



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)

# Desarrollo de Aplicación Catalogo TIC de la Dirección General de Tecnologías de Información de la Generalitat Valenciana

Especialidad: Desarrollo de aplicaciones de software libre

Autor: Antonio Félix Ferrando

Consultor: Gregorio Robles Martínez

Tutor Externo: Carmen Ferrer Ripollés

Fecha: 10/01/2014

Este documento se publica bajo la Licencia *GNU Free Documentation License* (GFDL). Léanse los [términos de uso](#) para más información.

Copyright (C) 2014 Antonio Félix Ferrando

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "Licencia".

## RESUMEN EJECUTIVO

En este proyecto vamos a ver el proceso de transformación de una aplicación que ejercía de pequeño inventario de una organización, hasta convertirse en el componente central de la Base de Datos de Gestión de la Configuración ([CMDB](#)) de activos software de dicha organización.

Para el desarrollo del proyecto, veremos no sólo como se utilizan herramientas del mundo del software libre, sino como se aplica su modelo de desarrollo colaborativo. La herramienta principal sobre la que se desarrolla el proyecto es el *framework* libre de la Generalitat Valenciana gvHIDRA. Veremos una pequeña introducción al mismo y realizaremos un análisis de la forja de desarrollo que actualmente se ofrece a los desarrolladores.

A través de las diferentes fases del ciclo de vida, podremos realizar una recopilación de los trabajos realizados para, finalmente, observar los resultados en las fases finales y evaluar el grado de éxito obtenido.

# ÍNDICE

1.-INTRODUCCIÓN.....	6
1.1.-Preámbulo.....	6
1.2.-Software Libre.....	8
1.3.-Motivación Personal.....	8
1.4.-Estructura de la memoria.....	9
2.-OBJETIVOS.....	11
3.-ENTORNO TECNOLÓGICO.....	12
3.1.-gvHIDRA.....	12
3.1.1.-Un poco de historia.....	12
3.1.2.- Pero, ¿Qué es gvHIDRA? .....	13
3.1.3.-Filosofía.....	14
3.1.4.-Arquitectura.....	15
3.2.-Plataforma de desarrollo.....	16
3.2.1.-Servicios.....	16
3.2.3.-Herramientas.....	17
3.3.-Entorno de desarrollo.....	22
4.-REQUISITOS.....	25
4.1.-Requisitos funcionales.....	25
4.2.-Requisitos no Funcionales.....	27
5.-ANÁLISIS.....	31
5.1.-Estructura funcional de los Activos Software.....	31
5.2.-Gestión de Roles y Permisos.....	34
5.2.1.-Nivel de Activos Software.....	36
5.2.2.-Nivel de Contratos.....	37
6.-DISEÑO.....	39
6.1.-Nuevo esquema relacional.....	39
6.1.1.-Contratos.....	43
6.1.2 Activos Software.....	45

6.2.-Proceso de migración: ETLs.....	48
6.3.-Decisiones de diseño en la interfaz de usuario.....	50
6.4.-Esquema de seguridad.....	52
6.4.1 Análisis de las medidas de seguridad del Framework gvHIDRA.....	52
6.4.2.-Análisis de las medidas de seguridad de la Infraestructura	53
6.4.3.-Diseño de las medidas de seguridad de a nivel de aplicación .....	55
7.-IMPLEMENTACIÓN.....	57
7.1.-Mantenimientos.....	57
7.1.1.-Activos Software.....	57
7.1.2.-Contratos.....	58
7.1.3.-Tablas maestras.....	59
7.2.-Categorización.....	60
7.3.-Listados.....	60
7.4.-Sinergias: JIRA, Portal y Gestión de la Demanda.....	61
7.5.-Refactoring.....	62
8.-CONCLUSIONES Y TRABAJOS FUTUROS.....	65
9.-ANEXOS.....	68
9.1.-Clases de Acceso a JIRA.....	68
10.-GLOSARIO.....	71
11.-BIBLIOGRAFÍA.....	73
12.-LICENCIA.....	74

## 1.-INTRODUCCIÓN

En esta introducción, en primer lugar, vamos a presentar los antecedentes que llevan a la necesidad de iniciar este proyecto. Recogeremos información relativa al funcionamiento de la organización, en este caso la Generalitat Valenciana, que permitirán, desde esa perspectiva, entender la magnitud e importancia del proyecto.

Por otro lado, veremos la estrecha relación de este proyecto con el software libre. Cabe destacar que no sólo se trata de un proyecto que utiliza herramientas procedentes de este mundo; sino que se embebe en los procesos y métodos del mismo durante todo el ciclo de vida.

En un tercer apartado repasaremos las motivaciones personales que me han impulsado a utilizar este proyecto como Proyecto Final de Master.

Finalmente, en el último punto, presentaremos de forma esquemática cual va a ser la estructura de esta memoria.

### **1.1.-Preámbulo**

Tras la última reestructuración del Gobierno Valenciano, el 9 de Septiembre de 2011, a través del decreto [119/2011 del Consell de la Generalitat](#), se creó la Dirección General de Tecnología de Información (en adelante DGTI) para asumir las competencias TIC de la Generalitat Valenciana. Concretamente, y tras algunas revisiones, *“ha asumido las competencias para la administración de la Generalitat Valenciana y sus organismos autónomos en materia de innovación tecnológica, seguridad de los sistemas de información, planificación, coordinación, autorización y control de las tecnologías de la información, las telecomunicaciones, las comunicaciones corporativas, el desarrollo de la sociedad digital y la administración electrónica, con excepción de las que sean ejercidas por la Conselleria de Sanidad en su ámbito competencial. ... Para la consecución de sus objetivos, la DGTI está organizada según el Reglamento Orgánico y Funcional de la Conselleria de Hacienda y Administraciones Públicas ([Decreto 20/2013, de 25 de enero, del Consell](#), publicado en el Diario de la Comunidad Valenciana el 29 de Enero de 2013)”*. (extracto del expediente [DESIG2013](#) CNMY13/DGTEC/7).

Una de las primeras necesidades de la nueva DGTI fue la de inventariar todos los

recursos que tenía que manejar en sus competencias. Por esta razón, se desarrolló la aplicación *CATI* que permitió registrar toda la información necesaria de diferentes conceptos:

- Aplicaciones: aspectos generales de las aplicaciones, tecnología, responsable funcional y técnico, su implicación en LOPD, participantes, sinergias y relación con contratos-asistencias externas.
- Contratos: Contratos vigentes por Conselleria y relación con aplicaciones.
- RRHH o personal: personal vinculado a la labor informática y su experiencia.



Figura 1: captura de pantalla del manual de la aplicación en su primera versión.

En un principio, la aplicación fue dirigida a los responsables de ámbito (en adelante [RA](#)) en las distintas Consellerias, ya que eran los usuarios poseedores de conocimientos suficientes y necesarios para cumplimentar la información. Se abrió un periodo durante el cual, estos perfiles fueron introduciendo y/o completando la información para poder realizar un mapa de aplicaciones, contratos y personal.

Una vez recogida la información, la Dirección General pasó a una fase de estudio en la que, entre otras, se evaluó la mejor manera de poder gestionar el servicio que se debía prestar a los diferentes departamentos de la Generalitat Valenciana. Se decide realizar una transformación cuyo objeto es mejorar la eficiencia y la eficacia de los servicios y dar respuesta a todas aquellas nuevas demandas que se planteen por la organización.

Estos cambios impactan directamente en la aplicación ya que se decide que *CATI* sea

la Base de Datos de Gestión de la Configuración ([CMDB](#)) de activos software de la Dirección General de Tecnologías de la Información. Esto significa, que todos los activos software se gestionarán a través de ella y se convertirá en pieza angular para el resto de componentes de la gestión del servicio.

Esta transformación de la aplicación, con un amplio compendio de requisitos funcionales y no funcionales, son el marco de este Proyecto Final de Master.

## **1.2.-Software Libre**

La relación del proyecto con el software libre viene marcada desde los inicios de la aplicación *CATI*. Esta aplicación se desarrolló con herramientas de software libre como gvHIDRA o Postgresql. En este trabajo presentaremos la más importante en cuanto a presencia en el código, gvHIDRA. Conoceremos este *framework* de desarrollo de aplicaciones de la Generalitat Valenciana a través de un pequeño resumen de sus funcionalidades en el tercer capítulo.

En todo caso, la vinculación de este proyecto con el mundo *open source* va más allá del uso de una tecnología o herramienta. Como veremos en los siguiente puntos, los métodos y procesos del software libre están presentes en todo el ciclo de vida del proyecto. Podremos comprobar como este proceso se pilota a través de una forja de desarrollo que la Dirección General ha proporcionado para sus desarrolladores. Observaremos como se lleva a cabo el proceso de construcción “a modo de bazar” como comentaba Eric S. Raymond en su famoso artículo [“La catedral y el Bazar”](#).

No podemos quitar el foco de uno de los aspectos más importantes que supone este proyecto en relación al software libre. La Dirección General tiene como uno de sus objetivos impulsar el uso de herramientas de fuentes abiertas. La apuesta por el uso de *CATI* como CMDB supone una apuesta clara por las herramientas libres huyendo de postulados oportunistas o coyunturales. Si el proyecto concluye de forma exitosa, esa apuesta supondrá un hito en cuanto a la presencia del software libre en la Generalitat Valenciana.

## **1.3.-Motivación Personal**



La elección de este proyecto no fue casual. Mi labor profesional en los últimos años se ha centrado en el *framework* gvHIDRA y aplicaciones desarrolladas con él. Los cambios en la gestión TIC de la Generalitat, suponen nuevas estrategias, nuevos criterios y nuevos objetivos. Como comentábamos anterior, la apuesta por este proyecto puede suponer un espaldarazo definitivo para la consolidación del uso del software libre en los planes de la Dirección General. Como es evidente, después de 7 años impulsando los proyectos de software libre, no podía renunciar a esta llamada.

Por otro lado, mi participación directa en el desarrollo de la primera versión de *CATI*, me proporcionó un vinculo emocional con el proyecto. Cambiar el destino de la aplicación para que sea una de las piezas angulares de la gestión del servicio es un reto, de por si, apasionante.

En todo esto, no puedo eludir la responsabilidad y el riesgo que supone acometer este proyecto de transformación. Son muchos los condicionantes que hacen difícil su conclusión exitosa. Será complicado satisfacer las necesidades de la Dirección General en cuanto a funcionalidad del proyecto pero más crítico si cabe puede ser la presión del plazo de entrega dada la criticidad del sistema dentro del servicio. Son dos objetivos ambiciosos, pero confío en podremos llevarlos a cabo.

