULL,
    CONSTRAINT     ID_CODAVE_PK PRIMARY KEY( ID_CODAVE )
);

```

**TBL\_EQUIPO**Descripción de la tabla:

En esta tabla guardaremos los datos de un equipo.

<b>TIPO</b>	<b>CAMPO</b>	<b>DESCRIPCION</b>
PK	<b>ID_EQUIPO</b>	Núm. de secuencia única
FK	<b>ID_MODELO</b>	Identificador de modelo
FK	<b>ID_ESTADO</b>	Identificador de estado
FK	<b>ID_HARD</b>	Identificador de hardware
FK	<b>ID_SOFT</b>	Identificador de software
FK	<b>ID_FUN</b>	Identificador de función
FK	<b>ID_ABS</b>	Identificador de ABS
FK	<b>ID_PROYECTO</b>	Identificador de proyecto
FK	<b>ID_UBICACION</b>	Identificador de ubicación
	<b>NUMSERIE</b>	Número de serie.
	<b>ETIQUETA</b>	Etiqueta NSE
	<b>NOMBRE</b>	Nombre del equipo
	<b>IP</b>	Dirección IP
	<b>ACTIVO</b>	Activo / No activo

Descripción de los tipos de datos necesarios para su representación:

<b>CAMPO</b>	<b>TIPO DE DATO</b>
<b>ID_EQUIPO</b>	SERIAL
<b>ID_MODELO</b>	INTEGER NOT NULL
<b>ID_ESTADO</b>	INTEGER NOT NULL
<b>ID_HARD</b>	INTEGER NOT NULL
<b>ID_SOFT</b>	INTEGER NOT NULL
<b>ID_FUN</b>	INTEGER NOT NULL
<b>ID_ABS</b>	INTEGER NOT NULL
<b>ID_PROYECTO</b>	INTEGER NOT NULL
<b>ID_UBICACION</b>	INTEGER NOT NULL
<b>NUMSERIE</b>	VARCHAR(20) NOT NULL
<b>ETIQUETA</b>	VARCHAR(12) NOT NULL
<b>NOMBRE</b>	VARCHAR(20) NOT NULL
<b>IP</b>	VARCHAR(15) NOT NULL
<b>ACTIVO</b>	BOOL NOT NULL

Script para la generación de la tabla:

```

CREATE TABLE TBL_EQUIPO
(
    ID_EQUIPO      SERIAL          ,
    ID_MODELO       INTEGER        NOT NULL,
    ID_ESTADO       INTEGER        NOT NULL,
    ID_HARD         INTEGER        NOT NULL,
    ID_SOFT         INTEGER        NOT NULL,
    ID_FUN          INTEGER        NOT NULL,
    ID_ABS          INTEGER        NOT NULL,
    ID_PROYECTO    INTEGER        NOT NULL,
    ID_UBICACION   INTEGER        NOT NULL,
    NUMSERIE       VARCHAR(20)   NOT NULL,
    ETIQUETA        VARCHAR(12)   NOT NULL,
    NOMBRE          VARCHAR(20)   NOT NULL,
    IP              VARCHAR(15)   NOT NULL,
    ACTIVO          BOOL           NOT NULL,
    CONSTRAINT ID_EQUIPO_PK PRIMARY KEY(ID_EQUIPO),
    CONSTRAINT ID_MODELO_FK FOREIGN KEY(ID_MODELO) REFERENCES
    TBL_MODELO(ID_MODELO),
    CONSTRAINT ID_ESTADO_FK FOREIGN KEY(ID_ESTADO) REFERENCES
    TBL_ESTADO(ID_ESTADO),
    CONSTRAINT ID_HARD_FK FOREIGN KEY(ID_HARD) REFERENCES
    TBL_HARDWARE(ID_HARD),
    CONSTRAINT ID_SOFT_FK FOREIGN KEY(ID_SOFT) REFERENCES
    TBL_SOFTWARE(ID_SOFT),
    CONSTRAINT ID_FUN_FK FOREIGN KEY(ID_FUN) REFERENCES
    TBL_FUNCION(ID_FUN),
    CONSTRAINT ID_ABS_FK FOREIGN KEY(ID_ABS) REFERENCES
    TBL_ABS(ID_ABS),
    CONSTRAINT ID_PROYECTO_FK FOREIGN KEY(ID_PROYECTO) REFERENCES
    TBL_PROYECTO(ID_PROYECTO),
    CONSTRAINT ID_UBICACION_FK FOREIGN KEY(ID_UBICACION) REFERENCES
    TBL_UBICACION(ID_UBICACION)
);

```

**TBL\_ESTADO:**Descripción de la tabla.

Esta tabla guardará el estado que podrá tener un equipo que puede ser: en uso, en almacén, dado de baja, donado a una ONG,..

<b>TIPO</b>	<b>CAMPO</b>	<b>DESCRIPCION</b>
PK	<b>ID_ESTADO</b>	Código estado.
	<b>ESTADO</b>	Descripción del estado.
	<b>ACTIVO</b>	Activo / No activo

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
ID_ESTADO	SERIAL
ESTADO	VARCHAR(40)
ACTIVO	BOOL NOT NULL

Script que genera la tabla:

```
CREATE TABLE TBL_ESTADO
(
    ID_ESTADO      SERIAL          ,
    ESTADO         VARCHAR( 40 )   ,
    ACTIVO         BOOL           NOT NULL,
    CONSTRAINT     ID_ESTADO_PK PRIMARY KEY( ID_ESTADO )
);
```

**TBL\_FUNCION:**Descripción de la tabla.

Tabla que describe la función del equipo, si es, de UAAU, de consulta, de biblioteca,...

TIPO	CAMPO	DESCRIPCION
PK	ID_FUN	Código de función
	FUNCION	Nombre de la función.
	ACTIVO	Activo / No activo

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
ID_FUN	SERIAL
FUNCION	VARCHAR(80)
ACTIVO	BOOL NOT NULL

Script para generar la tabla:

```
CREATE TABLE TBL_FUNCION
(
    ID_FUN      SERIAL          ,
    FUNCION    VARCHAR( 80 )   ,
    ACTIVO     BOOL           NOT NULL,
    CONSTRAINT     ID_FUN_PK PRIMARY KEY( ID_FUN )
);
```

**TBL\_HARDWARE:**Descripción de la tabla.

Tabla con la configuración hardware del equipo.

<b>TIPO</b>	<b>CAMPO</b>	<b>DESCRIPCION</b>
PK	<b>ID_HARD</b>	Código hardware.
FK	<b>ID_TIPO_HARD</b>	Id del tipo de hardware
	<b>HARDWARE</b>	Descripción de la configuración hardware.
	<b>ACTIVO</b>	Activo / No activo

Descripción de los tipos de datos necesarios para su representación:

<b>CAMPO</b>	<b>TIPO DE DATO</b>
<b>ID_HARD</b>	SERIAL
<b>ID_TIPO_HARD</b>	INTEGER
<b>HARDWARE</b>	VARCHAR( 80 )
<b>ACTIVO</b>	BOOL NOT NULL ,

Script de generación de la tabla:

```
CREATE TABLE TBL_HARDWARE
(
    ID_HARD          SERIAL           ,
    ID_TIPO_HARD     INTEGER         NOT NULL ,
    HARDWARE        VARCHAR( 80 )   ,
    ACTIVO          BOOL            NOT NULL ,
    CONSTRAINT ID_HARD_PK PRIMARY KEY(ID_HARD),
    CONSTRAINT ID_TIPO_HARD_FK FOREIGN KEY(ID_TIPO_HARD) REFERENCES
    TBL_TIPO_HARD(ID_TIPO_HARD)
);
```

**TBL\_MODELO**Descripción de la tabla.

Tabla que nos permitirá guardar el modelo de equipo

<b>TIPO</b>	<b>CAMPO</b>	<b>DESCRIPCION</b>
PK	<b>ID_MODELO</b>	Id de modelo
FK	<b>ID_TIPO_HARD</b>	
	<b>MODELO</b>	Modelo del equipo
	<b>DESCRIPCION</b>	Descripción
	<b>FABRICANTE</b>	Fabricante
	<b>ACTIVO</b>	Activo / No activo

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
ID_MODELO	SERIAL
ID_TIPO_HARD	INTEGER
MODELO	VARCHAR(20)
DESCRIPCION	VARCHAR(40)
FABRICANTE	VARCHAR(20)
ACTIVO	BOOL NOT NULL

Script de generación de la tabla:

```
CREATE TABLE TBL_MODELO
(
    ID_MODELO      SERIAL,
    ID_TIPO_HARD   INTEGER          NOT NULL,
    MODELO         VARCHAR( 20 )    ,
    DESCRIPCION    VARCHAR( 80 )    ,
    FABRICANTE     VARCHAR( 20 )    ,
    ACTIVO         BOOL            NOT NULL,
    CONSTRAINT ID_MODELO_PK PRIMARY KEY(ID_MODELO),
    CONSTRAINT ID_TIPO_HARD_FK FOREIGN KEY(ID_TIPO_HARD) REFERENCES
    TBL_TIPO_HARD(ID_TIPO_HARD)
);
```

**TBL\_PROYECTO**

En esta tabla guardaremos los datos de los diferentes proyectos para los cuales se necesita equipo informático, por ejemplo, renovación, E-CAP,...

Descripción de la tabla.

Tabla que nos describe el tipo de proyecto ( p.e. E-CAP, renovación,...)

TIPO	CAMPO	DESCRIPCION
PK	<b>ID_PROYECTO</b>	Id. De proyecto
	<b>PROYECTO</b>	Descripción del proyecto
	<b>FECHA</b>	Fecha del proyecto
	<b>ACTIVO</b>	Activo / No activo

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
ID_PROYECTO	SERIAL
PROYECTO	VARCHAR( 20 )
FECHA	DATE
ACTIVO	BOOL NOT NULL

Script de generación de la tabla:

```
CREATE TABLE TBL_PROYECTO
```

```

(
    ID_PROYECTO      SERIAL      ,
    PROYECTO         VARCHAR( 20 )   ,
    FECHA            DATE        ,
    ACTIVO           BOOL        NOT NULL ,
    CONSTRAINT       ID_PROYECTO_PK PRIMARY KEY( ID_PROYECTO )
);

```

**TBL\_RESOLUCION:**Descripción de la tabla.

Tabla va a almacenar la resolución de la avería.

TIPO	CAMPO	DESCRIPCION
PK	<b>ID_RES</b>	Id. de la resolución
	<b>CODIGO</b>	Número que identifica el código
	<b>RESOLUCION</b>	Descripción de la resolución
	<b>ACTIVO</b>	Activo/ No activo

Descripción de los tipos de datos necesarios para su representación:

CAMPO	DESCRIPCION
<b>ID_RES</b>	SERIAL
<b>CODIGO</b>	NUMERIC( 3 , 0 ) NOT NULL
<b>RESOLUCION</b>	VARCHAR( 80 )
<b>ACTIVO</b>	BOOL NOT NULL

Script de generación de la tabla:

```

CREATE TABLE TBL_RESOLUCION
(
    ID_RES          SERIAL      ,
    CODIGO          NUMERIC( 3 , 0 )   NOT NULL ,
    RESOLUCION      VARCHAR( 80 )     NOT NULL ,
    ACTIVO          BOOL        NOT NULL ,
    CONSTRAINT      ID_RES_PK PRIMARY KEY( ID_RES )
);

```

**TBL\_ROL:**Descripción de la tabla.

Tabla va a almacenar la información de cada ROL.

TIPO	CAMPO	DESCRIPCION
PK	<b>ID_ROL</b>	Id. del ROL
	<b>ROL</b>	Nombre del ROL
	<b>CODIGO</b>	Código numérico de ROL
	<b>ACTIVO</b>	Activo/ No activo

Descripción de los tipos de datos necesarios para su representación:

CAMPO	DESCRIPCION
ID_ROL	SERIAL
ROL	VARCHAR(50)
CODIGO	NUMERIC (2,0) NOT NULL
ACTIVO	BOOL NOT NULL

Script de generación de la tabla:

```
CREATE TABLE TBL_ROL
(
ID_ROL          SERIAL           ,
ROL             VARCHAR(50)      ,
CODIGO          NUMERIC(2,0)     NOT NULL,
ACTIVO          BOOL            NOT NULL,
CONSTRAINT ID_ROL_PK PRIMARY KEY(ID_ROL)
);
```

**TBL\_SAP**Descripción de la tabla:

Esta tabla nos permitirá guardar los datos de cada SAP.

TIPO	CAMPO	DESCRIPCION
PK	ID_SAP	Identificador de SAP.
FK	ID_AMBITO	Identificador de Ámbito
	SAP	Descripción
	ACTIVO	Activo / No activo.

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
ID_SAP	SERIAL
ID_AMBITO	INTEGER NOT NULL
AMBITO	VARCHAR(80) NOT NULL
ACTIVO	BOOL NOT NULL

Script que genera la tabla

```
CREATE TABLE TBL_SAP
(
ID_SAP          SERIAL           ,
ID_AMBITO       INTEGER          NOT NULL,
AMBITO          VARCHAR(80)      NOT NULL,
ACTIVO          BOOL            NOT NULL,
```

```

    ACTIVO          BOOL      NOT NULL,
CONSTRAINT      ID_SAP_PK PRIMARY KEY(ID_SAP),
CONSTRAINT      ID_AMBITO_FK FOREIGN KEY(ID_AMBITO) REFERENCES
                  TBL_AMBITO(ID_AMBITO)
);

```

**TBL\_SOFTWARE:**Descripción de la tabla.

Tabla donde vamos a guardar los datos del Software.

TIPO	CAMPO	DESCRIPCION
PK	<b>ID_SOFT</b>	Código software.
	<b>ID_TIPO_HARD</b>	Id de Tipo de hardware.
	<b>SOFTWARE</b>	Listado detallado del software instalado.
	<b>ACTIVO</b>	Activo / No activo

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
<b>ID_SOFT</b>	SERIAL
<b>ID_TIPO_HARD</b>	INTEGER
<b>SOFTWARE</b>	VARCHAR(80)
<b>ACTIVO</b>	BOOL      NOT NULL

Script de generación de la tabla:

```

CREATE TABLE TBL_SOFTWARE
(
    ID_SOFT          SERIAL           ,
    ID_TIPO_HARD     INTEGER          NOT NULL,
    SOFTWARE         VARCHAR(80)       ,
    ACTIVO           BOOL            NOT NULL,
    CONSTRAINT ID_SOFT_PK PRIMARY KEY(ID_SOFT),
    CONSTRAINT ID_TIPO_HARD_FK FOREIGN KEY(ID_TIPO_HARD) REFERENCES
                  TBL_TIPO_HARD(ID_TIPO_HARD)
);

```

**TBL\_TIPO\_HARD**Descripción de la tabla:

Tabla donde vamos a guardar los datos del tipo de hardware.

TIPO	CAMPO	DESCRIPCION
PK	<b>ID_TIPO_HARD</b>	Identificador
	<b>TIPO_HARD</b>	Tipo de hardware
	<b>ACTIVO</b>	Activo / No activo.

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
ID_USUARIO	SERIAL
TIPO_HARD	VARCHAR(40)
ACTIVO	BOOL NOT NULL

Script de generación de la tabla:

```
CREATE TABLE TBL_TIPO_HARD
(
    ID_TIPO_HARD      SERIAL          ,
    TIPO_HARD         VARCHAR( 40 )   ,
    ACTIVO            BOOL           NOT NULL,
    CONSTRAINT        ID_TIPO_HARD_PK PRIMARY KEY( ID_TIPO_HARD )
) ;
```

**TBL\_UBICACION:**Descripción de la tabla.

Tabla que guarda la ubicación de un equipo.

TIPO	CAMPO	DESCRIPCION
PK	ID_UBICACION	Id ubicacion
FK	ID_CENTRO	
	UBICACION	Descripcion de la ubicacion
	ACTIVO	Activo / No activo

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
ID_UBICACION	SERIAL
ID_CENTRO	INTEGER NOT NULL
UBICACION	INTEGER NOT NULL
ACTIVO	BOOL NOT NULL

Script de generación de la tabla:

```
CREATE TABLE TBL_UBICACION
(
    ID_UBICACION      SERIAL          ,
    ID_CENTRO         INTEGER        NOT NULL,
    UBICACION         VARCHAR( 20 )   ,
    ACTIVO            BOOL           NOT NULL,
    CONSTRAINT        ID_UBICACION_PK PRIMARY KEY( ID_UBICACION )
    CONSTRAINT        ID_CENTRO_FK FOREIGN KEY( ID_CENTRO ) REFERENCES
                      TBL_CENTRO( ID_CENTRO )
) ;
```

**TBL\_USUARIO**Descripción de la tabla:

Esta tabla nos permitirá guardar los datos de los usuarios.

TIPO	CAMPO	DESCRIPCION
PK	<b>ID_USUARIO</b>	Identificador de usuario.
FK	<b>ID_CENTRO</b>	Identificador de centro.
FK	<b>ID_ROL</b>	Identificador de Rol.
	<b>DNI</b>	DNI del usuario.
	<b>NOMBRE</b>	Nombre del usuario.
	<b>APELLIDO1</b>	Primer apellido.
	<b>APELLIDO2</b>	Segundo apellido.
	<b>LOGIN</b>	login del usuario.
	<b>PASWD</b>	password.
	<b>FECHA_ANT_VALIDACION</b>	Fecha de anterior validación.
	<b>FECHA_ULT_VALIDACION</b>	Fecha de última validación.
	<b>ACTIVO</b>	Activo / No activo.
	<b>PLANTILLA</b>	Plantilla / No Plantilla del centro

Descripción de los tipos de datos necesarios para su representación:

CAMPO	TIPO DE DATO
<b>ID_USUARIO</b>	SERIAL
<b>ID_CENTRO</b>	INTEGER NOT NULL
<b>ID_ROL</b>	INTEGER NOT NULL
<b>DNI</b>	VARCHAR(7) NOT NULL
<b>NOMBRE</b>	VARCHAR(20) NOT NULL
<b>APELLIDO1</b>	VARCHAR(20) NOT NULL
<b>APELLIDO2</b>	VARCHAR(20) NOT NULL
<b>LOGIN</b>	VARCHAR(8) NOT NULL
<b>PASWD</b>	VARCHAR(10) NOT NULL
<b>FECHA_ANT_VALIDACION</b>	DATE NOT NULL
<b>FECHA_ULT_VALIDACION</b>	DATE
<b>ACTIVO</b>	BOOL NOT NULL
<b>PLANTILLA</b>	BOOL NOT NULL

Script de generación de la tabla:

```
CREATE TABLE TBL_USUARIO
(
    ID_USUARIO          SERIAL           ,
    ID_CENTRO           INTEGER          NOT NULL,
    ID_ROL              INTEGER          NOT NULL,
    DNI                 VARCHAR(7)       NOT NULL,
    NOMBRE              VARCHAR(20)      NOT NULL,
    APELLIDO1           VARCHAR(20)      NOT NULL,
    APELLIDO2           VARCHAR(20)      NOT NULL,
    LOGIN               VARCHAR(8)       NOT NULL,
    PASSWORD            VARCHAR(10)      NOT NULL,
    FECHA_ANT_VALIDACION DATE           NOT NULL,
    FECHA_ULT_VALIDACION DATE           ,
)
```

```

    ACTIVO          BOOL           NOT NULL,
    PLANTILLA       BOOL           NOT NULL,
    CONSTRAINT      ID_USUARIO_PK PRIMARY KEY(ID_USUARIO),
    CONSTRAINT      ID_CENTRO_FK FOREIGN KEY(ID_CENTRO)
                    REFERENCES TBL_CENTRO(ID_CENTRO),
    CONSTRAINT      ID_ROL_FK   FOREIGN KEY(ID_ROL) REFERENCES
                    TBL_ROL(ID_ROL)