#### **1.4.-Estructura de la memoria**

En los siguientes capítulos vamos a abordar todos los aspectos relativos al desarrollo del proyecto de transformación de la aplicación *CATI*. En el siguiente capítulo, veremos en profundidad los objetivos que nos mueven a realizar el proyecto y que deberemos alcanzar a su finalización.

Para mejorar la comprensión del ámbito de trabajo, en el capítulo tercero nos centraremos en el entorno tecnológico que envuelve a la solución. Presentaremos la herramienta principal que, en este caso, es el *framework* gvHIDRA y explicaremos los componentes que forman la forja de desarrollo.

En el cuarto capítulo recogeremos los requisitos que se tomaron en las entrevistas con los usuarios y que completan las necesidades a cubrir por el proyecto de transformación. En la relación aparecen tanto requisitos funcionales como no

funcionales que se deberán de cumplir en la solución final para poder satisfacer las necesidades de la Dirección General.

Para mostrar los trabajos más importantes realizados en la fase de análisis de la solución, utilizaremos el capítulo quinto. Se revisarán los estudios realizados sobre los requisitos funcionales de mayor repercusión para entender el impacto de los mismos sobre el sistema actual.

En el capítulo sexto abordaremos el diseño. Recogeremos las tareas relativas a reestructuración del sistema de persistencia así como otros trabajos relativos a aspectos tan interesantes como el desarrollo de los procesos de migración o las políticas de seguridad.

No podía faltar en este repaso del ciclo de vida del proyecto la revisión del esfuerzo realizado en la fase de implementación. Esto lo haremos en el capítulo séptimo donde se mostrarán los desarrollos realizados en nuevos mantenimientos, las transformaciones realizadas en las ventanas principales y el proceso de *refactoring* y mejora continua habitual en los ciclos de desarrollo con metodologías ágiles.

Por último, en el capítulo octavo, recapitularemos de forma resumida los trabajos realizados y analizaremos el resultado global del proyecto atendiendo a los objetivos marcados. También, con el fin de plasmar la continuidad del proyecto, recogeremos algunos de los retos que quedarán pendientes para la evolución futura de *CATI*.

## 2.-OBJETIVOS

Como se comentaba en el resumen ejecutivo, este proyecto final de máster tiene como principal objetivo la transformación de una aplicación, *CATI*, en la CMDB de activos software de la Dirección General de Tecnologías de la Información de la Generalitat Valenciana.

Este cambio tiene una serie de condicionantes importantes que deben de ser tomados en consideración para entender la magnitud del cambio solicitado. En primer lugar, uno de los primeros aspectos a tener en cuenta, es la importancia de responder a las necesidades de la Dirección General con calidad y en plazo. La solicitud realizada por la Dirección General se requiere para la elaboración del nuevo modelo de gestión de la demanda. La incorporación de este nuevo sistema de gestión tiene unas fechas ya planificadas. Esto supone que el proyecto debe estar operativo al 100% en una fecha límite ya fijada. Por todo ello, podemos deducir que el objetivo principal es concluir el proyecto cubriendo el 100% de la funcionalidad, con calidad y en plazo.

Otro de los aspectos que no podemos descuidar, son las implicaciones que el nuevo destino de *CATI* conlleva. Ser la CMDB de la organización supone ser una de las piezas que componen el corazón de la gestión del servicio. Esto significa que *CATI* deberá estar preparada para integrarse fácilmente con el resto de piezas que componen el sistema. Es por tanto objetivo de este proyecto preparar la solución para sea realmente un componente más de la gestión del servicio.

Finalmente, como ya hemos nombrado en el apartado anterior, otro de los objetivos del proyecto es la consolidación del uso del software libre en la Generalitat Valenciana. La elección realizada por la Dirección General al lanzar este proyecto adelante, supone una apuesta clara y llena de confianza hacia el software libre. En todo caso, un éxito en la ejecución del proyecto supondría un espaldarazo aun mayor que, de buen seguro, ayudaría a abrirle puertas hacia ámbitos más críticos dentro de la organización.

## 3.-ENTORNO TECNOLÓGICO

En este capítulo, vamos a analizar dos puntos clave a la hora de entender el marco de trabajo proyecto. En primer lugar, presentaremos la herramienta de trabajo principal que se utilizará para el desarrollo de la aplicación: el *framework* gvHIDRA. Esta herramienta de software libre será la base principal del desarrollo y, gracias a sus características, nos permitirá centrar los esfuerzos en las particularidades del negocio de CATI.

En segundo lugar, nos centraremos en la plataforma tecnológica de la que vamos a disponer para desarrollar el proyecto. En este caso presentaremos la plataforma tecnológica que la DGTI ha habilitado para los proyectos de esta naturaleza; con las herramientas y servicios que ofrece. Como veremos, se trata de una forja de desarrollo de proyectos parecida a la de la mayor parte de proyectos de software libre que podemos encontrar hoy en día. Esta forja permitirá realizar un trabajo colaborativo entre todos los miembros del proyecto.

### **3.1.-gvHIDRA**

Es el acrónimo de gvHIDRA: **g**eneralitat **v**alenciana **H**erramienta Integral de Desarrollo **R**ápido de **A**plicaciones.



Dando una pequeña definición técnica, podemos decir que se trata de un entorno de trabajo (*framework*) para el desarrollo de aplicaciones de gestión en entornos web con PHP siguiendo una guía de estilo (una guía para unificar los criterios de aspecto y usabilidad en el proceso de desarrollo de aplicaciones). Si queremos clasificarla, la podemos definir como una herramienta RAD (*Rapid Application Development*) que se adapta a las necesidades particulares de las aplicaciones de las grandes organizaciones.

Siguiendo la definición de *Roger S. Pressman* en su libro Ingeniería del Software podemos decir que gvHIDRA “no es un patrón arquitectónico, sino un esqueleto con una colección de puntos de conexión que le permiten adaptarse a un dominio de un problema específico. Estos puntos permiten integrar clases o funcionalidades específicas del problema. En conclusión, es un conjunto de clases que cooperan”.

En los siguientes puntos iremos desgranando la historia y las características del proyecto para facilitar la comprensión y la repercusión del proyecto en su ámbito de actuación.

#### 3.1.1.-Un poco de historia

El Servicio de Organización e Informática (SOI) de la Conselleria de Infraestructuras y

Transporte (en adelante CIT) emprendió en 2007 el proyecto gvPONTIS cuyo objetivo principal era migrar todos los sistemas a sistemas de código abierto. Entre otras facetas afectadas se encontraban los lenguajes de programación (se seleccionaron PHP y Java), los SGBD (se decidió fomentar el uso de Postgresql), la arquitectura (pasar al desarrollo web/tres capas), etc.

Con todo ello, se decidió crear un proyecto en PHP que, basándose en la guía de estilo de las aplicaciones de la CIT, aportara valor a los desarrollos aumentando la productividad de los mismos.

Para ello, se debían contemplar dificultades como el cambio al paradigma N-capas, peculiaridades del entorno web (HTML, Javascript, etc), el enfoque OpenSource, etc. Por tanto se decidió incorporar como requerimiento la simplificación del entorno de trabajo para un desarrollador.

Con todo ello, se creó un proyecto (igep: Implementación de la Guía de Estilo en PHP) cuya primera versión estable se etiquetó el 16-11-2004. Este proyecto, al ser liberado con licencia *GNU GPL* versión 2 tomó la denominación de gvHIDRA.

### 3.1.2.- Pero, ¿Qué es gvHIDRA?

En este apartado, vamos a recoger algunas de las características que hacen al *framework* una herramienta eficiente:

- Patrones: para facilitar el desarrollo se han definido una serie de patrones de diseño. Estos patrones definen la forma de representación de la información (formato tabular, registro, etc) y la forma con la que interacciona el usuario con dicha información. Esto nos lleva a que con la selección de un patrón estamos marcando el diseño global de la ventana.
- Componentes complejos: la experiencia acumulada nos dice que todas las aplicaciones necesitan de una serie de componentes “complejos”. Ventanas de selección, listas enlazadas, acciones de interfaz (en web tecnología AJAX), mensajes de información, etc. El *framework* genera estos componentes simplificando su utilización en las aplicaciones.
- Operaciones preprogramadas y parametrizables: al igual que con los componentes, cierta problemática se repite en todas las aplicaciones. Por esa razón, se ha generalizado y resuelto en el *framework*; siendo incorporada a las aplicaciones de forma transparente. Algunos de los casos más típicos son: control de concurrencia, búsquedas inteligentes, CRUD, validaciones, transacciones, etc.
- Soporte a diferentes SGBD: a través del proyecto PEAR::MDB2, el *framework* permite trabajar con diversos SGBD. Concretamente se han probado Oracle, Postgresql, SQL Server y Mysql.

- Listados: uno de los mayores retos en el ámbito del proyecto fue generar informes de forma tan versátil como los *datawindows* de *PowerBuilder*. Gracias al proyecto *jasperreports* se ha conseguido.
- Arquitectura MVC: el *framework* garantiza la arquitectura *MVC* forzando la separación de la lógica de negocio de la presentación mediante una distribución física de los ficheros fija.
- Control de la vista: el *framework* intenta simplificar la labor del programador, por ello no se es necesario introducir ninguna línea de código *HTML* o *Javascript* en la aplicación. Simplemente tiene que centrar sus esfuerzos en la parte de negocio (PHP). Para la parte de presentación tiene una serie de plantillas (una por patrón) en las que únicamente tiene que introducir los componentes a utilizar (*plugins*).
- Customs o temas: para poder definir las particularidades propias de la organización, el *framework* se distribuye en una arquitectura *App/Custom/core* que permite modificar tanto el aspecto (CSS, imágenes, etc) como definir comportamiento propio para la organización.
- Testing: el *framework* incorpora la herramienta *PHPUnit* para que se puedan realizar testeos sobre el código generado.
- Autenticación: Para poder incorporarse en diferentes organismos, el *framework* incorpora un mecanismo de validación extensible siendo capaz de acoplarse a cualquier sistema de validación a través de PHP.
- Depuración y Auditoría: Tanto para facilitar la depuración durante el desarrollo como para realizar tareas de auditoría en explotación, el *framework* incorpora un mecanismo sencillo mediante registro en BD.

### 3.1.3.-Filosofía

Tras haber visto que aporta gvHIDRA, vamos a destacar un aspecto de diseño que nos revela la filosofía del proyecto. Evidentemente, se trata de un proyecto *Open Source* y que, aunque ha sido concebido en la Generalitat Valenciana, ha sido diseñado para poder ser utilizado en todo tipo de organizaciones.

Esto lo podemos comprobar simplemente al observar la arquitectura de configuraciones y la carga de las mismas. Para permitir la reutilización, las características de configuración se cargan en forma de herencia en el siguiente orden:

1. *FW*: carga las características de configuración por defecto del *framework*.
2. *Custom*: permite modificar las características de configuración para toda la organización. De este modo, no será necesario añadir dicha modificación en todas las aplicaciones.
3. *App*: permite modificar la configuración a nivel de la aplicación actual.