);

```

## 4.4. Diseño de la Intefície Gráfica de Usuario

Un gran número de sitios Web que pueden ser visitados en la actualidad en Internet marean y confunden al visitante. Esto se debe a una mala arquitectura de la información y se observa en que las páginas están mal estructuradas y se pierde tiempo buscando la información o bien porque éstas no están actualizadas o bien porque abusen de mecanismos que llaman la atención y al final no se pueda distinguir la información importante de la que no.

Durante el proceso de diseño se ha tenido en cuenta el tipo de usuario que iba a hacer uso de la aplicación, sus necesidades, así como el ámbito profesional en el que se iba a implantar. El diseño tenía que ser sencillo, que facilitase una ejecución simple, de forma intuitiva y con un aprendizaje rápido de la aplicación.

### 4.4.1 Características de la aplicación Web.

A la hora de diseñar la aplicación se han definido por objetivos a alcanzar:

- **Navegabilidad** por el sitio Web de forma intuitiva y sencilla tanto para usuarios principiantes como avanzados.
- **Facilidad, usabilidad y rapidez.**
- **Homogeneidad y Consistencia** en el diseño de las páginas que componen la aplicación Web.
- **Look and Feel** de Microsoft Windows.
- **Coherencia** con otras aplicaciones que usan los usuarios.
- **Combinación de colores** no saturados, en tonalidades suaves y fríos que transmiten calma, seguridad al usuario.
- **Un tipo y tamaño de letra** proporcional a la resolución del monitor.
- **Diseño escalable y de fácil mantenimiento.**

Esta breve enumeración de características no pretende ser el documento *estilo guía* que recoja la información a tener en cuenta para diseñar la GUI tan solo es una aproximación.

#### 4.4.2 Maqueta de las GUI

Llegados a este punto y teniendo en cuenta las características descritas en el punto anterior vamos a presentar las maquetas (“mock-up”) de papel de las GUI más importantes que forman la aplicación.

Sabiendo la navegabilidad podemos plantear escenarios “storyboards” para comprobar como las GUI soportan las diferentes tareas. Esta maqueta se puede presentar a los usuarios para una primera evaluación antes de realizar “mock-up” ejecutables.

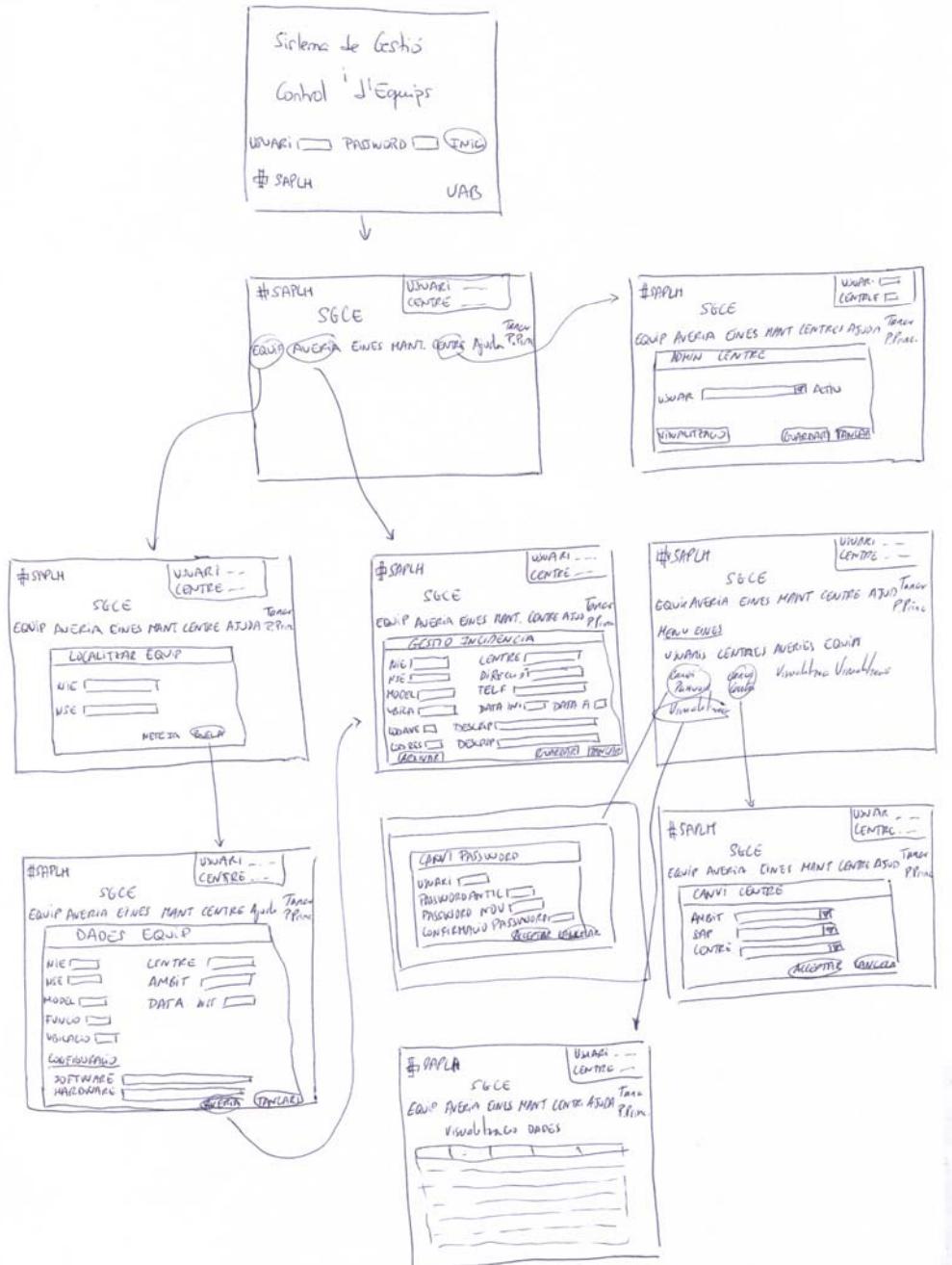


Figura 4.11: “Mock-up” de papel de las principales pantallas de la aplicación.

#### 4.4.3 Aplicación de las características a las páginas Web.

En el apartado 4.6.1 se establecieron las características importantes que debían aplicarse a las páginas Web de la aplicación. En este apartado se hará una demostración de la aplicación de dichas características.

En la figura 4.12 se puede ver la página de acceso a la aplicación, en ella podemos apreciar que se han aplicado varias de las características. Una de las características aplicadas es el color no saturado y suave de fondo de la página y cómo destaca el color también suave de las letras del título y las etiquetas de “*Usuari*“ y “*Password*“. Estos colores fríos transmiten serenidad y seguridad al usuario que va a acceder. Otra característica aplicada se encuentra en la disposición de los cuadros de texto para introducir el login y la contraseña, esta disposición es exactamente igual al formato de acceso de los programas e-CAP y SIAPWEB, se mantienen a la misma altura horizontal. El tamaño de letra utilizado es proporcional a la resolución definida y el tipo de letra es amigable.

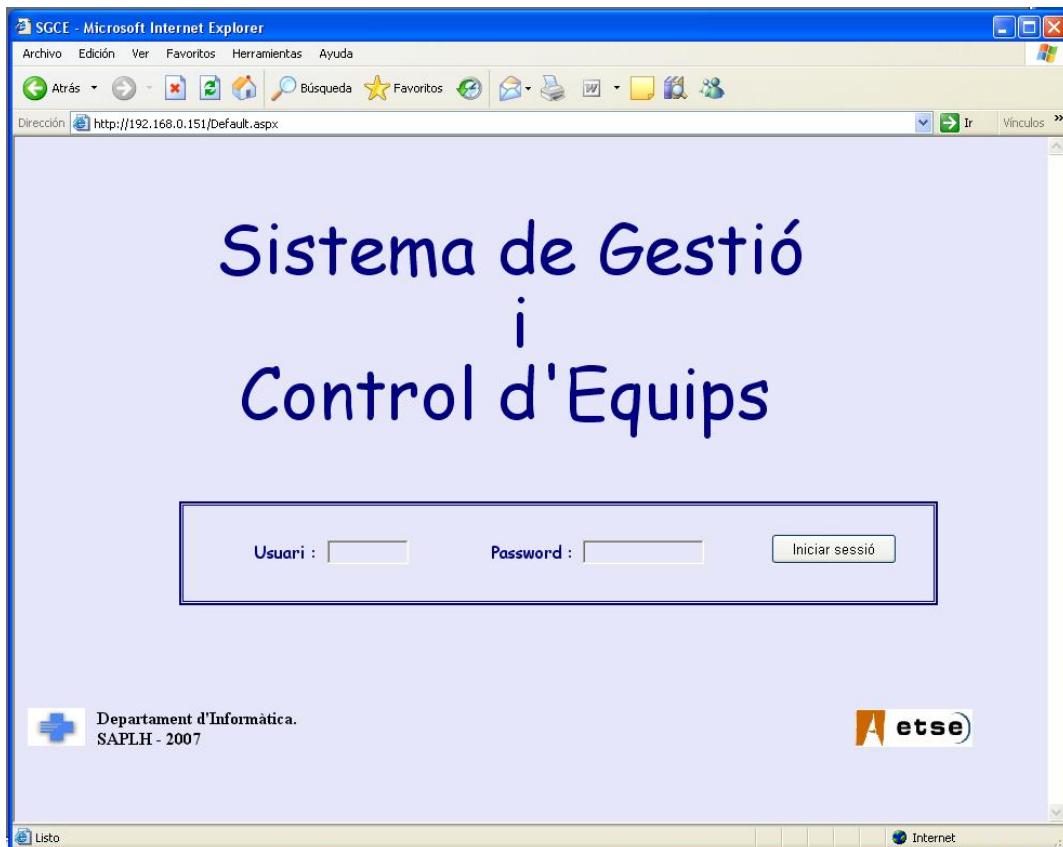


Figura 4.12: Entrada a la aplicación.

La homogeneidad y consistencia de la aplicación se ha conseguido haciendo uso de una característica de ASP.NET 2.0, las páginas maestras. Las páginas maestras permiten definir que todas las páginas tengan elementos comunes como logotipos, encabezados, menús,...

En la figura 4.13 podemos ver la pantalla principal de la aplicación, en la figura 4.14 vemos el formulario Web *Localitzar un Equip* y en la figura 4.15 vemos el formulario *Dades de l'Equip*. En todas estas capturas de pantallas podemos apreciar varias características aplicadas. Primeramente vemos cómo la apariencia se mantiene en cada formulario dando homogeneidad a la aplicación. Asimismo podemos ver una distribución del espacio para los distintos componentes que aparecen, el logotipo en la parte superior izquierda, información del usuario que se ha registrado en la parte superior derecha, el título del sistema en la parte superior central, un menú principal con las funcionalidades principales en la línea horizontal inferior al título. Además, el uso de los colores, el tipo y tamaño de letra antes mencionados se preservan.

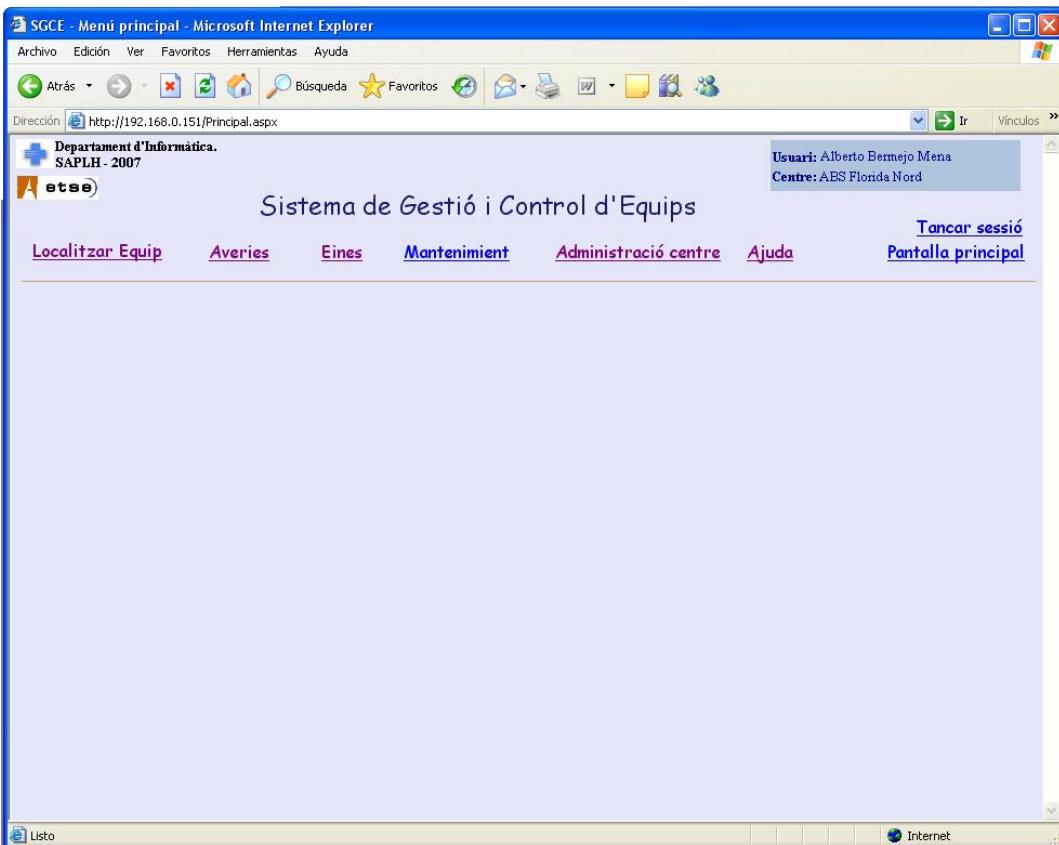


Figura 4.13: Pantalla principal de la aplicación.

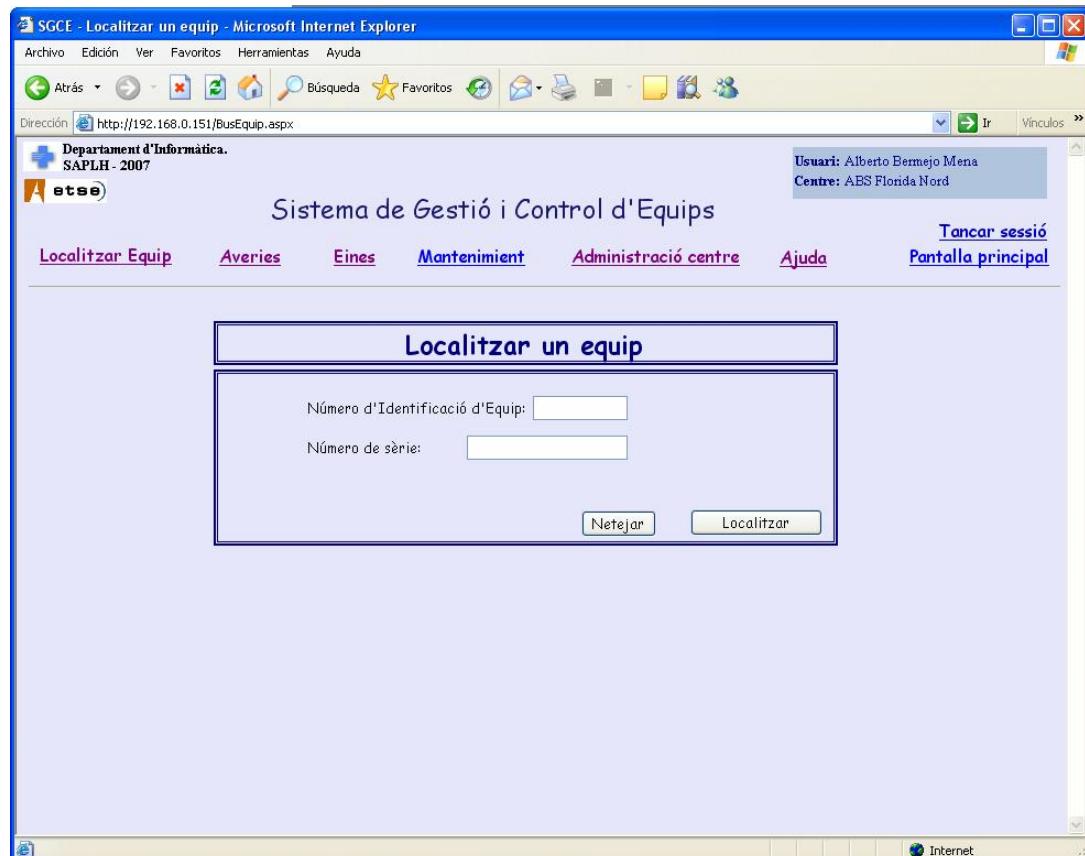


Figura 4.14: Pantalla Localitzar Equip.



Figura 4.15: Pantalla que visualiza los datos de un equipo.

La navegabilidad que se pretende ofrecer al usuario queda manifestada mediante el menú principal que aparece en todos los formularios de la aplicación Web, de esta manera, con independencia de la tarea que realice el usuario en ese momento, podrá cambiarla mediante la selección de una opción del menú principal, sin necesidad de tener que finalizar la tarea actual.

Esta característica se puede observar en las capturas de pantalla que aparecen en las figuras 4.16, 4.17 y 4.18, donde el usuario puede pasar de gestionar una avería (figura 4.16) a administrar a los usuarios del centro (figura 4.17) o a cambiar de centro (figura 4.18).

Figura 4.16: Formulario de gestión de una incidencia en un equipo.

The screenshot shows a Microsoft Internet Explorer window with the title "SGCE - Administració del centre. - Microsoft Internet Explorer". The address bar shows the URL "http://192.168.0.151/AdminCentre.aspx". The page header includes the logo of the Department of Informatics (SAPLH - 2007) and the ETSE logo. The top right corner displays the user information: "Usuari: Alberto Bermejo Mena" and "Centre: ABS Florida Nord". The main menu at the top includes links for "Localitzar Equip", "Averies", "Eines", "Manteniment", "Administració centre", "Ajuda", and "Pantalla principal". A "Tancar sessió" (Logout) link is also present. The central content area is titled "Administració del centre" and contains a form for managing users. The form fields include "Usuari: Alberto Apellido1 Apellido2" (with a dropdown arrow), a checked checkbox labeled "ACTIU", and two buttons: "Guardar" and "Sortir sense guardar".

Figura 4.17: Formulario para administrar los usuarios del centro.

The screenshot shows a Microsoft Internet Explorer window with the title "SGCE - Canvi d'Àmbit i Centre. - Microsoft Internet Explorer". The address bar shows the URL "http://192.168.0.151/CanviCentre.aspx". The page header includes the logo of the Department of Informatics (SAPLH - 2007) and the ETSE logo. The top right corner displays the user information: "Usuari: Alberto Bermejo Mena" and "Centre: ABS Florida Nord". The main menu at the top includes links for "Localitzar Equip", "Averies", "Eines", "Manteniment", "Administració centre", "Ajuda", and "Pantalla principal". A "Tancar sessió" (Logout) link is also present. The central content area is titled "Canvi de centre" and contains a form for changing location, SAP and ABS. The form fields include dropdown menus for "Àmbit: Costa de Ponent", "SAP: L'Hospitalet del Llobregat", and "Centre: ABS Amadeu Torner". At the bottom of the form are two buttons: "Acceptar" and "Cancelar".

Figura 4.18: Formulario para cambiar de Ámbito, SAP y/o ABS.

## Capítulo 5

# IMPLEMENTACIÓN Y PRUEBA

Se presentaran a continuación los detalles más técnicos del proceso de desarrollo del proyecto.

Primeramente se explicará la estructura del sitio Web creado, a continuación se definirán y explicaran detalles más concretos de la implementación aplicando las tecnologías explicadas en el capítulo 2. Finalmente, la parte de prueba del software.

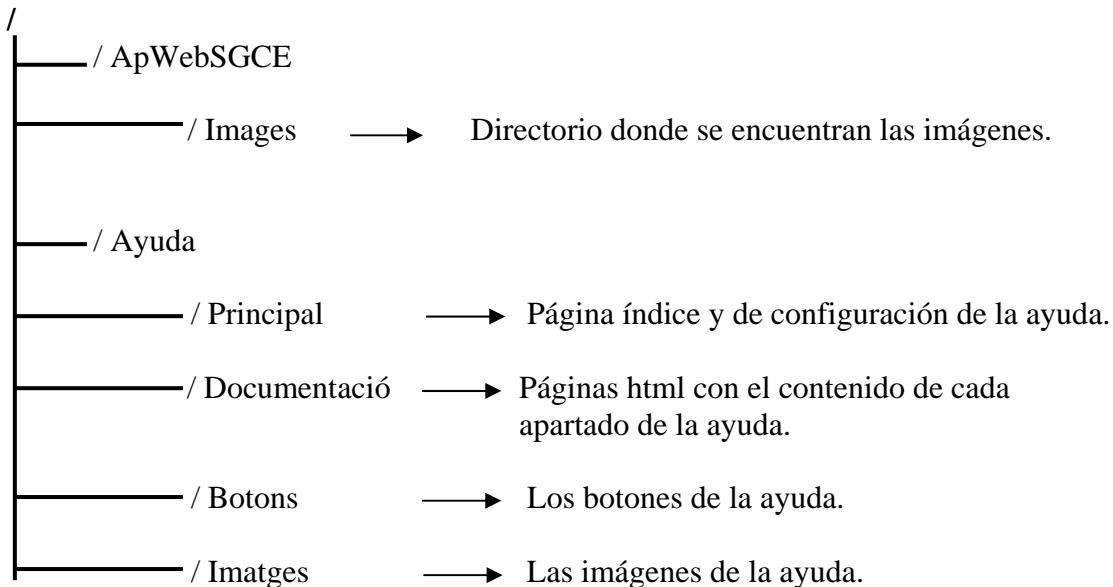
### 5.1 Implementación de la aplicación.

Dividiremos esta parte de la implementación en dos secciones, la primera la centraremos en la estructura del sitio Web diseñado mostrando y explicando cada parte en que se divide. A continuación, veremos los detalles de los objetos comerciales o de negocio necesarios para llevar a cabo las tareas definidas así como la capa de acceso a datos.

#### 5.1.1 Estructura del sitio Web.

Los ficheros que componen el sitio Web están separados en dos directorios, en ApWebSGCE tenemos los ficheros que componen la aplicación, los formularios Web y las imágenes; y en el directorio Ayuda tenemos los ficheros que componen la ayuda.

La estructura del sitio Web es:



### 5.1.2 Implementación de las clases de la lógica de negocio.

En el capítulo 2 se presentaron algunas de las principales características de la plataforma .NET. Además de las descritas, este framework ofrece a los desarrolladores la posibilidad de crear su propio espacio de nombres (*namespace*), podríamos decir que *namespace* es un conjunto de clases relacionadas y con un objetivo. Aprovechando esta característica se ha definido el espacio de nombres SGCEset, el conjunto de clases que representan los objetos comerciales, las entidades, de la capa de negocios y mantienen una relación estrecha con las tablas principales de la base de datos. SGCEset está formado por las clases Usuario, Centro, Averia y Equipo creando la estructura de las entidades necesarias para construir la lógica de negocio de la aplicación.

Se ha seguido el paradigma de Orientación a Objetos que es sin lugar a dudas el más utilizado por los desarrolladores para encarar la fabricación de aplicaciones de software, ya que permite representar de manera relativamente simple modelos y realidades muy complejas y esto hace que el software sea más fácil de programar, comprender y mantener. Por otro lado más de 20 años de investigación y desarrollo de este paradigma indican que está maduro para afrontar aplicaciones del mundo actual.

La implementación de las clases se ha realizado mediante el lenguaje de programación Visual C# .NET, un lenguaje de programación creado especialmente para .NET, sigue el paradigma de Orientación a Objetos y tiene una sintaxis derivada de C, C++, Delphi y Java.

Se han definido cuatro entidades principales para la aplicación: la clase Usuario, la clase Equipo, la clase Averia y la clase Centro.

La clase Usuario es la clase que define a un usuario del sistema para ello se ha dotado de los atributos necesarios para su especificación, se han declarado las propiedades para acceder a esos atributos, así como los métodos para crearlo, destruirlo e inicializarlo.

De cara a mantener consistencia en el código escrito se han llevado a seguido una nomenclatura para los atributos, las propiedades y los métodos definidos en cada clase. De este modo tenemos los siguientes prefijos:

Para los atributos:

- prv: indica ámbito privado
- Int: indica tipo entero, de 32 bits.
- Str: indica tipo string.
- Bol: indica tipo boolean.

Para las propiedades y métodos:

- p: indica propiedad.
- m: indica método.

Los atributos definidos son de ámbito privado a la clase y son los siguientes:

```
private Int32 prvIntIdCentro;
private string prvStrNombreCentro;
private Int32 prvIntIdABS;
private string prvStrNombreABS;
private Int32 prvIntIdUsuario;
private Int32 prvIntIdRol;
private string prvStrNombreRol;
private string prvStrDni;
private string prvStrNombre;
private string prvStrApellido1;
private string prvStrApellido2;
private Boolean prvBolActivo;
private Boolean prvBolPlantilla;
```

Las propiedades han sido definidas de ámbito público y de lectura / escritura, especificadas para acceder a los atributos de la clase; son las siguientes:

```
public string pDni
public Int32 pIdUsuario
public string pNombre
public string pApellido1
public string pApellido2
public Int32 pIdRol
public string pNombreRol
public Int32 pIdCentro
public string pNombreCentro
public Int32 pIdABS
public string pNombreABS
public Boolean pActivo
public Boolean pPlantilla
```

Como ejemplo detallaremos la definición de una propiedad en C#:

```
public string pDni
{
    get { return prvStrDni; }           // obtenemos el valor
    set { prvStrDni = value; }         // guardamos el valor
}
```

Los métodos definidos en esta clase son, el constructor por defecto, el destructor por defecto y un método para inicializar el objeto instanciado.

```
//Constructor por defecto
public Usuario() { }

//Destructor por defecto
~Usuario() { }

// Método mInicializar.
public Usuario mInicializar(Usuario MiUsuario, NpgsqlDataReader dr)
```

Este método recibe un objeto de la clase Usuario y un objeto NpgsqlDataReader con información de la base de datos, su función es establecer las propiedades del objeto con los datos del NpgsqlDataReader y devolver el objeto inicializado.

La clase Equipo define a un equipo en el sistema, se han determinado los atributos, las propiedades y métodos necesarios para su especificación; se verán a continuación:

Los atributos definidos:

```
private Int32 prvIntIdEquipo;
private Int32 prvIntIdModelo;
```

```
private string prvStrModelo;
private Int32 prvIntIdEstado;
private string prvStrEstado;
private Int32 prvIntIdHard;
private string prvStrHard;
private Int32 prvIntIdSoft;
private string prvStrSoft;
private Int32 prvIntFun;
private string prvStrFuncion;
private Int32 prvIntIdCentro;
private string prvStrCentro;
private Int32 prvIntIdABS;
private string prvStrABS;
private string prvStrUP;
private Int32 prvIntIdProyecto;
private string prvStrProyecto;
private string prvStrFechaProyecto;
private Int32 prvIntIdUbicacion;
private string prvStrIdUbicacion;
private string prvStrNumSerie;
private string prvStrEtiqueta;
private string prvStrNombre;
private string prvStrIP;
private Boolean prvBolActivo;
```

Las propiedades definidas:

```
public Int32 pIdEquipo
public Int32 pIdTipoHardware
public String pNombreTipoHardware
public Int32 pIdModelo
public String pNombreModelo
public Int32 pIdEstado
public String pNombreEstado
public Int32 pIdHard
public String pDescHard
public Int32 pIdSoft
public String pDescSoft
public Int32 pIdFun
public string pNombreFuncion
public Int32 pIdCentro
public string pNombreCentro
public Int32 pIdABS
public string pNombreABS
```

```

public string pUP
public Int32 pIdProyecto
public string pNombreProyecto
public String pFechaProyecto
public Int32 pIdUbicacion
public string pNombreUbicacion
public string pNumSerie
public string pEtiqueta
public string pNombre
public string pIP
public Boolean pActivo

```

Los métodos definidos son el constructor por defecto, el destructor por defecto y un método para inicializar el objeto.