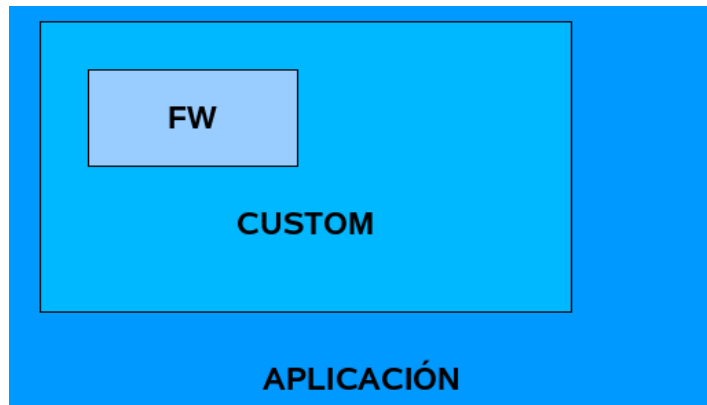


Figura 2: Estructura en capas funcionales de gvHIDRA.

A esta arquitectura de carga, tenemos que añadir que gvHIDRA utiliza carga a dos niveles (estática y dinámica), de forma que permite adaptarse a requisitos evaluables en tiempo de ejecución.

### 3.1.4.-Arquitectura

Para los conocedores de la arquitectura web, en este apartado vamos a adentrarnos, de forma concisa, en la arquitectura interna de gvHidra. Básicamente con esto pretendemos que el lector pueda entender la arquitectura del *framework* a través de un ejemplo sencillo. Mediante un gráfico, veremos como un estímulo de pantalla atraviesa toda la arquitectura del *framework* y como este responde.

El siguiente diagrama de secuencia presenta los diferentes actores (clases) que intervienen en la ejecución de una acción. Este diagrama no pretende ser una referencia precisa de todos los procesos internos de gvHidra, pero será muy útil a efectos aclaratorios.

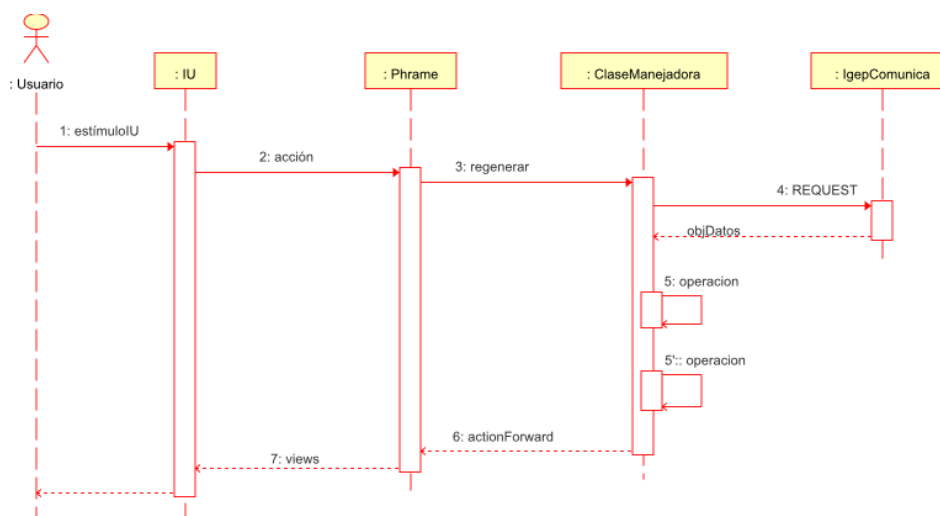


Figura 3: Diagrama de Secuencia de esquema E/S de una petición en gvHIDRA

### 3.2.-Plataforma de desarrollo

Una vez presentada la herramienta, vamos a conocer la plataforma de desarrollo que vamos a utilizar. Para el desarrollo de la aplicación CATI, se va a contar con los servicios de la forja de gvHIDRA. En primer lugar, aclararemos que es la forja, los proyectos que alberga y la accesibilidad de los mismos.

La forja es un conjunto de servicios que se ofrecen para facilitar el desarrollo colaborativo de los proyectos gvHIDRA en el ámbito de la Dirección General de Tecnologías de la Información. Con estos servicios, se pretende ayudar a los diferentes equipos de desarrollo y fomentar la colaboración entre los mismos. La forja está destinada exclusivamente para albergar proyectos desarrollados con gvHIDRA en el ámbito de la DGTI, el propio *framework* y sus proyectos auxiliares.



Figura 4: captura de pantalla de la página de entrada a la forja de gvHIDRA.

Es importante destacar que todos los proyectos son, por definición, de acceso privado; excepto el *framework* (proyecto gvHIDRA), el generador (proyecto Genaro) y la aplicación de prueba (proyecto Demo). Esto significa que sólo los usuarios autorizados y con permisos en el proyecto podrán acceder a la información, código, ficheros, etc.

#### 3.2.1.-Servicios

En relación a los servicios, la forja está ubicada en la máquina <http://gvhidra.gva.es>, está abierta a internet y ofrece los siguientes:



1. Gestión de identidad centralizado
2. Gestor de tickets
3. Control de versiones
4. Foro/Wiki
5. Integración continua
6. Monitorización
7. Control de estadísticas
8. Gestor de documentos
9. Gestor de Archivos y/o Documentos
10. Gestor de Noticias
11. Calendario
12. Gantt
13. Planificación de versión
14. Lista de correo de gvHIDRA
15. Formación: Cursos, talleres, etc.

### 3.2.3.-Herramientas

A continuación realizaremos una recopilación de las herramientas más destacadas de la forja. Veremos una breve descripción de cada una de ellas.

#### Subversion

Subversion es un sistema de control de versiones diseñado específicamente para reemplazar al popular CVS. Es software libre bajo una licencia de tipo Apache/BSD y se conoce popularmente como SVN por ser el nombre de la herramienta utilizada en la línea de comandos.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos equipos. En la siguiente figura podemos ver la estructura del trunk del proyecto *CATI*.



Figura 5: captura de pantalla de la visualización HTTP del control de versiones.

## Redmine

Redmine es una herramienta para la gestión de proyectos y el seguimiento de errores escrita usando el *framework Ruby on Rails*. Incluye un calendario y unos diagramas de Gantt para la representación visual de la línea del tiempo de los proyectos. El diseño está significativamente influenciado por Trac, otra herramienta con características similares.

Es software libre y de código abierto, disponible bajo la Licencia Pública General de GNU v2. En la figura 6 podemos ver la página de inicio del proyecto CATI en el redmine de la forja.

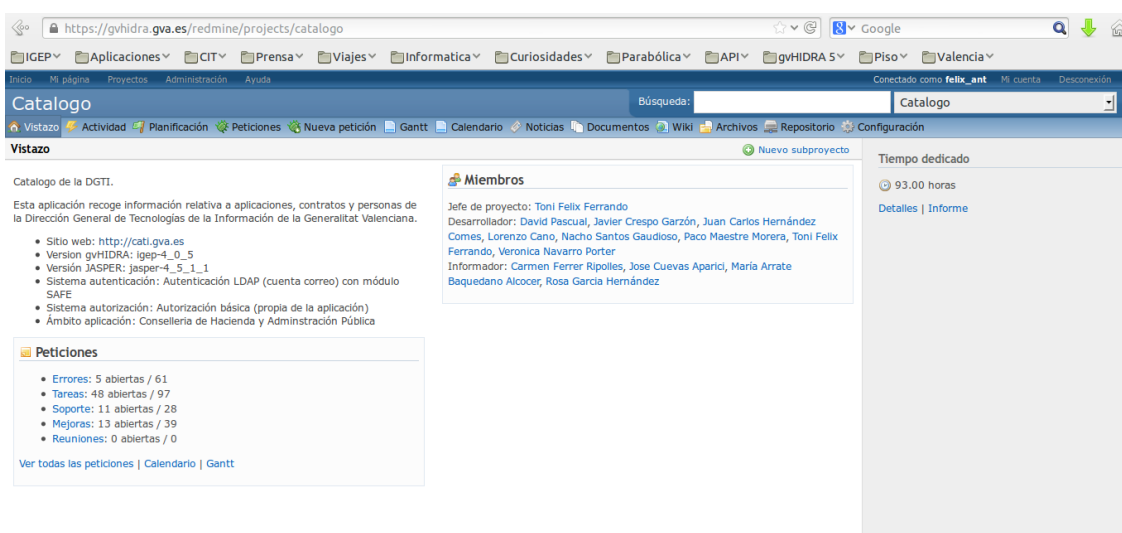


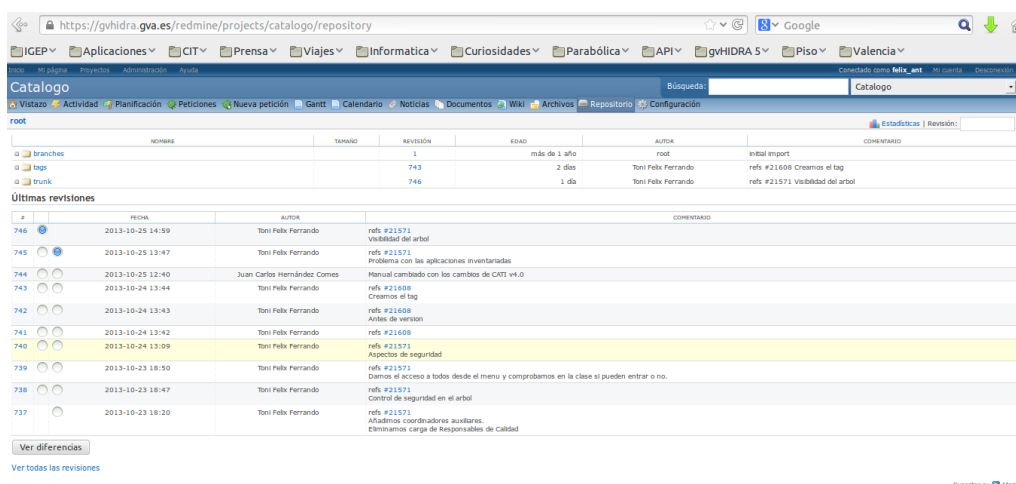
Figura 6: captura de pantalla de la página principal de la aplicación CATI en la forja gvHIDRA

### Características:

- Soporta múltiples proyectos.
- Roles flexibles basados en control de acceso.
- Sistema de seguimiento de errores flexible.
- Diagramas de Gantt y calendario.
- Administración de noticias, documentos y archivos.
- Fuentes web y notificaciones por correo electrónico.
- Integración SCM (Subversion, CVS, Git, Mercurial, Bazaar y Darcs).
- Soporta diferentes bases de datos (MySQL, PostgreSQL y SQLite).