```

//Constructor por defecto
public Equipo() { }

//Destructor por defecto
~Equipo() { }

// Método mInicializar.
public Equipo mInicializar(Equipo MiEquipo, NpgsqlDataReader dr)

```

El método mInicializar tiene la misma estructura y función que el método mInicializar de la clase Usuario; recibe un objeto Equipo y un NpgsqlDataReader y tiene como objetivo devolver el objeto recibido inicializado con los datos del NpgsqlDataReader.

La clase Averia define una avería de un equipo en el sistema y al igual que las dos clases arriba explicadas consta de atributos, propiedades y métodos; se verán a continuación:

Los atributos que componen esta clase son:

```

private Int32 prvIntIdAveria;
private Int32 prvIntIdCodAve;
private Int32 prvIntIdResolucion;
private Int32 prvIntIdUsuario;
private Int32 prvIntIdEquipo;
private string prvStrNumAveria;
private string prvStrF_INI;
private string prvStrF_FIN;
private Boolean prvBolActivo;

```

Las propiedades que la componen son:

---

```

public Int32 pIdAveria
public Int32 pIdCodigoAveria
public Int32 pIdResolucion
public Int32 pIdUsuario
public Int32 pIdEquipo
public String pNumAveria
public String pFechaInicial
public String pFechaFinal
public Boolean pActivo

```

Los métodos son el constructor por defecto, el destructor por defecto y el de inicializar el objeto instanciado:

```

//Constructor por defecto.
public Averia() { }

//Destructor por defecto.
~Averia() { }

// mIniciar.
public Averia mIniciar(Averia MiAveria, NpgsqlDataReader dr)

```

El método `mIniciar` desempeña la misma función que los métodos homólogos en las clases antes mencionadas.

La clase Centro define un centro en el sistema y de la misma manera que las clases arriba especificadas posee atributos, propiedades y métodos que se verán a continuación.

Los atributos de esta clase son:

```

private Int32 prvIntIdCentro;
private string prvStrNombre;
private Int32 prvIntIdAmbito;
private string prvStrAmbito;
private Int32 prvIntIdsAP;
private string prvStrSAP;
private Int32 prvIntIdABS;
private string prvStrABS;
private Int32 prvIntUpABS;
private string prvStrUP;
private string prvStrAbreviatura;

```

```

private string prvStrTelefono;
private string prvStrFax;
private string prvStrDireccion;
private string prvStrPoblacion;
private Int32 prvIntCP;
private string prvStrProvincia;
private Boolean prvBolActivo;

```

Las propiedades de esta clase son:

```

public Int32 pIdCentro
public Int32 pIdAmbito
public string pNombreAmbito
public Int32 pIdsAP
public string pNombreSAP
public Int32 pIdABS
public string pNombreABS
public Int32 pIdUP
public string pNumUP
public string pAbreviatura
public string pNombre
public string pTelefono
public string pFax
public string pDireccion
public string pPoblacion
public Int32 pCP
public string pProvincia
public Boolean pActivo

```

Los métodos de esta clase son:

```

//Constructor por defecto
public Centro() { }

//Destructor
~Centro() { }

//mInicializar
public Centro mInicializar(Centro MiCentro, NpgsqlDataReader dr)

```

El método `mInicializar` es similar a los arriba descritos.

En las figura 5.1 y 5.2 podemos ver a las cuatro clases que forman el namespace SGCEset. Se puede observar los atributos, las propiedades y los métodos que implementan cada una.

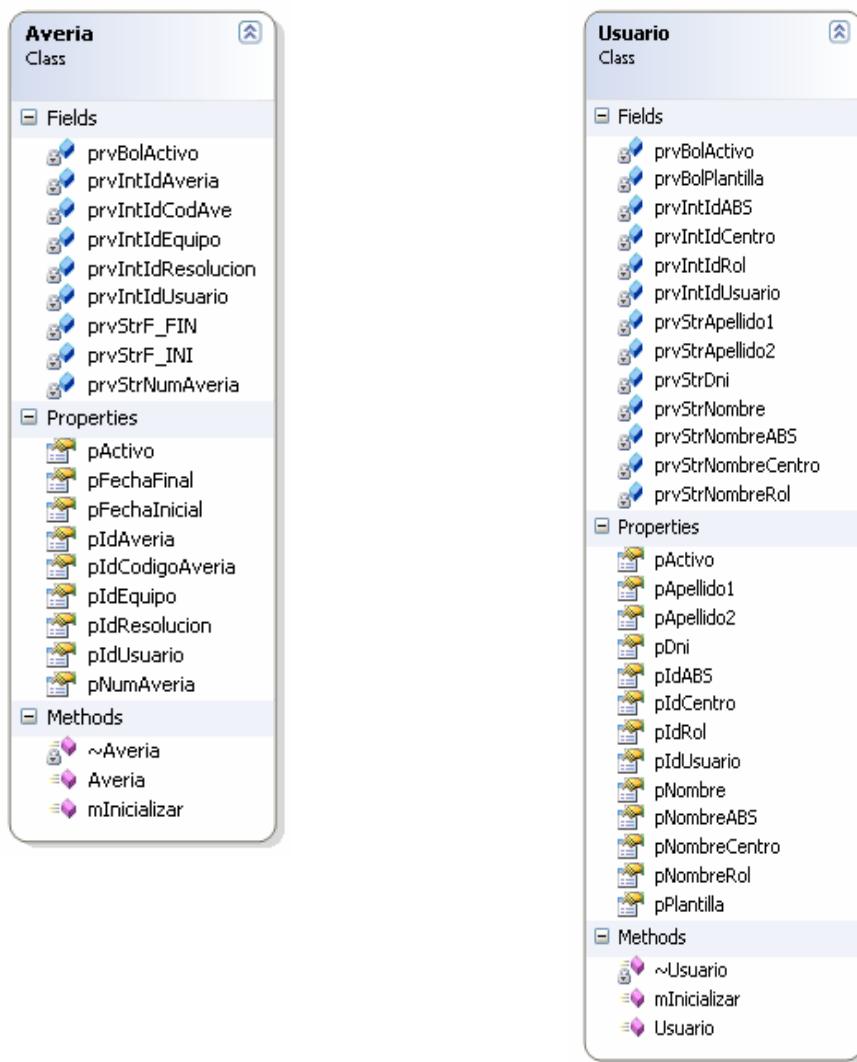


Figura 5.1: Las clases Averia y Usuario que componen el namespace SGCEset

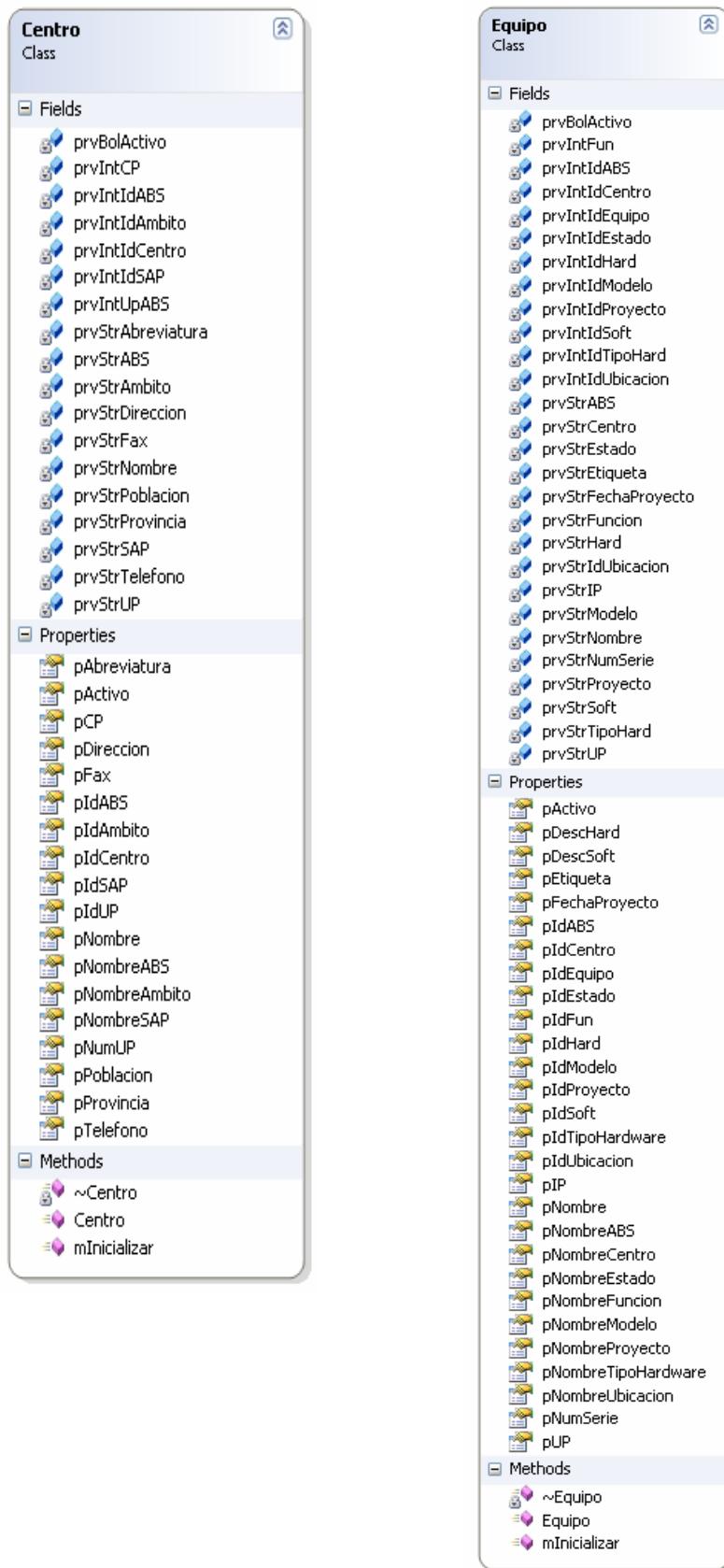


Figura 5.2: Las clases Centro y Equipo que componen el namespace SGCEset

### 5.1.3 La implementación de la clase de la capa de acceso a datos.

ADO.NET permite definir escenarios con conexión permanente a la base de datos, y sin conexión permanente a la base de datos. Debido a la arquitectura de la aplicación basada en cliente-servidor se decidió que el escenario sin conexión permanente permitiría un mejor uso de los recursos de los que se dispone, ofrecería una mejor utilización del ancho de banda de la red de telecomunicaciones del centro, un mejor rendimiento de la base de datos al no tener que soportar un número constante de conexiones... Así, cuando fuera necesaria información de la base de datos se abriría una conexión se obtendrían los datos o se almacenarían y se cerraría la conexión.

Una vez definido el escenario de trabajo, se decidió que era necesario especificar una clase que implementara todos los accesos necesarios a la base de datos, que estuvieran reunidos en una sola ubicación y no dispersos por la aplicación, que gestionara la conexión, esto es, abriendo y cerrando la conexión cuando ya se hubiera realizado la transacción. Esta clase se definió como la clase DAL (Data Access Layer, capa de acceso a datos), la cual gestiona la conexión y ofrece una interfaz a los objetos de la capa de negocios.

El diseño centralizado en esta clase de los accesos a la base de datos ofrece una serie de ventajas. La primera es que desde la aplicación solo hay un único punto de acceso posible a la base de datos, lo que reduce la posibilidad de accesos no autorizados. La segunda es que si no hay conexión o hay problemas solo es necesario mirar esta clase. La tercera es que se crea una manera uniforme de acceso a los datos. La cuarta un único repositorio de consultas SQL que permite reutilización mediante la invocación del método

La clase DAL pertenece al espacio de nombres BaseDatos, este espacio de nombre ha sido definido por si en un futuro se añade una clase que permita acceder a la base de datos Oracle.

DAL está compuesta por un atributo privado, *conn*, la cadena de conexión y por los métodos que permiten el acceso a la base de datos. Estos métodos componen la interfaz de la capa de acceso a datos. Se verá a continuación con más detalle la especificación de la clase:

La clase consta de un atributo:

```
NpgsqlConnection conn //La cadena de conexión.
```

Los métodos que componen esta clase se detallan a continuación:

```
//Metodo mValidarUsuario devolverá un registro con la información del usuario.
public NpgsqlDataReader mValidarUsuario(string login, string password)

//Método mBuscarEquipoNIE nos devolverá un registro con la información del
//equipo.
public NpgsqlDataReader mBuscarEquipoNIE(string etiqueta, int IDabs)
```

---

```
//Método mBuscarEquipoNSE devolverá un registro con la información del equipo.  
public NpgsqlDataReader mBuscarEquipoNSE(string numserie, int IDabs)  
  
//Método mObtenerModelo devolverá un registro con la información del modelo.  
public NpgsqlDataReader mObtenerModelo(int IDmodelo)  
  
//Método mObtenerFuncion devolverá un registro con la información de la  
//función.  
public NpgsqlDataReader mObtenerFuncion(int IDfuncion)  
  
//Método mObtenerHardware devolverá un registro con la información del  
//software.  
public NpgsqlDataReader mObtenerHardware(int IDhardware)  
  
//Método mObtenerSoftware devolverá un registro con la información del  
//software.  
public NpgsqlDataReader mObtenerSoftware(int IDsoftware)  
  
//Método mObtenerEstado devolverá un registro con la información del estado.  
public NpgsqlDataReader mObtenerEstado(int IDEstado)  
  
//Método mObtenerUbicacion devolverá un registro con la información de la  
//ubicación dentro del centro.  
public NpgsqlDataReader mObtenerUbicacion(int IDubicacion, int IDabs)  
  
//Método mObtenerProyecto devolverá un registro con la información del  
//proyecto.  
public NpgsqlDataReader mObtenerProyecto(int IDproyecto)  
  
//Método mObtenerTipoHardware devolverá un registro con la información del  
//tipo de hardware.  
public NpgsqlDataReader mObtenerTipoHardware(int IDtipohard)  
  
//Método mObtenerResolucion devolverá un registro con la información de la  
//resolución.  
public NpgsqlDataReader mObtenerResolucion(int IDresolucion)  
  
//Método mObtenerCodigoResolucionTipoHard devolverá los códigos de resolución  
//relacionados con un tipo de hardware.  
public NpgsqlDataReader mObtenerCodigoResolucionTipoHard(int IDTipohard)  
  
//Método mObtenerCodigoAveria devolverá un registro con la información del  
//código de avería.  
public NpgsqlDataReader mObtenerCodigoAveria(int IDCodave)  
  
//Método mObtenerCodigoAveriaTipoHard devolverá los códigos de avería  
//relacionados con un tipo de hardware.  
public NpgsqlDataReader mObtenerCodigoAveriaTipoHard( int IDTipohard)  
  
//Método mObtenerAveria devolverá un registro con la información de la avería.  
public NpgsqlDataReader mObtenerAveria(int IDaveria)
```

---

```
//Método que recuperara las averías cerradas.  
public DataSet mRecuperarAveriasCerradas(int IDabs)  
  
//Método que recuperara las averías abiertas.  
public DataSet mRecuperarAveriasAbiertas(int IDabs)  
  
//Método que recuperara las todas las averías.  
public DataSet mRecuperarAveriasTodas(int IDabs)  
  
//Método que recuperara los usuarios activos.  
public DataSet mRecuperarUsuariosActivos(int IDabs, int IDUsuario)  
  
//Método que recuperara los usuarios activos.  
public DataSet mRecuperarUsuariosNoActivos(int IDabs, int IDUsuario)  
  
//Método que recuperara los usuarios activos.  
public DataSet mRecuperarUsuariosTotales(int IDabs, int IDUsuario)  
  