Una de las vistas más utilizadas es la integración SCM para realizar tareas de

supervisión. En la figura 7 podemos ver la estructura del SVN del proyecto, los últimos cambios del repositorio (*commits*) y su vinculación con las tareas de desarrollo.



REVISIÓN	FECHA	AUTOR	COMENTARIO
746	2013-10-25 14:59	Toni Felx Ferrando	refs: #21571 Visibilidad del arbol
745	2013-10-25 13:47	Toni Felx Ferrando	refs: #21571 Problemas con las aplicaciones inventariadas
744	2013-10-25 12:40	Juan Carlos Hernández Comes	Manual cambiado con los cambios de CATI v1.0
743	2013-10-24 13:44	Toni Felx Ferrando	refs: #21608 Creamos el tag
742	2013-10-24 13:43	Toni Felx Ferrando	refs: #21608 Antes de version
741	2013-10-24 13:42	Toni Felx Ferrando	refs: #21608
740	2013-10-24 13:09	Toni Felx Ferrando	refs: #21571 Aspectos de seguridad
739	2013-10-23 18:50	Toni Felx Ferrando	refs: #21571 Damos el acceso a todos desde el menu y comprobamos en la clase si pueden entrar o no.
738	2013-10-23 18:47	Toni Felx Ferrando	refs: #21571 Control de seguridad en el arbol
737	2013-10-23 18:20	Toni Felx Ferrando	refs: #21571 Añadimos coordinadores auxiliares. Eliminamos carga de Responsables de Calidad

Figura 7: captura de la vista SCM del repositorio de CATI

## OpenLDAP

OpenLDAP es una implementación libre y de código abierto del protocolo *Lightweight Directory Access Protocol* (LDAP) desarrollada por el proyecto OpenLDAP. LDAP es un protocolo de tipo cliente-servidor para acceder a un servicio de directorio independiente de la plataforma. Entre otras cosas, nos permite contener los datos (logins, claves) de una serie de usuarios y realizar la autenticación en máquinas clientes a través de un único servidor OpenLDAP.

Está liberado bajo su propia licencia *OpenLDAP Public License*. Un directorio LDAP destaca sobre los demás tipos de bases de datos por las siguientes características:

- es muy rápido en la lectura de registros.
- permite replicar el servidor de forma muy sencilla y económica.
- muchas aplicaciones de todo tipo tienen interfaces de conexión a LDAP y se pueden integrar fácilmente.
- Dispone de un modelo de nombres globales que asegura que todas las entradas son únicas.
- Permite múltiples directorios independientes.
- Funciona sobre TCP/IP y SSL.
- La mayoría de aplicaciones disponen de soporte para LDAP.
- La mayoría de servidores LDAP son fáciles de instalar, mantener y optimizar.

Para su gestión, la forja utiliza la herramienta [phpldapadmin](http://phpldapadmin.org/). Esta herramienta web

permite realizar una gestión sencilla del servidor LDAP. A continuación, en la figura 8, podemos apreciar los diferentes grupos de trabajo (uno por proyecto) que actualmente existen en la forja:

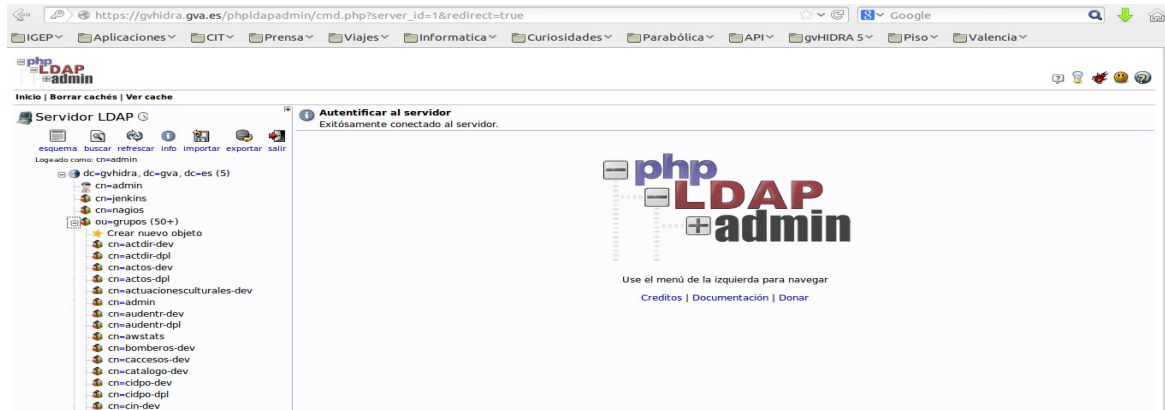


Figura 8: captura de pantalla de la instalación phpldapadmin de la forja.

## Jenkins

Jenkins es un software de Integración continua open source escrito en Java. Esta basado en el proyecto Hudson.

Proporciona integración continua para el desarrollo de software. Es un sistema corriendo en un servidor que es un contenedor de servlets, como Apache Tomcat. Soporta herramientas de control de versiones como CVS, Subversion, Git, Mercurial y Clearcase y puede ejecutar proyectos basados en Apache Ant y Apache Maven, así como scripts de shell y programas batch de Windows.

El desarrollador principal es Kohsuke Kawaguchi. Liberado bajo licencia MIT, es software libre.

## Awstats

Esta aplicación permite analizar los accesos realizados al servidor web a efectos estadísticos. Con ella podremos ver el tráfico de red, los recursos más descargados, número de visitas, etc.

Por ejemplo, en la siguiente figura podemos apreciar el tráfico de la forja en relación a visitas diferentes en los últimos meses.

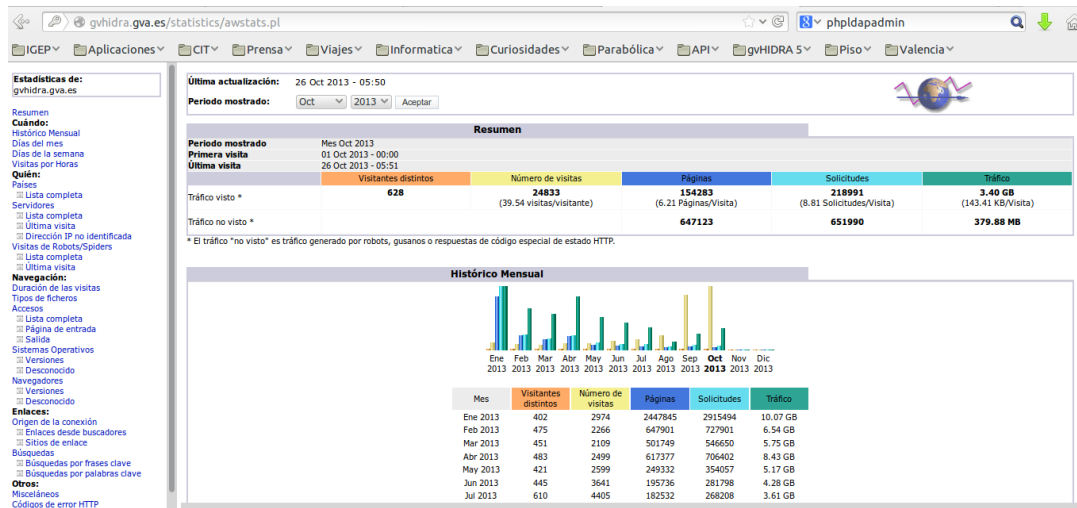


Figura 9: captura de pantalla del awstats mostrando del gráfico del histórico mensual de visitas a la forja.

## Nagios

Esta aplicación nos permitirá monitorizar el rendimiento de los servidores y nos notificará mediante correo electrónico cuando se produzcan alertas o fallos en alguno de los servicios.

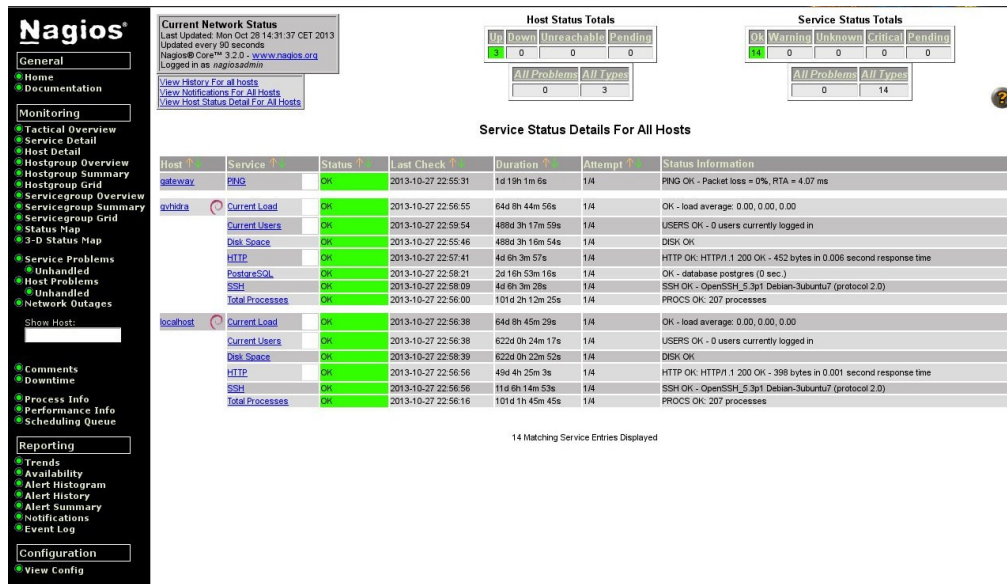


Figura 10: captura del panel de control de Nagios.

## Mailman

Para gestionar la lista de soporte de gvhidra y los e-newsletters se va a utilizar este software de la FSF. Se basa en una interfaz web que hace más fácil la gestión de los

usuarios y las listas.

Se distribuye bajo la licencia GNU GPL v3. Se ha escrito en Python en su mayor parte con alguna pequeña parte de código en C (aspectos de seguridad).

Aunque la lista no se encuentra físicamente en la máquina [gvhidra.gva.es](http://gvhidra.gva.es), forma parte de los servicios que ofrece la Dirección General a los programadores del *framework*.