//Método que recuperara los Equipos activos.  
public DataSet mRecuperarEquiposActivos(int IDabs, int IDUsuario)  
  
//Método que recupera la información de una avería de un equipo.  
public NpgsqlDataReader mObtenerAveriaPorIdEquipo(int ID_Equipo)  
  
//Método mObtenerRol recupera la información del Rol.  
public NpgsqlDataReader mObtenerRol(int IDrol)  
  
//Método mObtenerUsuario recupera la información de un Usuario.  
public NpgsqlDataReader mObtenerUsuario(int IDusuario)  
  
//Método mObtenerUsu recupera la información de un Usuario independientemente  
//de su estado.  
public NpgsqlDataReader mObtenerUsu(int IDusuario)  
  
//Método mObtenerListaUsuarios reupera los usuarios de un ABS.  
public NpgsqlDataReader mObtenerListaUsuario(int IDabs)  
  
//Método mObtenerCentro recupera la información de un centro.  
public NpgsqlDataReader mObtenerCentro(int IDcentro)  
  
//Método mObtenerSAP devolverá un registro con la información del SAP.  
public NpgsqlDataReader mObtenerSAP(int IDsap)  
  
//Método mObtenerABS devolverá un registro con la información del ABS.  
public NpgsqlDataReader mObtenerABS(int IDabs)  
  
//Método mObtenerABSporIdSAP recupera las ABS de un SAP.  
public NpgsqlDataReader mObtenerABSporIdSAP(int IDsap)
```

---

```

//Método mObtenerSAPsPorAmbito devolverá un registro con todos los centros.
public NpgsqlDataReader mObtenerSAPsPorAmbito(int IDAmbito)

//Método mObtenerCentrosPorSAP devolverá un registro con todos los centros que
//pertenezcan a una SAP.
public NpgsqlDataReader mObtenerCentrosPorSAP(int IDsap)

//Método mObtenerCentro devolverá todos los centros.
public NpgsqlDataReader mObtenerCentros()

//Método mObtenerAmbito devolverá un registro con la información del Ámbito.
public NpgsqlDataReader mObtenerAmbito(int IDambito)

//Método que recupera el Ámbito mediante el SAP.
public NpgsqlDataReader mObtenerIdAmbitoPorIdSAP(int IdSAP)

//Método mObtenerEquipo devolverá un registro con la información del equipo.
public NpgsqlDataReader mObtenerEquipo(int IDEquipo)

//Método mObtenerNombresAmbito devolverá un registro con la información del
//Ámbito.
public NpgsqlDataReader mObtenerNombresAmbito()

//Método mInsertarAveria dará de alta un registro en la TBL_AVERIA y
//devolverá un entero indicando las filas afectadas.
public int mInsertarAveria(int IDCodave, int IDUsuario, int IDEquipo, string
NumAveria, string fecha)

//Método mInsertarNuevaContraseña cambia la contraseña de un usuario y
//devolverá un entero indicando las filas afectadas.
public int mInsertarNuevaContraseña(int IdUsuario, string NuevaContraseña)

//Método mSolucionarAveria y devolverá un entero indicando las filas
//afectadas.
public int mSolucionarAveria(int IDAveria, int IDResAve, string fecha)

// Actualizará la base de datos con el nuevo valor plantilla y devolverá un
//entero indicando las filas afectadas.
public int mActualizarUsuAdminCentro(int IDUsuario, Boolean plantilla)

//Insertará un Nuevo usuario en la base de datos.
public int mInsertarNuevoUsuario(int pIdABS, int pIdRol , string pDni, string
pNombre, string pApellido1, string pApellido2, string pLogin,
string pPassword, string pAntValidacion, string UltValidacion, bool
pActivo, bool pPlantilla )

//Añade un nuevo equipo al sistema.
public int mInsertarEquipoNuevo(int pIdModelo, int pIdEstado , int pIdHard ,
int IdSoft, int pIdFun, int pIdABS, int pIdProyecto, int pIdUbicacion ,
int pIdTipoHardware, string pNumSerie, string pEtiqueta, string pNombre,
string pIP, bool pActivo )

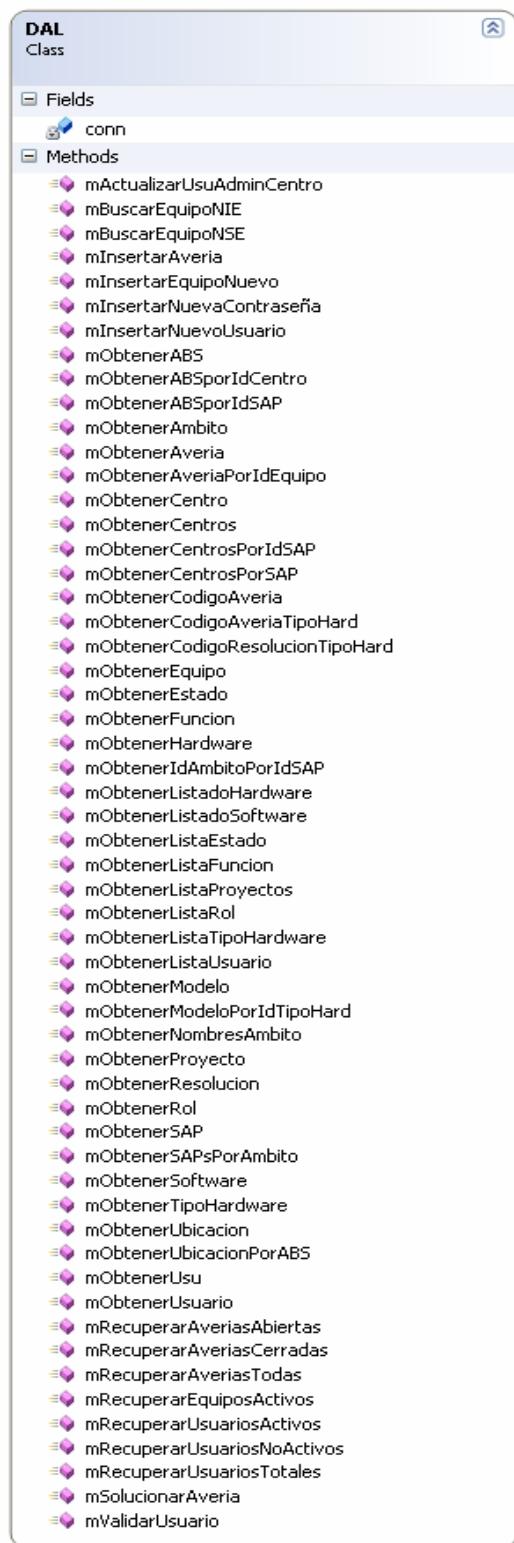
//Obtiene la lista de roles.
public NpgsqlDataReader mObtenerListaRol()

```

---

```
//Obtiene una lista con los tipo de hardware.  
public NpgsqlDataReader mObtenerListaTipoHardware()  
  
//Obtiene una lista con los proyectos.  
public NpgsqlDataReader mObtenerListaProyectos()  
  
//Obtiene una lista con los estados.  
public NpgsqlDataReader mObtenerListaEstado()  
  
//Obtiene una lista con la configuración software dependiendo del tipo de  
//hardware.  
public NpgsqlDataReader mObtenerListadoSoftware(int IdTipoHard)  
  
//Obtiene una lista con la configuración hardware dependiendo del tipo de  
hardware.  
public NpgsqlDataReader mObtenerListadoHardware(int IdTipoHard)  
  
//Obtiene una lista con la función  
public NpgsqlDataReader mObtenerListaFuncion()  
  