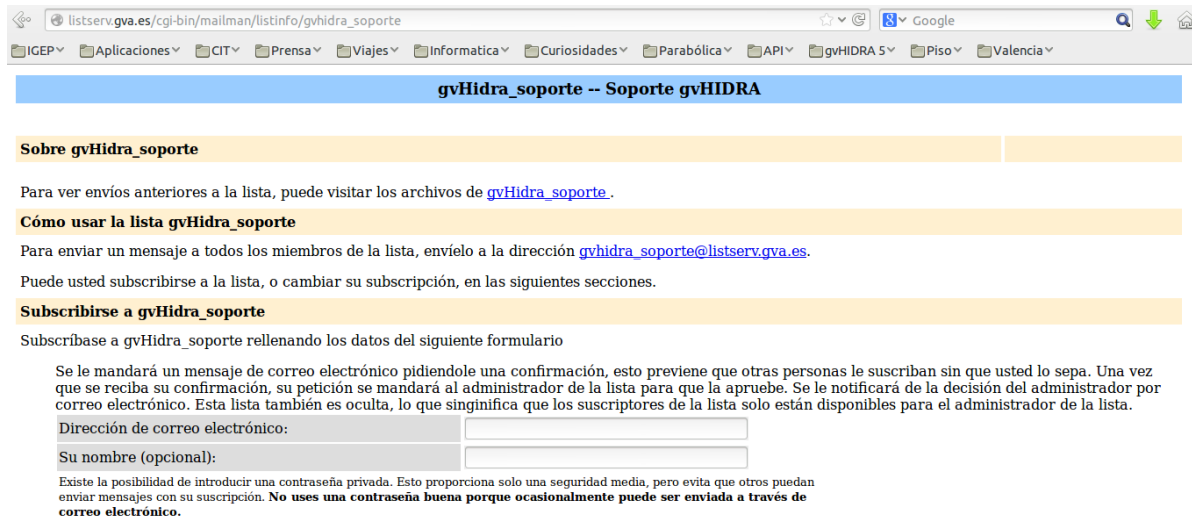


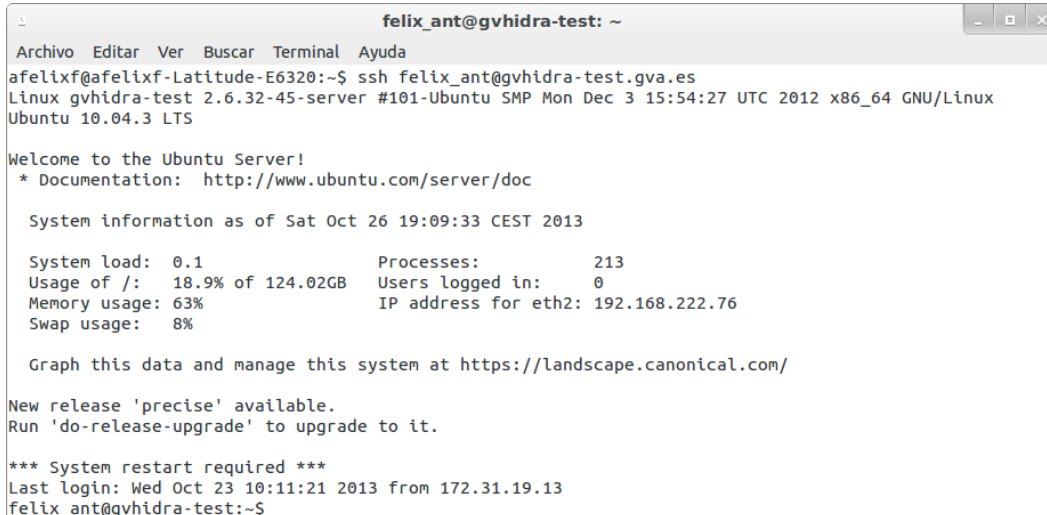
Figura 11: captura de pantalla de página de inscripción a la lista gvHIDRA.

### 3.3.-Entorno de desarrollo

De forma opcional, la DGTI ofrece un entorno de desarrollo de aplicaciones gvHIDRA que puede ser utilizado para desarrollar aplicaciones dentro de dicho ámbito. Este entorno se encuentra ubicado en la máquina <http://gvhidra-test.gva.es> y sólo está disponible en el anillo de la Generalitat Valenciana y a aquellas IPs que lo soliciten. Ofrece los siguientes servicios:

1. Servidor Web: Apache
2. PHP
  - Puerto 80 PHP 5.2.17
  - Puerto 83 PHP 5.3.17
  - Puerto 84 PHP 5.4.7
  - *Próximamente* Puerto 85 PHP 5.5
- SGBD
  - Puerto 5432 Postgres
  - Puerto 1521 Oracle
  - Puerto 3306 MySQL
3. SSH

Para conectarse al entorno de desarrollo, los programadores deben utilizar SSH y ubicar el código en su directorio \$HOME/svnwrk. La configuración de Apache permite que cada usuario puede entrar en su espacio de trabajo a través de la ruta <http://gvhidra-test.gva.es/~nomusuario>. A continuación mostramos una captura con una conexión a la máquina donde destaca el uso de la conexión sin password.



```

felix_ant@gvhidra-test: ~
Archivo Editar Ver Buscar Terminal Ayuda
afelixf@afelixf-Latitude-E6320:~$ ssh felix_ant@gvhidra-test.gva.es
Linux gvhidra-test 2.6.32-45-server #101-Ubuntu SMP Mon Dec 3 15:54:27 UTC 2012 x86_64 GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to the Ubuntu Server!
 * Documentation: http://www.ubuntu.com/server/doc

System information as of Sat Oct 26 19:09:33 CEST 2013

System load:  0.1          Processes:      213
Usage of /:   18.9% of 124.02GB  Users logged in:  0
Memory usage: 63%          IP address for eth2: 192.168.222.76
Swap usage:   8%

Graph this data and manage this system at https://landscape.canonical.com/

New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Wed Oct 23 10:11:21 2013 from 172.31.19.13
felix_ant@gvhidra-test:~$
  
```

Figura 12: captura de consola con conexión a la máquina de desarrollo.

Para hacer un uso más sencillo del entorno, se recomienda a los programadores utilizar la herramienta *sshfs* que permite montar una unidad remota por un tunel *ssh*. A continuación mostramos una captura con un *script* de ejemplo de montaje mediante dicha herramienta y una conexión.



```

afelixf@afelixf-Latitude-E6320: ~
Archivo Editar Ver Buscar Terminal Ayuda
afelixf@afelixf-Latitude-E6320:~$ cat bin/montarHIDRA.sh
#Montaje de unidades de RED
DESADIR=$HOME/felix_anthIDRA
echo "Inicio de sesión -> Montaje de unidades de red"
#Comprobamos si YA esta montado el firectorio
if [ -f $DESADIR/udMontada ]
then
    echo "Directorio public_html de ya montado "
else
    echo "Montando directorio desarrollo..."
    if sshfs felix_ant@gvhidra-test.gva.es:/home/felix_ant/public_html $DESADIR -oreaddir_ino,work
around=rename,direct_io,uid=$UID,gid=$GROUPS,ServerAliveInterval=15
    then
        echo "Montaje de entorno de desarrollo de gvHIDRA"
    else
        echo "Problemas al montar el directorio!!"
    fi
fi
afelixf@afelixf-Latitude-E6320:~$ bin/montarHIDRA.sh
Inicio de sesión -> Montaje de unidades de red
Montando directorio desarrollo...
Montaje de entorno de desarrollo de gvHIDRA
afelixf@afelixf-Latitude-E6320:~$ █
  
```

Figura 13: captura de consola con script de montaje de la unidad por *sshfs*.

Tal y como se ha comentado, a través del navegador, el programador puede acceder a su *home* para ver la evolución de su trabajo. A continuación mostramos una captura

del contenido de una carpeta de desarrollo de un programador.

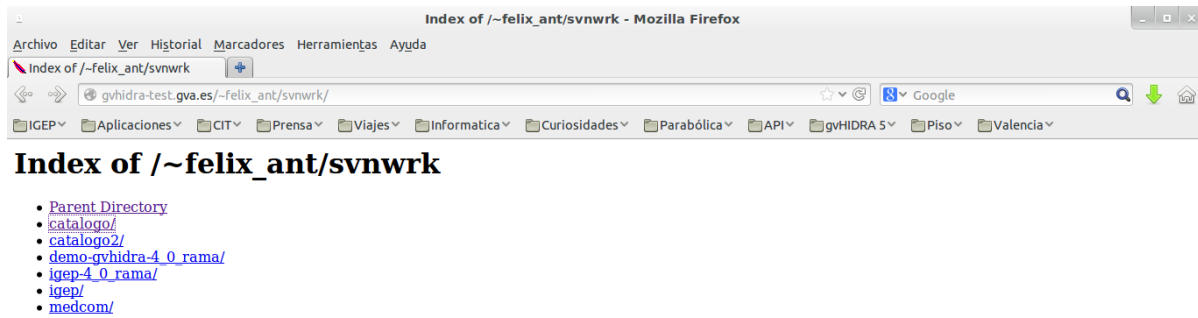


Figura 14: captura de navegador con el contenido de la unidad montada en el espacio de usuario.

Con todo esto, hemos visto el entorno tecnológico que utilizaremos durante la realización de este proyecto de transformación de la aplicación *CATI*. En los siguientes apartados, veremos como tanto la herramienta principal como la forja de desarrollo son dos recursos presentes en todo el ciclo de vida.



## 4.-REQUISITOS

Una vez ubicados, en cuanto a lo referente al entorno tecnológico, pasamos a realizar un resumen de los requisitos más importantes para la realización de este Proyecto Final de Master.

Las sucesivas reuniones con los usuarios funcionales han concluido con la definición de una serie de requisitos funcionales y no funcionales que a continuación desarrollaremos en detalle.

### **4.1.-Requisitos funcionales**

Identificador	RF-01
Nombre	Catálogo de Activos Software
Descripción	<p>En la versión anterior se desarrolló el concepto de Aplicaciones, pero ahora se debe completar para convertirlo a <a href="#">Elemento de Configuración (CI)</a> de Activo Software. Se recogerá más información para completar el catálogo de activos y la gestión de los mismos. Se deberá adaptar la información existente para que se exprese según este nuevo concepto.</p> <p>Es importante tener en cuenta que no debe perderse la información actual. Esta información se obtuvo gracias al esfuerzo compartido por todos los RA y los Jefes de Servicio en la primera fase. La nueva información deberá completar a la ya existente en la medida de lo posible.</p>
Prioridad	Alta
Aceptado	Aceptado

Identificador	RF-02
Nombre	Integración con la Gestión de la Demanda

Descripción	<p>Se va a desarrollar un sistema integral de Gestión de la Demanda para la Dirección General. Debe realizarse el proceso de integración con la <i>CMDB</i> de Activos Software para poder realizar el trabajo de forma adecuada. Dentro de este apartado tenemos la Gestión de Incidencias que se realizará con <a href="#">JIRA</a> o la Gestión de los grupos de escalado (contratos externos, empresas adjudicatarias, responsables, etc).</p> <p>Este requisito requerirá un esfuerzo importante para gestionar las sinergias entre los diferentes componentes que participarán en dicha gestión.</p>
Prioridad	Alta
Aceptado	Aceptado

Identificador	RF-03
Nombre	Nueva estructura orgánica de los A.S. de la DGTI
Descripción	<p>Se requiere plasmar la nueva estructura orgánica de los Activos Software de la Dirección General en la aplicación <i>CATI</i>.</p> <p>La interfaz debe ser simple y navegable (p.e. estructura arborea). Se deberán poder ver/modificar los responsables de cada uno de los niveles de la estructura orgánica de los Activos Software.</p>
Prioridad	Alta
Aceptado	Aceptado

Identificador	RF-04
Nombre	Cambio de roles y permisos.
Descripción	<p>Con el nuevo concepto de Activos Software, todo el personal de la Dirección General podrá acceder en modo lectura a la información. Los permisos de modificación se asignarán según la estructura de la DGTI y/o la responsabilidad técnica.</p>

	Se deberán eliminar algunos roles actualmente activos como el correspondiente a la figura de Responsable de Ámbito y crear algunos nuevos como el de Coordinador.
Prioridad	Alta
Aceptado	Aceptado

Identificador	RF-05
Nombre	Nuevos listados
Descripción	<p>Se requerirá la creación de nuevos listados de Activos Software, de Gestión de la Demanda y/o de los nuevos Expedientes.</p> <p>De inicio, la nueva estructura de la información requerirá modificar los listados existente; pero también será necesario explotar la información con otros filtros y salidas.</p>
Prioridad	Media
Aceptado	Aceptado

#### 4.2.-Requisitos no Funcionales

Identificador	RNF-01
Nombre	Migrar al SGBD Oracle
Descripción	<p>La aplicación actual trabaja con PostgreSQL, pero por necesidades de integración con otros sistemas de la DGTI, se requiere que se migre a Oracle. Esta migración facilitará la integración con otros proyectos como AGRUPA o gvCESTA.</p> <p>Para realizar la migración desde el esquema actual de PostgreSQL se podrá hacer uso de herramientas como el <a href="#">Pentaho Data Integration (Keettle)</a> que facilitan la creación de <a href="#">ETLs</a> de migración/transformación.</p>

Prioridad	Alta
Aceptado	Aceptado

Identificador	RNF-02
Nombre	Uso de información centralizada
Descripción	<p>La aplicación actual no tenía acceso a los datos comunes de la Generalitat. La Dirección General va a centralizar todos estos esquemas comunes (como la propia <i>CATI</i>) en una base de Datos Oracle.</p> <p>Al desplegarse la aplicación en los nuevos entornos tendrá acceso a esta información por lo que se debe revisar la integración con la guía de personas y con los órganos comunes.</p>
Prioridad	Alta
Aceptado	Aceptado

Identificador	RNF-03
Nombre	Seguridad
Descripción	<p>Al aumentar la relevancia de la aplicación se deben realizar mayores controles de seguridad.</p> <p>Se debe revisar la política de seguridad tanto a nivel de configuración de servidor como a nivel de código. Se propondrá realizar una auditoría de seguridad en la que se realicen algunos test de intrusión.</p>
Prioridad	Alta
Aceptado	Pendiente

Identificador	RNF-04
Nombre	Nuevos entornos
Descripción	La DGTI está montando nuevos entornos de desarrollo,

	preproducción y producción para todas las aplicaciones. Está pendiente tener disponible dichos entornos en breve y se deberá de migrar la aplicación a ellos.
Prioridad	Alta
Aceptado	Aceptado

Identificador	RNF-05
Nombre	Nivel de servicio y rendimiento
Descripción	<p>La relevancia que ha asumido la aplicación hacen que sea un requisito garantizar unos niveles mínimos de servicio y de rendimiento.</p> <p>En primer lugar, se debe tener en consideración que no puede cortarse el servicio de la aplicación actual ya que es fuente única de información. Se planificará el proceso de migración y la accesibilidad a la información pudiendo tener habilitado el acceso a la versión anterior de <i>CATI</i> en modo lectura hasta tener la nueva versión accesible.</p> <p>Por otro lado, es importante tener en cuenta que el nuevo destino de la aplicación conlleva un mayor número de peticiones. Por ello, se deben hacer pruebas de carga (estrés) sobre el aplicativo para validar su funcionamiento con un mayor número de usuarios concurrentes.</p> <p>Por último, es reseñable que el <i>time to market</i> es un factor clave del proyecto. Evidentemente no se trata de una ventaja competitiva, pero sí de una presión interna por las sinergias con otros componentes y por la coyuntura en constante evolución/cambio de la DGTI. Recordemos que los pliegos sobre los que se apoyará la Dirección para transformar el servicio se publicarán con la información de la aplicación <i>CATI</i>.</p>
Prioridad	Alta

Aceptado

Aceptado

## **5.-ANÁLISIS**

En este apartado vamos a estudiar los aspectos más importantes que se han revisado para analizar el impacto de los requerimientos anteriormente indicados en el proyecto de transformación.

No pretendemos profundizar en la fase de análisis ni realizar un compendio de documentos técnicos. El objetivo en este apartado es dar un pequeño vistazo al trabajo realizado para identificar el grado de afección de los requerimientos en el sistema actual y las decisiones tomadas más importantes para resolverlos.

En el primer apartado, vamos a desgranar la nueva estructura funcional que se requiere para los activos software. Veremos como se ha organizado la DGTI internamente mediante una estructura a tres niveles y cuales son las características más importantes que afectarán a los activos software.

En el segundo apartado estudiaremos el impacto que supone el cambio de destino de la aplicación en cuanto a la gestión de roles y permisos. Veremos la eliminación de algunos roles y la aparición de nuevos actores. Como era previsible, la gestión de permisos se verá afectada casi en su totalidad.

### ***5.1.-Estructura funcional de los Activos Software***

La prestación de servicios TIC a la Generalitat Valenciana precisa de una transformación cuyo objeto sea mejorar la eficiencia y la eficacia de los servicios que actualmente reciben los departamentos y, al mismo tiempo, dé respuesta a todas aquellas nuevas demandas de servicio que se planteen por la organización.

En este marco, la DGTI ha iniciado un proceso por implantar un nuevo modelo de gestión de las tecnologías de la información y las comunicaciones de la Generalitat Valenciana cuyo objetivo principal es proporcionar soluciones que mejoren la eficiencia técnica y económica de los servicios TIC.

Para la consecución de sus objetivos, la DGTI está organizada según el Reglamento Orgánico y Funcional de la Conselleria de Hacienda y Administraciones Públicas

(Decreto 20/2013, de 25 de enero, del Consell, publicado en el Diario de la Comunidad Valenciana el 29 de Enero de 2013).

Como se recoge en la toma de requisitos (RF-03), uno de los cambios que se requiere para adaptar la aplicación *CATI* es plasmar dicha nueva estructura funcional a partir de la cual se van a gestionar los activos software en la Dirección General. Esta estructura se compone de 3 niveles (categorías, áreas y marcos) que permitirán agrupar los diferentes activos por su valor funcional. Esta estructura de niveles también permitirá realizar una mejor distribución de roles y dependencias.

Los activos software por tanto se asociarán a una categoría, área y marco según su destino funcional de forma que se podrá distribuir con mayor eficacia su gestión en el seno de la Dirección General. A continuación mostramos la estructura funcional que se ha hecho pública para el expediente de contratación CNMY13/DGTEC/7 DESIG-2013.



CATEGORÍA	ÁREA	MARCO
1. ADMINISTRACIÓN ELECTRÓNICA, GESTIÓN DE PORTALES Y PLATAFORMAS TECNOLÓGICAS	1.1 PORTALES	1.1.1 PORTALES
	1.2 PLATAFORMAS I	1.2.1 GESTIÓN DE IDENTIDADES Y USUARIOS
		1.2.2 G. DE LA DEMANDA, G. DE PROYECTOS, INFORMÁTICA Y OTRAS APLICACIONES DE SOPORTE
	1.3 PLATAFORMAS II	1.3.1 BI Y CUADROS DE MANDO
		1.3.2 GIS, FRAMEWORKS Y LLIUREX
	1.4 ADM. ELECTRÓNICA	1.4.1 ADMINISTRACIÓN ELECTRÓNICA
		1.4.2 FIRMA Y PORTAFIRMAS, G. EXPEDIENTES ELECTRÓNICOS Y GESTIÓN DOCUMENTAL
1.4.3 PLATAFORMA DE INTEROPERABILIDAD		
2 GESTIÓN DE SUBVENCIONES, GESTIÓN ADMINISTRATIVA COMÚN, DE RECURSOS HUMANOS ECONOMÍA Y	2.1 GESTIÓN ADMINISTRACIÓN COMÚN	2.1.1 RECURSOS HUMANOS
		2.1.2 MASTIN CORPORATIVO, REGISTRO E/S, GESTIÓN ECONÓMICA Y CONTRATACIÓN
		2.1.3 SANCIONES, INSPECCIONES, ARCHIVO, QUEJAS, CONVENIOS, RECURSOS Y DATOS COMUNES
		2.1.4 PRESIDENCIA, GABINETES Y ALTA DIRECCIÓN
	2.2 SUBENCIONES I	2.2.1 SERVEF
		2.2.2 ECONOMÍA, INDUSTRIA, COMERCIO Y CONSUMO, DG. DE TRABAJO Y DGTI (SIDE)
2.3 SUBENCIONES II	2.3.1 AGRICULTURA	
3 SISTEMAS DE ORDENACIÓN DE ACTIVIDADES, SISTEMAS DEL ÁMBITO SOCIAL, SEGURIDAD E	3.1 ECONOMIA Y EMPLEO	3.1.1 SERVEF
		3.1.2 ECONOMÍA, INDUSTRIA, COMERCIO Y SECTOR PÚBLICO EMPRESARIAL
	3.2 INFRAESTRUCTURAS Y MEDIO AMBIENTE	3.2.1 MEDIO AMBIENTE
		3.2.2 INFRAESTRUCTURAS
	3.3 DEPENDENCIA	3.3.1 DEPENDENCIA I
		3.3.2 DEPENDENCIA II
	3.4 AGRICULTURA Y BIENESTAR	3.4.1 BIENESTAR SOCIAL E IVAJ
		3.4.2 AGRICULTURA
3.5 TRABAJO Y ESTADÍSTICA	3.5.1 TRABAJO Y ESTADÍSTICA	
3.6 JUSTICIA Y GOBERNACIÓN	3.6.1 JUSTICIA Y GOBERNACIÓN	
4 SISTEMAS DE GESTIÓN DE EDUCACIÓN, CULTURA Y DEPORTES	4.1 GESTIÓN ACADÉMICA Y ADMINISTRATIVA DE LOS CENTROS EDUCATIVOS	4.1.1 MÓDULOS PRINCIPALES
		4.1.2 MÓDULOS ADMINISTRATIVOS
		4.1.3 G. DEL CONOCIMIENTO Y DW, INSPECCIÓN, ADMINISTRACIÓN SIN PAPELES, E/S Y NUEVOS PROCEDIMIENTOS
	4.2 ORDENACIÓN Y ADMINISTRACIÓN DE CENTROS DOCENTES, PERSONAL DOCENTE Y UNIVERSIDADES	4.2.1 REGISTRO, FORMACIÓN Y SERVICIOS PARA EL PERSONAL DOCENTE
		4.2.2 OTRAS APLICACIONES PARA CENTROS E INSPECCIÓN EDUCATIVA
		4.2.3 UNIVERSIDADES, ESTUDIOS SUPERIORES Y ENSEÑANZAS DE RÉGIMEN ESPECIAL
4.3 CULTURA, DEPORTE Y POLÍTICA LINGÜÍSTICA	4.3.1 CULTURA DEPORTE Y POLÍTICA LINGÜÍSTICA	

## 5.2.-Gestión de Roles y Permisos

Tal y como refleja el requisito funcional RF-04, se debe realizar una reestructuración de los roles y permisos de la aplicación para poder satisfacer las nuevas necesidades.