//Obtiene los modelos de equipos dependiendo del tipo de hardware.  
public NpgsqlDataReader mObtenerModeloPorIdTipoHard(int IdTipoHard)
```

En la figura 5.3 vemos una representación de la clase DAL donde apreciamos el atributo **conn** y los métodos arriba detallados:



**Figura 5.3: La clase de acceso a datos (DAL).**

## 5.2 Prueba

El conjunto de pruebas realizadas a lo largo del desarrollo de la aplicación se podrían clasificar en las pruebas realizadas por mí como, desarrollador de software, y por las realizadas por usuarios independientes, pertenecientes al departamento de informática.

Se han llevado a cabo pruebas de validación de cada funcionalidad especificada que demuestran que el sistema cumple con sus requisitos. También se han llevado a cabo pruebas funcionales o de caja negra estudiando las entradas proporcionadas y las salidas obtenidas. Así como pruebas de componentes y pruebas de integración comprobando el correcto funcionamiento del sistema global siguiendo las especificaciones.

De los diagramas de flujo presentados en el capítulo 4, centrado en el diseño, se obtienen para cada funcionalidad el flujo básico de eventos, el flujo normal y correcto, pero de estos diagramas se extraen flujos alternativos formados por los casos excepcionales.

El conjunto de pruebas realizadas están basadas en casos de uso, se han definido unos escenarios a partir de estos casos de uso, como ejemplo a las pruebas realizadas tenemos:

Caso de uso: búsqueda de equipo.

**Camino básico:** Visualización de los datos del equipo.

**Precondiciones:** El usuario se encuentra en la pantalla principal de la aplicación.

- 1) **Iniciamos la búsqueda:** El usuario accede al menú de localizar un equipo.
- 2) **Mostrar formulario:** El sistema muestra el formulario indicando las diferentes opciones de búsqueda.
- 3) **Introducción de datos:** El usuario introduce el NSE o NIE.
- 4) **Muestra formulario con datos del equipo:** El sistema muestra los datos del equipo a localizar.

**Poscondiciones:** el caso de uso finaliza cuando el usuario vuelve al menú principal o al formulario de “Alta d’incidencia”.

### Flujos alternativos:

- 1) NIE incorrecto.
- 2) NSE incorrecto.

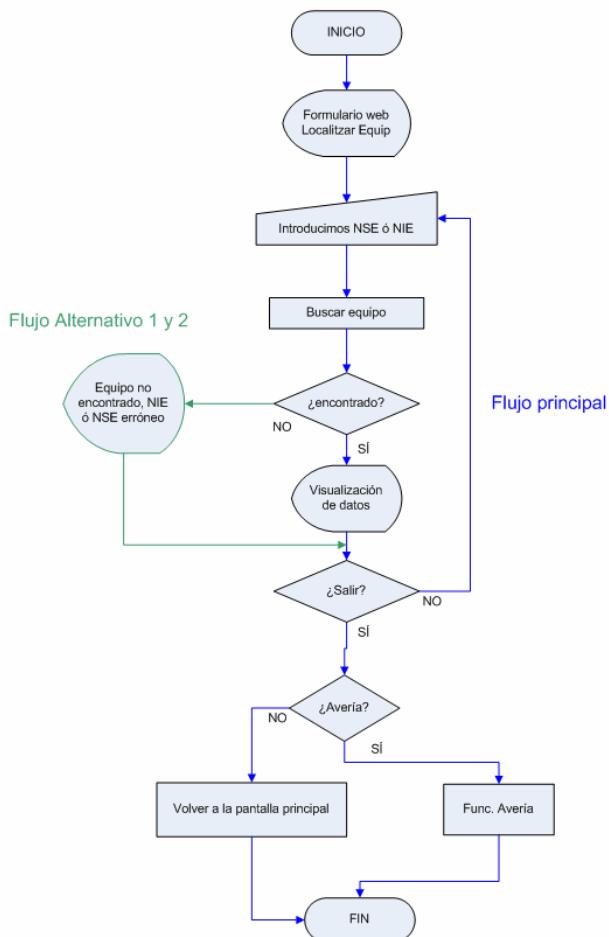
Podemos derivar estos escenarios:

**Escenario 1:** Obtención de los datos correctamente.

**Escenario 2:** NIE erróneo, no hay datos del equipo.

**Escenario 3:** NSE erróneo, no hay datos del equipo.

Escenario 1	Flujo básico		
Escenario 2	Flujo básico	F. alternativo 1	
Escenario 3	Flujo básico	F. alternativo 1	F. alternativo 2



**Figura 5.4 Representación de escenarios.**

El conjunto de pruebas realizadas durante el proceso de desarrollo del proyecto ha permitido detectar y solucionar un conjunto de fallas del sistema. El resultado de las

pruebas permite demostrar el correcto funcionamiento de la aplicación cumpliendo con las especificaciones definidas.

## Capítulo 6

# MANUAL DE USUARIO

En este capítulo veremos la guía del usuario, redactada para facilitar el aprendizaje y ayudar al usuario de la aplicación cuando quiera realizar alguna tarea en el sistema.

Como primera parte encontramos cómo acceder al sistema y validarnos ante él. A continuación, nos centraremos en la funcionalidad para localizar un equipo, parte fundamental para poder iniciar una incidencia. Seguidamente, tenemos la gestión de una incidencia, desde su alta hasta la conclusión. Una vez vistas estas funcionalidades describiremos la parte de la aplicación centrada en la gestión de los propios usuarios de la aplicación y solo accesible a los responsables de cada centro. Para terminar veremos cómo cambiar de centro, SAP o Ámbito, funcionalidad restringida a los informáticos.

### 6.1 Introducción

El *Sistema de Gestió i Control d'Equips informàtics*, en adelante SGCE, se ha desarrollado para facilitar la gestión y el control del parque informático de un CAP, de un ABS, de un SAP, y/o de un Ámbito.

La gestión de un parque informático está formado un conjunto de tareas muy variadas y a la vez complementarias, por estos motivos, se han separado en otros más simples para facilitar el cumplimiento de los objetivos.

## 6.2 Entrada a la aplicación

Esta es la pantalla de entrada a la aplicación,



Figura 6.1: Entrada a la aplicación.

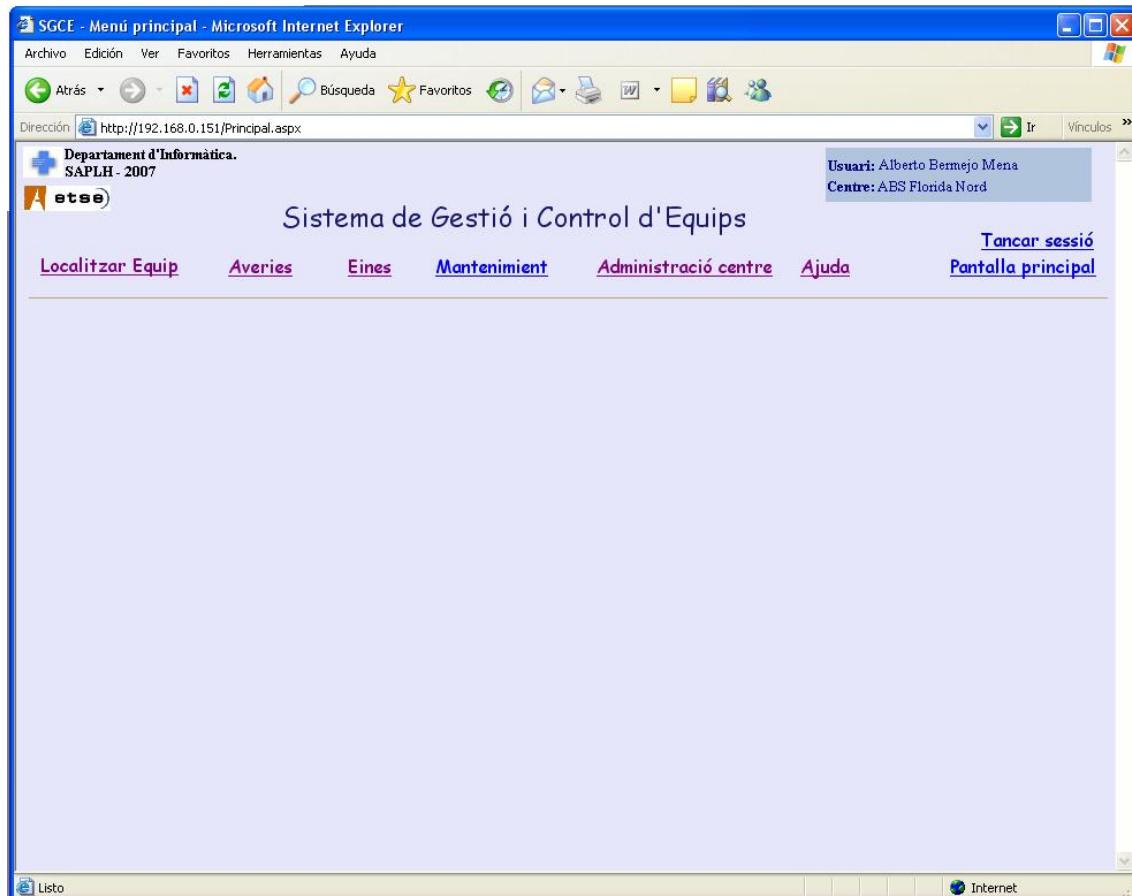
Para poder acceder a las funcionalidades del sistema los usuarios se tienen que autenticar i validar mediante el login de usuario y password.

Una vez el usuario ha introducido los datos en los cuadros de texto correspondientes se debe clicar al botó **Iniciar sessió**, para iniciar sesión en el sistema.

Si en el proceso de autentificación se han introducido datos incorrectos, entonces, el sistema muestra un mensaje donde informa del error al usuario.

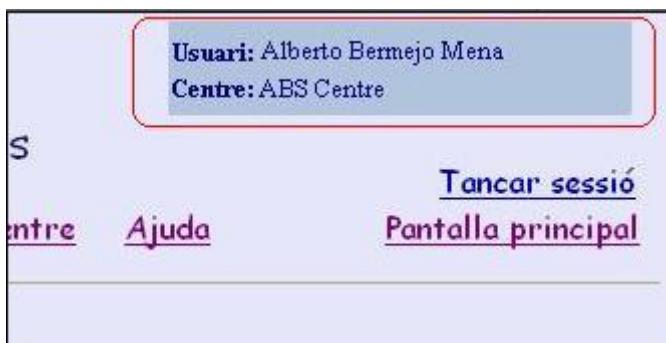
Si un usuario no tiene ni login ni password o tiene problemas para acceder al sistema, tendrá que ponerse en contacto con el Departament d'Informàtica.

Una vez el usuario se haya autentificado y validado correctamente aparecerá la pantalla principal donde el sistema muestra las funcionalidades principales, como por ejemplo, Localitzar Equip, Averies, Ajuda, ...



**Figura 6.2: Pantalla principal**

La pantalla principal muestra información del usuario validado y a qué centro pertenece, esta información está situada en la parte superior derecha, como se puede ver en la siguiente figura:



**Figura 6.3: Información del usuario validado.**

Si un usuario tiene que cambiar la contraseña, tiene que ir al menú *Eines* y escoger la opción cambio de contraseña.

Aparecerá un formulario para cambiar la contraseña:

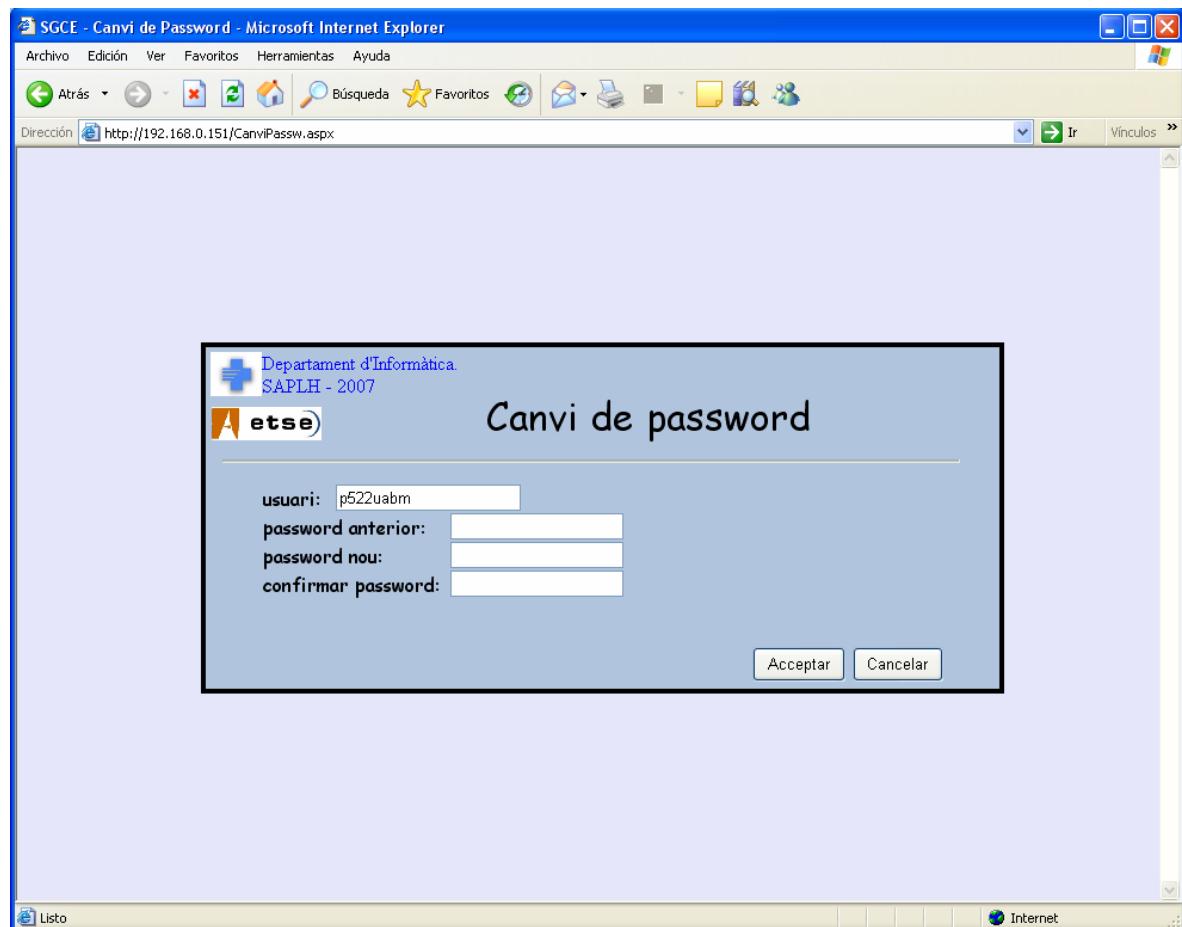
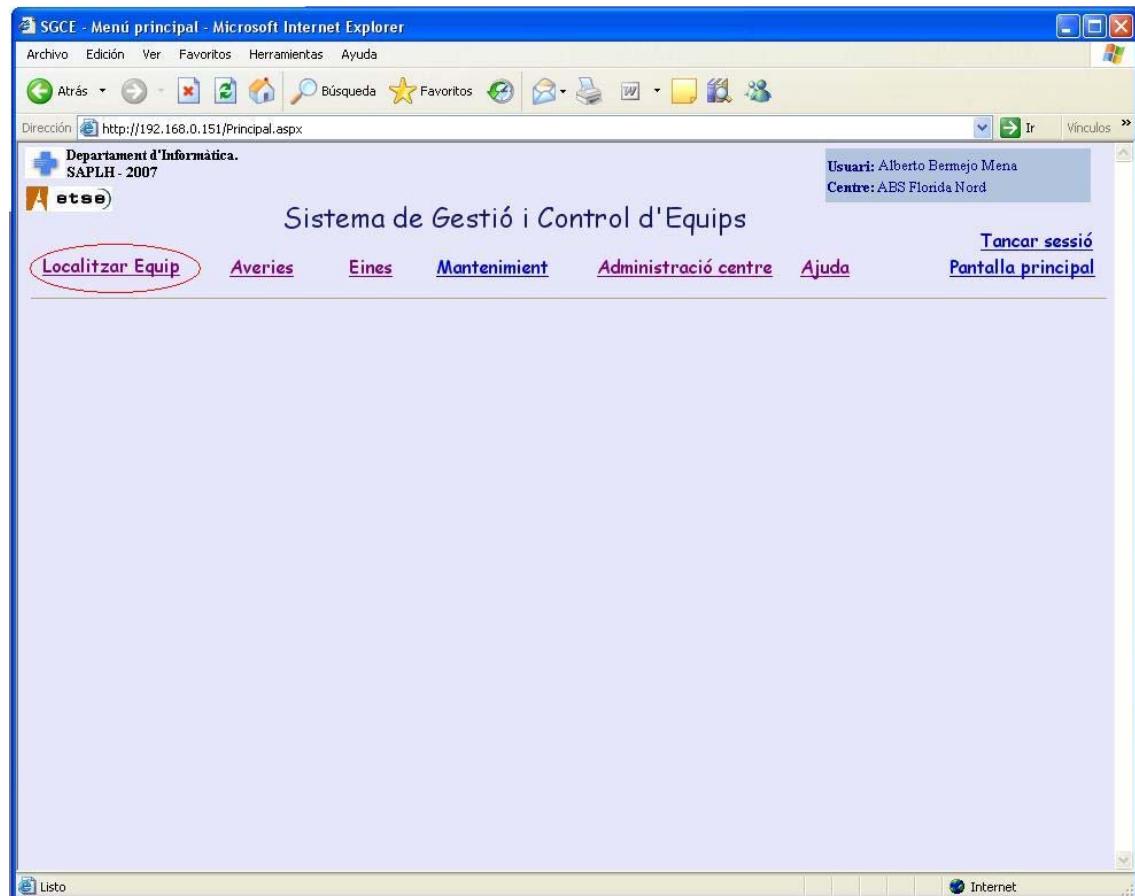


Figura 6.4: Formulario de cambio de contraseña.

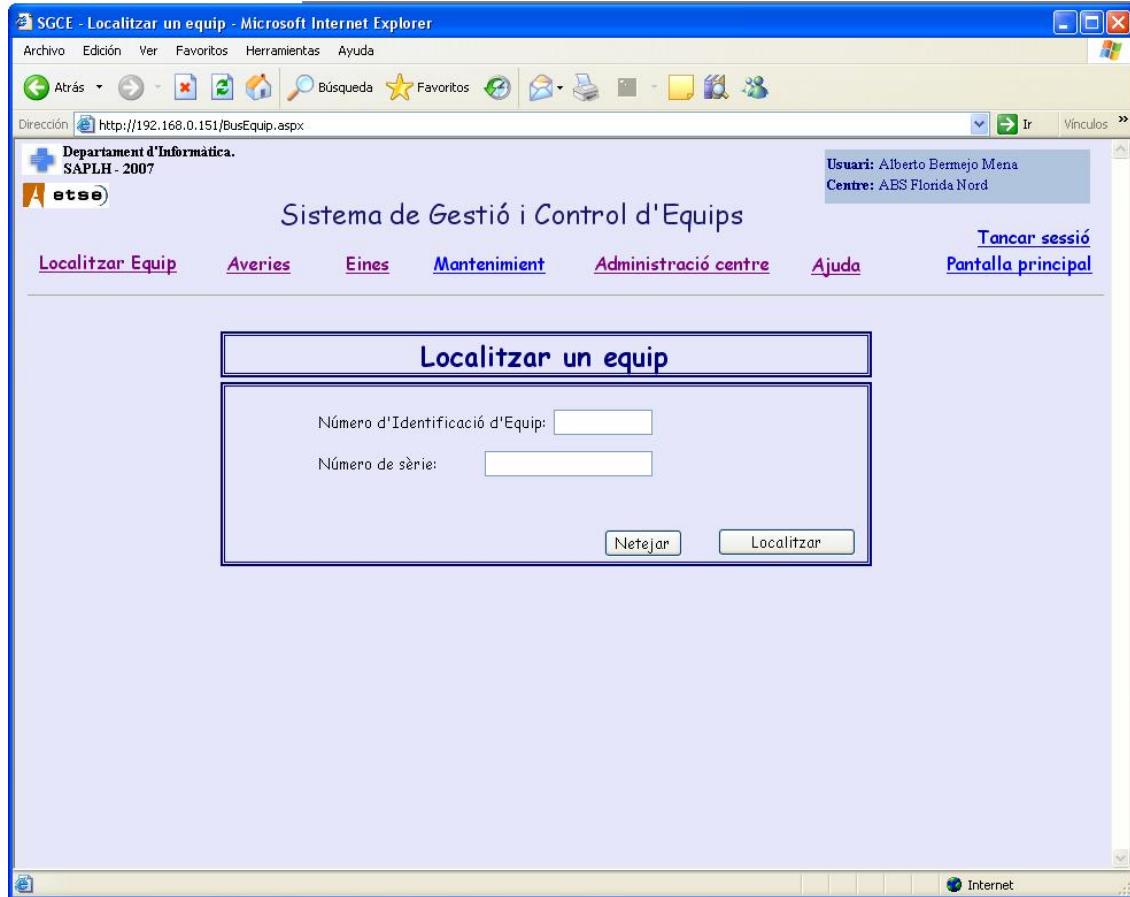
### 6.3 Localizar un equipo

Esta funcionalidad permite encontrar un equipo en el centro, se accede mediante la opción *Localitzar Equip* del menú principal [Localitzar Equip](#), como se puede ver en la siguiente figura:



**Figura 6.5:** Selección de la funcionalidad Localitzar Equip en el menú principal.

A continuación aparecerá el formulario *Localitzar un Equip*:



**Figura 6.6: Pantalla Localitzar Equip.**

En esta pantalla se puede localizar un equipo introduciendo el Número de Identificación de Equipo (en adelante NIE) ó el Número de Serie de Equipo (en adelante NSE) en los cuadros de texto correspondientes. Mediante el botón Netejar  se puede borrar el contenido de los cuadros de texto para poder realizar otra consulta.

a) Localizar un equipo introduciendo el NIE:

Recordamos que el NIE se puede encontrar en una etiqueta enganchada al equipo. Se debe llenar el cuadro de texto con los datos de la etiqueta y apretar el botón localizar,

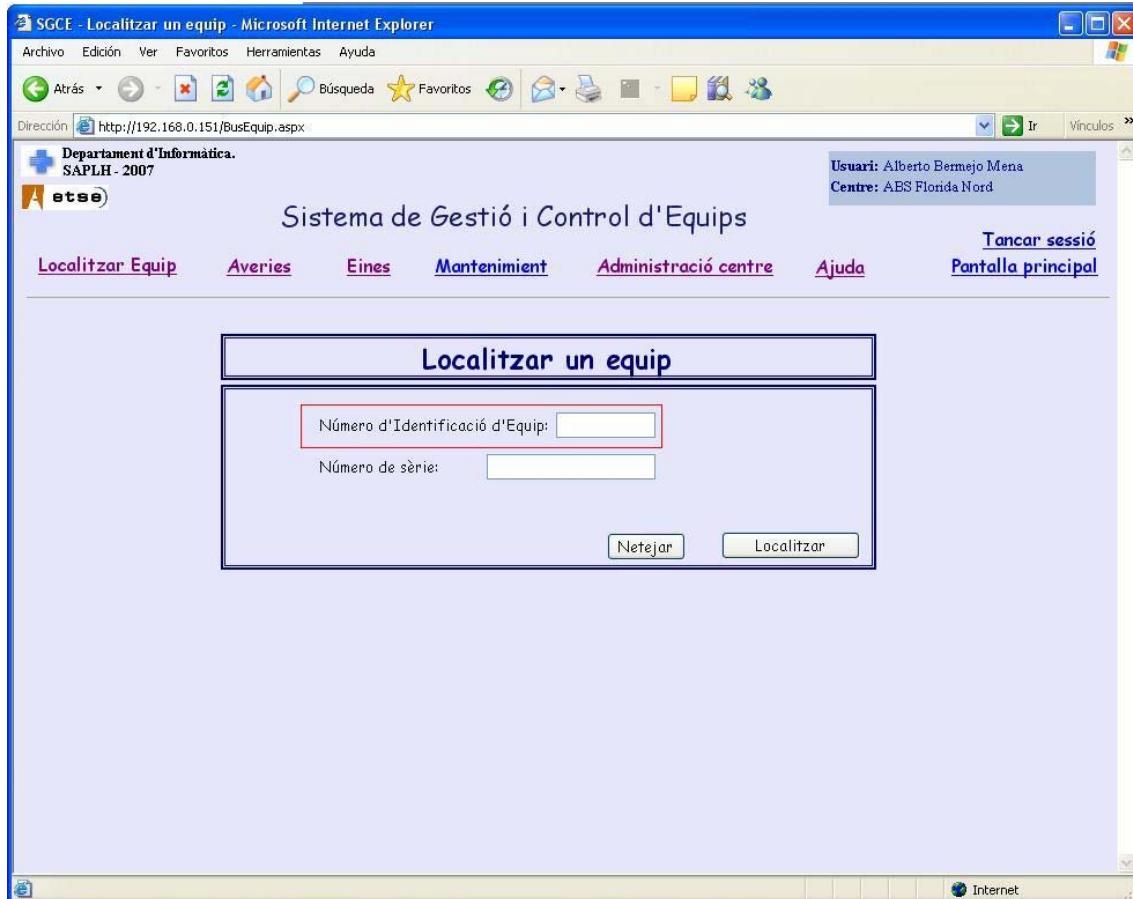


Figura 6.7: Localizar equipo por NIE.

b) Localizar un equipo introduciendo el NSE:

Esta secuencia alfanumérica se puede encontrar en la parte posterior del equipo ó, como antes, en una etiqueta en el equipo.

Se debe llenar el cuadro de texto con los datos y se clica a continuación en el botón localizar **Localizar**, como antes.

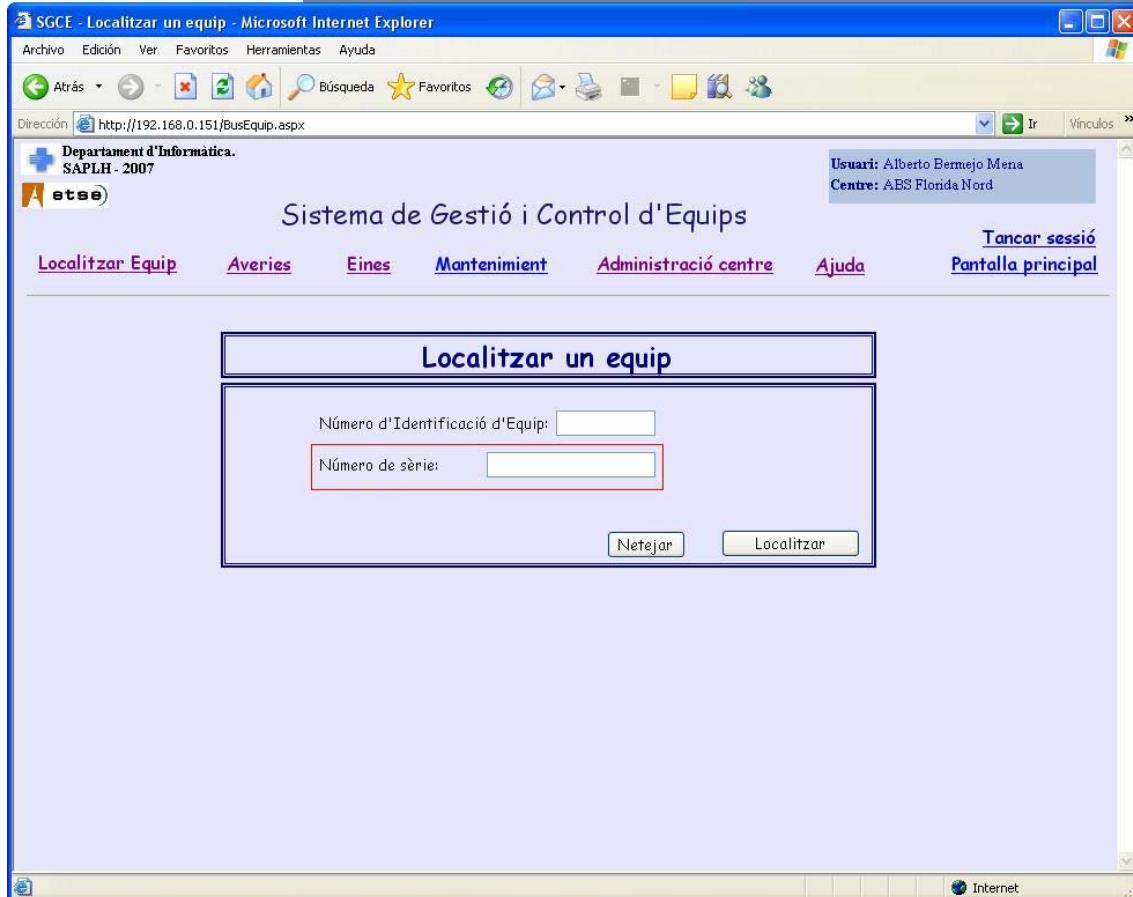


Figura 6.8: Localizar equipo por NSE.

Si los datos introducidos corresponden a un equipo del centro aparecerá una pantalla (ver figura 6.9) donde se verán los datos descriptivos de este equipo, en caso contrario, el sistema nos informará del error con un mensaje (ver figura 6.10).

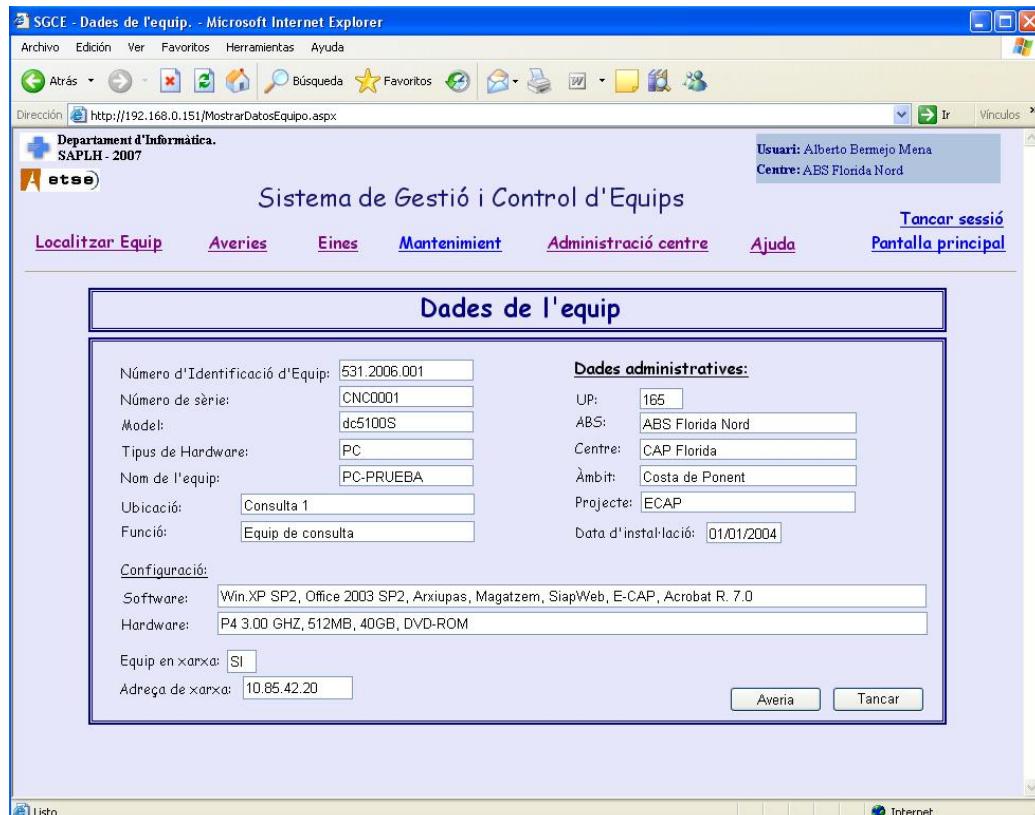


Figura 6.9: Visualización de los datos del equipo.

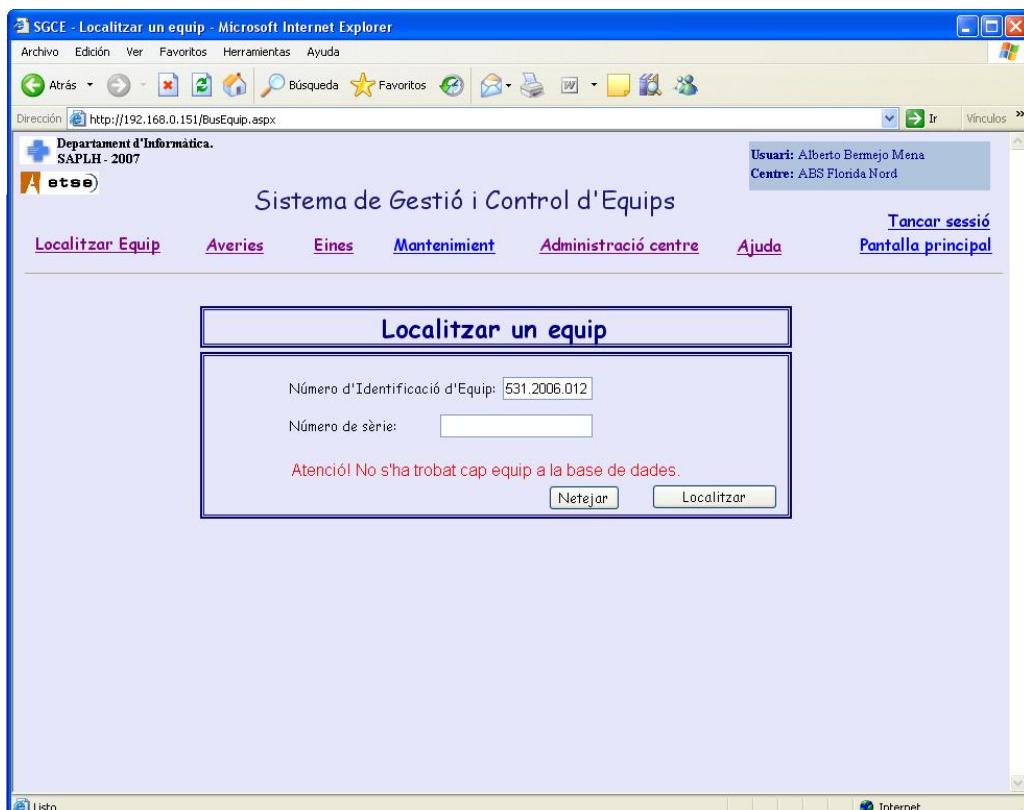


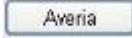
Figura 6.10: Mensaje de error a la hora de localizar un equipo.

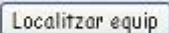
## 6.4 Gestión de una incidencia

La gestión de una incidencia en el sistema está formada por diferentes partes, empieza con la localización de un equipo, el alta de la incidencia en la base de datos y finaliza cuando se archiva una vez resuelta.

Explicaremos con más detenimiento este proceso, la gestión de una incidencia la podemos dividir en:

- 1) La localización del equipo.**
- 2) El alta de la incidencia en el sistema.**
- 3) Resolución de la incidencia.**

Pera acceder a este formulario, el usuario puede elegir del menú principal *Localitzar Equip* ó el botón Avería  del formulario *Mostrar Dades de l'Equip*.

Si el usuario escoge la primera opción, el formulario de gestión de una incidencia aparecerá vacío y tendrá que apretar el botón localizar equipo  para poder localizar el equipo y dar de alta la incidencia.

**1) La localización del equipo** está explicada en “*Localitzar un equip*” 

### 2) El alta de la incidencia en el sistema:

Una vez tenemos relleno el formulario con los datos del equipo, se tiene que introducir el número de avería que será proporcionado por la empresa de reparación de equipos y seleccionar el código de avería o la descripción de la avería que más se ajusta, lo vemos en la siguiente figura:

**Figura 6.11:** Datos necesarios para dar de alta la incidencia.

Los controles desplegables *Codi d'averia* i *Descripció d'averia* facilitaran la tarea de seleccionar la descripción más adecuada de la incidencia (ver figura 6.12 y 6.13).



**Figura 6.12:** Desplegable Codi d'averia

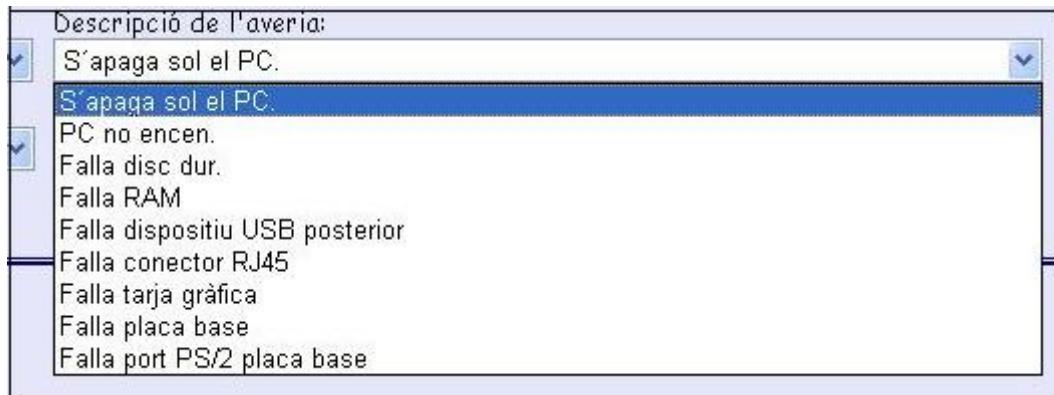


Figura 6.13: Desplegable Descripción de la avería.

Una vez se haya seleccionado el código de avería (y/o la descripción de la avería) e introducido el número de avería, se ha de clicar en el botón guardar **Guardar**, para dar de alta la incidencia en el sistema. Para cancelar el alta de la incidencia se tiene que apretar el botón *sortir sense guardar* **Sortir sense guardar**.

### 3) Resolución de la incidencia:

Antes de nada se tiene que localizar el equipo con la incidencia y después elegir un código de resolución que sea el más aproximado a la actuación del técnico.



Figura 6.14: Selección del código de resolución.

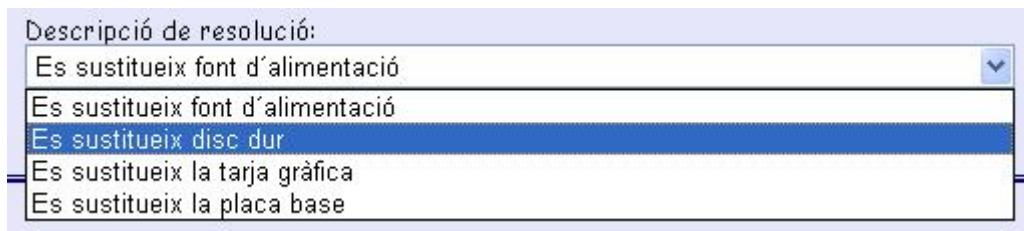


Figura 6.15: Selección de la descripción de la resolución.

Una vez escogida la resolución correcta se tiene que clicar en el botón **Solucionar i arxivar**  , para cerrar y archivar la incidencia.

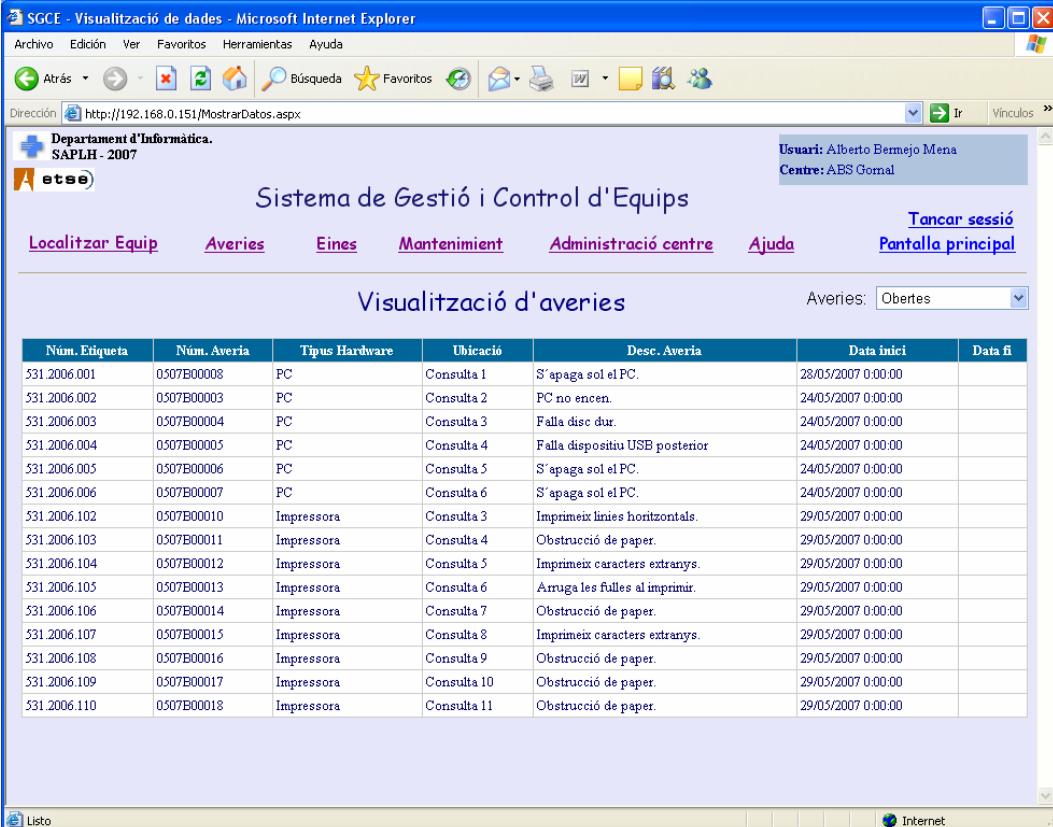
## 6.5 Visualización de datos

Mediante el menú *Eines* puede visualizar datos.

Los datos que puede visualizar son:

- 1) Las averías abiertas, cerradas y totales,**
- 2) Los equipos del centro,**
- 3) Los usuarios del centro (si dispone de permiso)**

1) Visualizar averías:



The screenshot shows a Microsoft Internet Explorer window displaying the SGCE system interface. The title bar reads "SGCE - Visualització de dades - Microsoft Internet Explorer". The address bar shows the URL "http://192.168.0.151/MostrarDatos.aspx". The page header includes the department "Departament d'Informàtica SAPLH - 2007" and user information "Usuari: Alberto Bermejo Mena Centre: ABS Gornal". The main menu at the top includes "Localitzar Equip", "Averies", "Eines", "Manteniment", "Administració centre", "Ajuda", and "Pantalla principal". A sub-menu titled "Visualització d'averies" is currently selected. A dropdown menu labeled "Averies:" contains the option "Obertes". The main content area displays a table of open faults (Averies) with the following columns: Núm. Etiqueta, Núm. Averia, Tipus Hardware, Ubicació, Desc. Averia, Data inici, and Data fi. The table lists 18 entries, each corresponding to a different fault description and location. At the bottom of the table, there is a link "Averies: Obertes" which likely links back to the current view.

Figura 6.16: Visualización de averías.

Mediante el control selección de estado  puede cambiar los datos a visualizar:

- 2) Visualización de equipos

Tipus Hardware	Model	Núm. Etiqueta	Núm. Serie	Ubicació
Impressora	Laserjet 1150	531.2006.100	VNC1000	Consulta 1
Impressora	Laserjet 1150	531.2006.101	VNC1001	Consulta 2
Impressora	Laserjet 1150	531.2006.102	VNC1002	Consulta 3
Impressora	Laserjet 1150	531.2006.103	VNC1003	Consulta 4
Impressora	Laserjet 1150	531.2006.104	VNC1004	Consulta 5
Impressora	Laserjet 1150	531.2006.105	VNC1005	Consulta 6
Impressora	Laserjet 1150	531.2006.106	VNC1006	Consulta 7
Impressora	Laserjet 1150	531.2006.107	VNC1007	Consulta 8
Impressora	Laserjet 1150	531.2006.108	VNC1008	Consulta 9
Impressora	Laserjet 1150	531.2006.109	VNC1009	Consulta 10
Impressora	Laserjet 1150	531.2006.110	VNC1010	Consulta 11
Impressora	Laserjet 1150	531.2006.111	VNC1011	Consulta 12
Impressora	Laserjet 1150	531.2006.112	VNC1012	Consulta 13
Impressora	Laserjet 1150	531.2006.113	VNC1013	Consulta 14
Impressora	Laserjet 1150	531.2006.114	VNC1014	Consulta 15

Figura 6.17: Pantalla visualizar equipos.

## 3) Visualización de usuarios.

Login	Nom	Primer cognom	Segon cognom	Actiu
javier	Javier	Serrano	Garcia	<input checked="" type="checkbox"/>
p522uabm	Alberto	Bermejo	Mena	<input checked="" type="checkbox"/>
p522uapp	Alberto	Apellido1	Apellido2	<input checked="" type="checkbox"/>
p522urgn	Rocio	Garcia	Naranjo	<input checked="" type="checkbox"/>

Figura 6.18: Visualización de usuarios.

Mediante el control de situación de usuarios  puede cambiar los datos a visualizar.

## 6.6 Administración del centro

La finalidad de esta funcionalidad es poder facilitar la tarea de selección de usuarios que realizaran la gestión de incidencias, de esta manera, la supervisión y la responsabilidad recaen en personas concretas mejorando en eficacia.



**Figura 6.19:** Formulario Administración del centro.

Mediante el desplegable se puede visualizar los usuarios del centro.

La operativa a seguir será:

1. Escoger un usuario:
2. Seleccionar si esta activo ó no:  ACTIU
3. Clicar en el botón guardar:

\*Los cambios es guardaran de forma inmediata a la base de datos.

De esta manera el responsable del centro seleccionará las personas encargadas de la gestión de les incidencias.

El botón **Visualització d'usuaris**, permite obtener un listado con todos los usuarios del centro, permite tener una visión global de la función de cada usuario.

Para salir del formulario podemos clicar en el botón **sortir sense guardar**, el usuario puede escoger otra funcionalidad del menú principal.

*\*Salir sin guardar solo afecta a la última modificación no guardada previamente.*

## 6.7 Cambio de Àmbito, SAP o Centro

Permite a un usuario con un rol elevado poder cambiar de Àmbit y/o de Centre para poder gestionar el equipos de más de un centro, este sería el caso de un usuario que gestione una SAP.

Se puede acceder al formulario mediante el menú principal *Eines / Canvi Centre*:

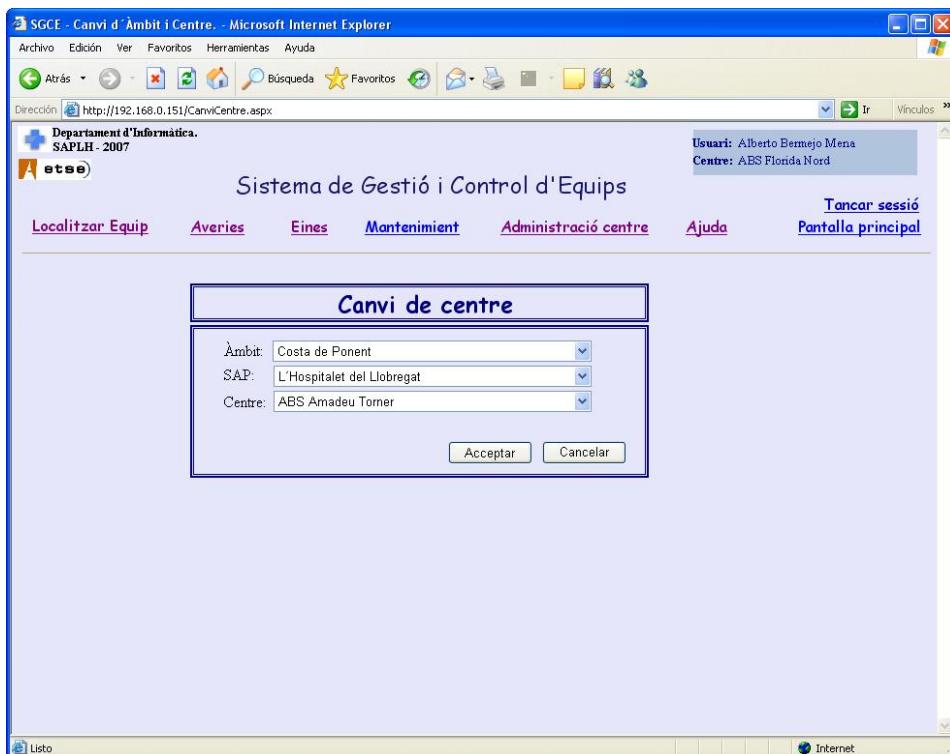