La versión anterior de *CATI* tenía un esquema de permisos compuesto por 4 roles de usuario y 4 módulos de control de acceso.

Roles:

- P\_ADMIN: usuarios con rol administrador que pueden acceder a toda la información sin restricciones. Son los únicos que pueden administrar Usuarios.
- P\_CENTRAL: usuarios pertenecientes a la DGTI y censados en la aplicación. Pueden acceder a la información del catálogo sin restricción de visibilidad.
- P\_DEPAR: usuarios pertenecientes a la DGTI y censados en la aplicación. Son como los usuarios P\_CENTRAL con un filtro de visibilidad que limita su visión a una Conselleria.
- P\_USER: rol de los usuarios DGTI no censados en la aplicación que pueden acceder al sistema únicamente para indicar sus datos de experiencia personal.

Restricciones de acceso: para los usuarios P\_CENTRAL y P\_DEPAR, se regula su acceso a Aplicaciones, Contratos y Personal mediante módulos que pueden tener los siguientes valores:

- Sin acceso: no se tiene acceso al módulo.
- Lectura: sólo se tiene acceso a lectura.
- Escritura: se tiene acceso lectura/escritura

Las nuevas exigencias del catálogo han llevado a migrar a un nuevo esquema. La mayor novedad es permitir el acceso lectura a todos los usuarios de la aplicación (siempre que sean miembros de la DGTI). Esto supone que el esquema migre a la siguiente estructura de roles y permisos:

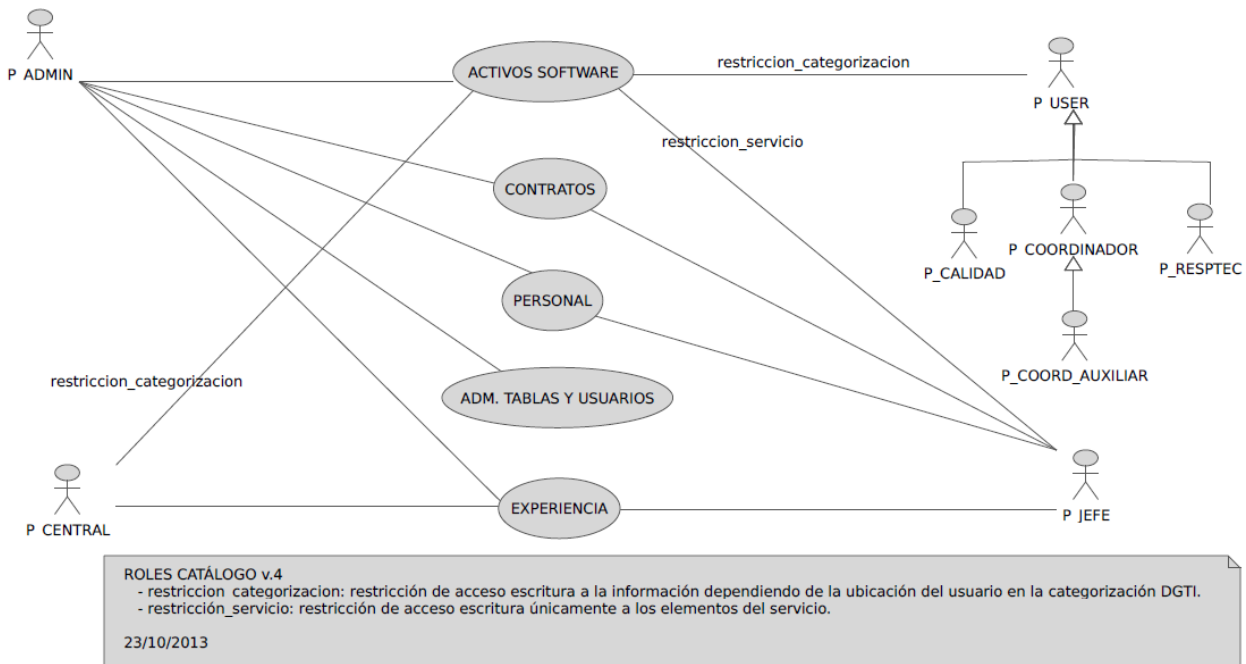


Figura 15: Diagrama de Casos de Uso de alto nivel de CATI.

Tenemos los siguientes roles:

Roles:

- P\_ADMIN: usuarios con rol administrador que pueden acceder a toda la información sin restricciones. Son los únicos que pueden administrar Usuarios y Tablas Maestras.
- P\_CENTRAL: usuarios pertenecientes a la DGTI y censados en la aplicación. Pueden acceder a la información de Activos Software con un filtro relativo a la categorización funcional de la DGTI.
- P\_USER: usuarios no censados de la aplicación. Tienen acceso a lectura a los Activos Software y, acceso a escritura dependiendo de los diferentes roles que pueden asumir:
  - P\_CALIDAD: ser responsable de calidad de la categoría del A.S.
  - P\_RESPTEC: ser responsable técnico del A.S.
  - P\_COORDINADOR: ser coordinador de la categoría, el área o el marco del A.S.
  - P\_COORD\_AUXILIAR: para ayudar a los coordinadores en su trabajo puede delegar permisos en usuarios que tendrán los mismos privilegios que un coordinador en una categoría, área o marco concreto.
- P\_JEFE: usuarios Jefes de Servicios a la DGTI y censados en la aplicación.

Tienen acceso a los Activos Software (con limitación de escritura sobre los A.S. de su servicio). Acceden a Contratos y Personal sin restricciones.

Restricciones de acceso: para los usuarios P\_USER y P\_JEFE, se regula su acceso a Activos Software mediante módulos: que pueden tener los siguientes valores:

- Servicio: permite limitar el acceso escritura de los usuarios P\_JEFE a los A.S.
- Categorización: permite limitar el acceso escritura a los usuarios P\_USER dependiendo de su posición en la categorización funcional de la DGTI.

En cuanto a Casos de Uso, vemos que la aplicación mantiene los tres grandes bloques de información añadiendo una gestión de tablas maestras más completa. A continuación veremos en detalle los dos niveles con mayor cambio: Activos Software y Contratos.

### 5.2.1.-Nivel de Activos Software

Explotando el diagrama a más bajo nivel, podemos estudiar las entidades con más cambios que, como adelantábamos, son los Activos Software y los Contratos.

En primer lugar, la figura 16 muestra las entradas correspondientes a los Activos Software. Hay varios aspectos que conviene destacar. El primer cambio destacable es en el mantenimiento principal de aplicaciones que ha pasado a denominarse mantenimiento de Activos Software. No es el único cambio, ya que como comentábamos anteriormente, también ha sufrido modificaciones relativas a las restricciones de acceso a la información. Ahora todos los usuarios tendrán acceso a lectura limitándose los permisos sólo para la escritura según los criterios Servicio y Categorización.

Bajando en la figura, vemos los mantenimientos correspondientes a la categorización. En ellos se podrán definir los diferentes niveles de la misma y su relación con los Activos Software.

Por último, tal y como se solicitaba en el requisito RF-03, se ha preparado un acceso a la estructura a través de una navegación jerárquica. Esta navegación permitirá al

usuario visualizar de forma rápida e intuitiva tanto los niveles de la estructura como los miembros que la componen a nivel de la DGTI y a nivel de los posibles contratos.

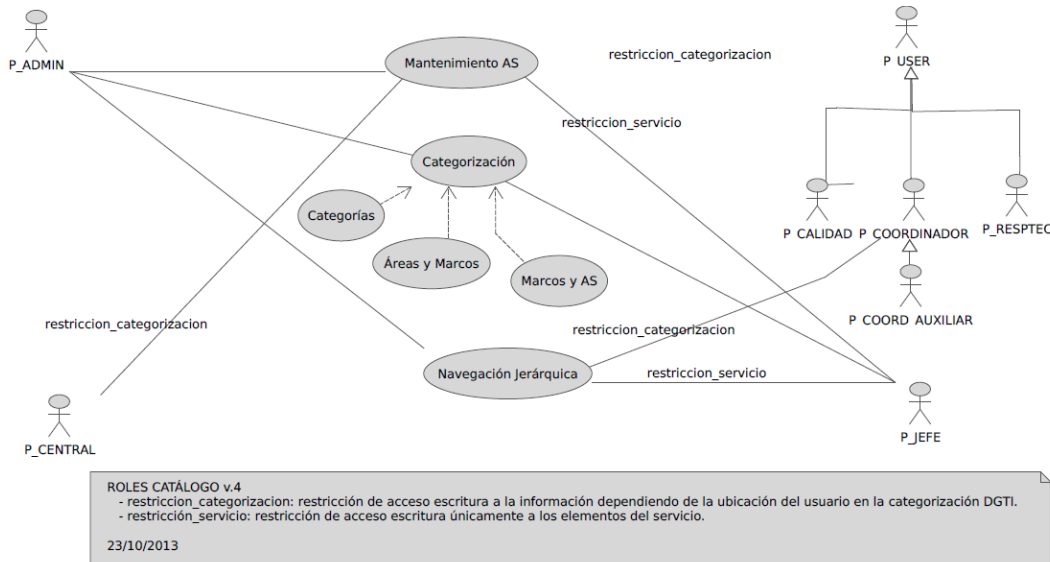


Figura 16: diagrama de Casos de Uso explotando los Activos Software.

### 5.2.2.- Nivel de Contratos

En la figura 17 podemos ver explotado el cambio en la estructura de los contratos en la nueva versión de la aplicación.