Figura 6.20: Formulari Canvi de centre.

El desplegable **Àmbit** **Ambit: Barcelona**, permite cambiar de Àmbito y una vez escogido el Àmbit, con el desplegable **SAP**, **SAP: L'Hospitalet del Llobregat** seleccionamos el SAP, a continuación con **Centre** **Centre: ABS Centre**, podemos elegir el Centre.

Mediante el botón *aceptar*  , confirmamos; y con el botón *cancelar*  , cancelamos y salimos.

## Capítulo 7

# CONCLUSIONES

### 7.1 Conclusiones

En este proyecto se ha analizado la evolución del equipo informático de una organización, evaluado diferentes herramientas de desarrollo e implementado una aplicación Web.

Como consecuencia de la perdida de competencias causadas por reajustes en la estructura y organización de los diferentes departamentos de Sistemas de Información del CatSalut esta aplicación no se va a implantar como solución generada al problema presentado.

A pesar de no implantarse en el SAPLH, se puede decir que los objetivos marcados en su inicio se han logrado:

- Los requerimientos de usabilidad y navegabilidad definidos se han logrado ya que la aplicación es fácil de usar, posee una interfaz amigable y no requiere apenas conocimientos de incidencias de equipos debido a que la aplicación es en su mayoría autocontenido.
- Se había propuesto un sistema multicentro y multiusuario que se adaptara a la estructura organizativa del CatSalut y se ha logrado. A través de este sistema usuarios con diferente rol pueden hacer uso de la aplicación y cambiar de centros, ABS, SAPs y Ámbitos.

- Se ha creado una Web que implementa las funcionalidades requeridas, gestiona las sesiones, controla el acceso de los usuarios y permite la interacción con una base de datos.
- Al ser una aplicación Web no requiere instalación en el cliente basta con crear un acceso directo a la URL de la aplicación Web.
- Finalmente al estar desarrollada en .NET es fácilmente portable a cualquier plataforma.

A modo personal, este PFC ha significado una experiencia muy positiva, ya que he adquirido y consolidado conocimientos cada uno de los roles representados en el transcurso del desarrollo del proyecto, como “*Project manager*” he aprendido el valor de una buena planificación de los recursos disponibles, como analista he aprendido a saber observar e interpretar las necesidades de los usuarios de la aplicación y centrarme en esas necesidades profundizando más adelante en el sistema y por último, como desarrollador he aprendido lo importante que es conocer las herramientas de creación de software para ser más productivo.

## 7.2 Líneas de futuro

Para finalizar, si en algún momento esta aplicación se llegara a implantar dentro del CatSalut, se comentan a continuación una serie de posibles ampliaciones a realizar en la aplicación:

- En primer lugar, dotar a la aplicación de medidas para su funcionamiento en un entorno seguro, a pesar de ser ejecutada en una intranet:
  - Dotar de mecanismos de encriptación de la información ya sea con las ofrecidas por el propio Framework o por terceros.
  - Comunicaciones seguras como Secure Socket Layer (SSL).
- En segundo lugar, incorporar al sistema la tecnología AJAX para mejorar el rendimiento mediante el envío de datos de forma asíncrona y solo de las partes necesarias de los formularios Web de la aplicación.
- Otra posibilidad, haciendo uso de la tecnología wi-fi sería rediseñar los formularios Web para dispositivos móviles como PDAs, teléfonos inteligentes o “smartphones”, TabletsPCs,...

## BIBLIOGRAFÍA

- [Des01] Desarrollador 5 estrellas: Programa de formación de Microsoft .NET Framework ofrecido por Microsoft de forma gratuita.  
<http://www.mslatam.com/latam/msdn/comunidad/dce2005/>
- [Lib06] Jesse Liberty and Dan Hurwitz. Programación con ASP.NET 2.0. Ed. ANAYA Multimedia – O'Reilly, 2006. ISBN: 84-415-2052-6
- [Oni03] Fritz Onion. Essential ASP.NET with Examples in C#. Ed. Pearson Education, Inc. 2003 ISBN: 0-201-76040-1
- [Mat05] Neil Matthew and Richard Stones. Beginning Databases with PostgreSQL: From Novice to Professional, Second Edition. Ed. Apress, 2005. ISBN: 1-59059-478-9
- [Mon01] Mono Hispano, url relacionada con el proyecto Mono con documentación en español. <http://www.monohispano.es/index.php/Portada>
- [Msd01] URL de la biblioteca de Microsoft con toda la documentación relacionada con herramientas de desarrollo, lenguajes,... de Microsoft .NET Framework. <http://msdn2.microsoft.com/es-es/library/default.aspx>
- [Pos01] URL de PostgreSQL con documentación en español e inglés.  
<http://www.postgresql.cl/>
- [Pro01] Proyecto Mono, url oficial del Proyecto Mono con documentación, descargas,... [http://www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page)
- [Rus03] Charlie Russel y otros. Guía completa de Microsoft Windows Server 2003. Running+. Ed. McGraw-Hill / Interamericana de España S.A.U. 2003. ISBN: 84-481-3953-4
- [Sil99] A. Silberschatz y otros. Fundamentos de Bases de Datos. 3<sup>a</sup> Edición. Ed. McGraw-Hill / Interamericana de España S.A.U. 1999. ISBN: 84-481-2021-3
- [Som02] Ian Sommerville. Ingeniería del Software. 6<sup>a</sup> Edición. Ed. Pearson Educación, México. 2002. ISBN: 970-26-0206-8
- [Uni01] Universidad .NET, url con formación relacionada con Microsoft .NET Framework. <http://www.microsoft.com/spanish/msdn/comunidad/uni.net/>

Firmado: Alberto Bermejo Mena

Bellaterra, 1 de junio de 2007

## RESUMEN

Este proyecto se presenta como una solución al problema aparecido por la implantación de la estación clínica de trabajo, e-CAP, en los centros de atención primaria del servicio de atención primaria de la población de l'Hospitalet del Llobregat. La solución desarrollada es una aplicación Web que proporciona la funcionalidad requerida para llevar a cabo el control del equipo informático así como la gestión de las incidencias. Esta aplicación Web está enfocada al personal de atención al usuario de los centros nombrados anteriormente con la supervisión del departamento de Sistemas de Información de cada servicio de atención primaria.

## RESUM

Aquest projecte es presenta com una solució al problema aparegut per la implantació de l'estació clínica de treball, e-CAP, en els centres d'atenció primària del servei d'atenció primària de la població de l'Hospitalet del Llobregat. La solució desenvolupada és una aplicació Web que proporciona la funcionalitat requerida per a dur a terme el control de l'equip informàtic així com la gestió de les incidències. Aquesta aplicació Web està enfocada al personal d'atenció a l'usuari dels centres nomenats anteriorment amb la supervisió del departament de Sistemes d'Informació de cada servei d'atenció primària.

## SUMMARY

This project appears like a solution to the problem appeared by the implantation of the clinical station of work, and-CAP, in the centres of primary attention of the service of primary attention of the population of l' Hospitalet del Llobregat. The developed solution is an application Web that provides the required functionality to carry out the control of the computer science equipment as well as the management of the incidences. This application Web is focused to the personnel of attention to the user of the centres named previously with the supervision of the department of Information systems of each service of primary attention.