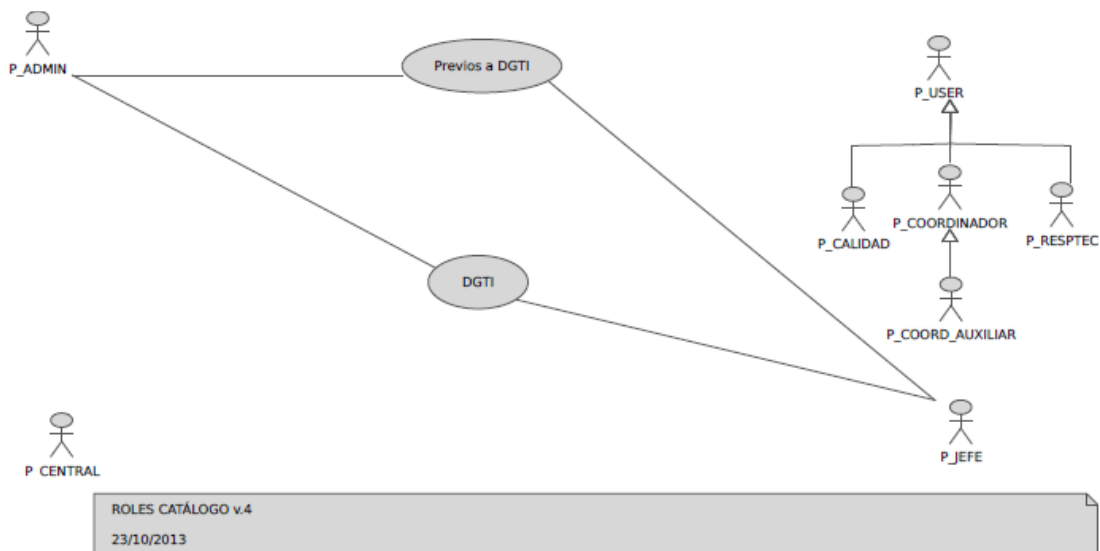


Figura 17: diagrama de Casos de Uso explotando los Contratos

Cabe destacar como se ha respetado la antigua estructura de contratos para evitar una colisión con los contratos anteriores. La opción de acceso a contratos antiguos pasa a llamarse *Previos a DGTI* y la opción *DGTI* corresponderá al mantenimiento de los nuevos contratos por lotes de la Dirección General.

## 6.-DISEÑO

En este apartado vamos a destacar algunos de los procesos implementados durante esta fase para llevar a cabo el proyecto. Uno de los más importantes pasa por la redefinición de la capa de persistencia (esquema relacional) para pasar de un concepto aplicación a un concepto *Configured Item*.

También veremos como se ha rediseñado el esquema relacional y los procesos [ETL](#) que se lanzarán a través de la herramienta *Pentaho Data Integration (Kettle)* para transformar la información. Recordemos, que uno de los requisitos no funcionales, concretamente el RNF-01, requiere la migración a un SGBD Oracle; por lo que estos procesos se han definido como críticos.

Por último resumiremos las decisiones tomadas en relación a la interfaz de usuarios y veremos las medidas aplicadas en materia de seguridad de los sistemas de información.

### **6.1.-Nuevo esquema relacional**

Para entender el proceso tenemos que ver cual es la situación de partida. Para ello contamos con el siguiente esquema origen (figura 18).





## 2. Relación Aplicaciones-Contratos:

- a) Cambio estructural: La licitación de los nuevos contratos de la Dirección General así como su relación con los Activos Software va a tratarse de un modo diferente al que se implementó en esta versión inicial. Por ello se requerirá la creación de una nueva estructura de contratos que se organice por lotes.
- b) Nueva información: Se necesitará información relativa no sólo al adjudicatario de un expediente; sino de los recursos que participan en la ejecución del servicio de dicho expediente.

## 3. Categorización:

- a) Tal y como se explica en la toma de requisitos, una de las necesidades que debe cubrir la aplicación para convertirse en la *CMBD* de Activos Software de la DGTI es poder plasmar la estructura orgánica de estos activos en relación a los contratos y a los *CIs*. Por esta razón, se debe crear una estructura a tres niveles que permita relacionar estas dos entidades.

## 4. Utilización de tablas internas para datos comunes:

- a) Relación con datos de RRHH: en la versión anterior, ya que se trataba de un desarrollo desplegado en un entorno externo y que funcionaba en modo *stand-alone*, se replicaron las entidades relativas a la Guía de Personas. Al cambiar a un entorno corporativo donde ya se tiene acceso a esta información centralizada, se deben eliminar dichas entidades y cambiar las referencias.
- b) Relación con información Orgánica: al igual que en el caso anterior, el sistema replicó las tablas con la información Orgánica de la Generalitat Valenciana. Una vez desplegado en el entorno corporativo, se debe solventar este problema.

Analizados los requisitos y evaluado su impacto en la estructura actual, se decide migrar al esquema que mostramos en la siguiente figura. Podemos observar que los tres grandes “temas” se mantienen en el esquema (Aplicaciones, Contratos y RecursosPropios).

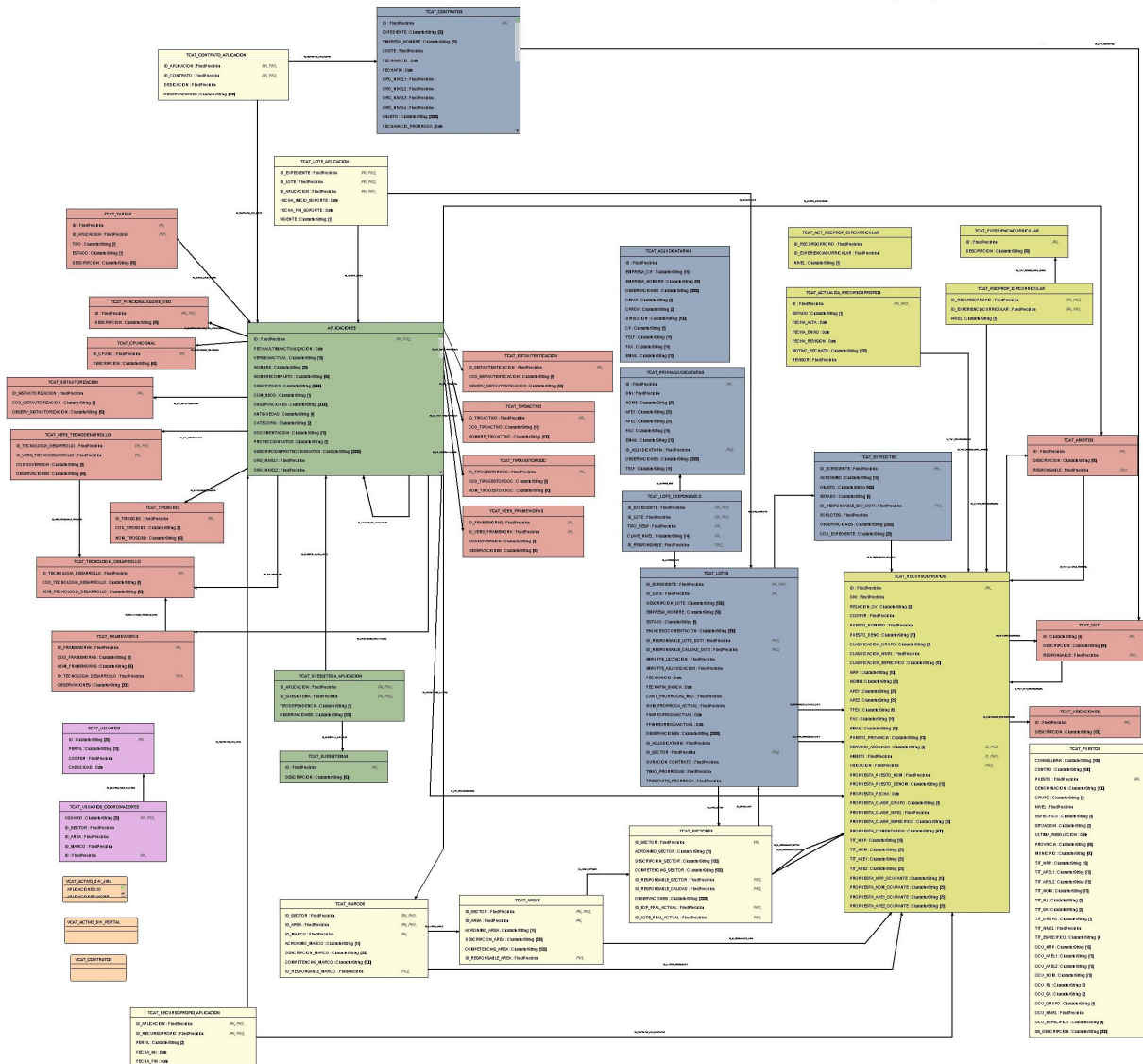


Figura 19: esquema relacional de la nueva versión de CATI.

Para facilitar la lectura, se ha agrupado funcionalmente las entidades con los siguientes colores:

- Color rojo: relativo a tablas maestras que aportan información descriptiva a los elementos.
- Color verde: relativo a los CIs o Aplicaciones Software. Nótese que para que el impacto en la migración sea lo menos posible no se ha modificado el nombre de la entidad principal.
- Color Azul: relativo a contratos. Se puede advertir que hay dos entidades; una corresponderá a los contratos antiguos y otra a las nuevas contrataciones de la

Dirección General. En el siguiente apartado aclararemos este punto.

- Color verde: información relativa a los recursos propios. El cambio a tenido poco impacto sobre esta información como posteriormente veremos.
- Color Rosa/Magenta: relativo a la información de administración y gestión de usuarios y roles.
- Color Naranja: las vistas que se han creado para facilitar la comunicación de otros sistemas (JIRA, Portales, etc).

En un primer análisis de las decisiones de diseño adoptadas, podemos fácilmente observar que se ha hecho especial hincapié en dos aspectos:

1. Minimizar el impacto del cambio en el Sistema existente. Por esta razón se han mantenido nombres de tablas y reutilizado algunos campos ya existentes. Evidentemente, a la hora de aplicar esta premisa, siempre ha habido un compromiso entre la practicidad y las buenas prácticas.
2. Preservar la información histórica: esta premisa se ve claramente reflejada en el mantenimiento de la antigua estructura de gestión de contratos que, actualmente no se adapta a las necesidades del servicio.

A continuación, para comprender en profundidad las decisiones de diseño adoptadas, analizaremos los cambios realizados en las dos familias funcionales que más se han visto afectadas: Contratos y Activos Software.

### 6.1.1.-Contratos

La siguiente imagen (figura 20) muestra un detalle de como se almacena en el nuevo sistema la información relativa a los contratos.

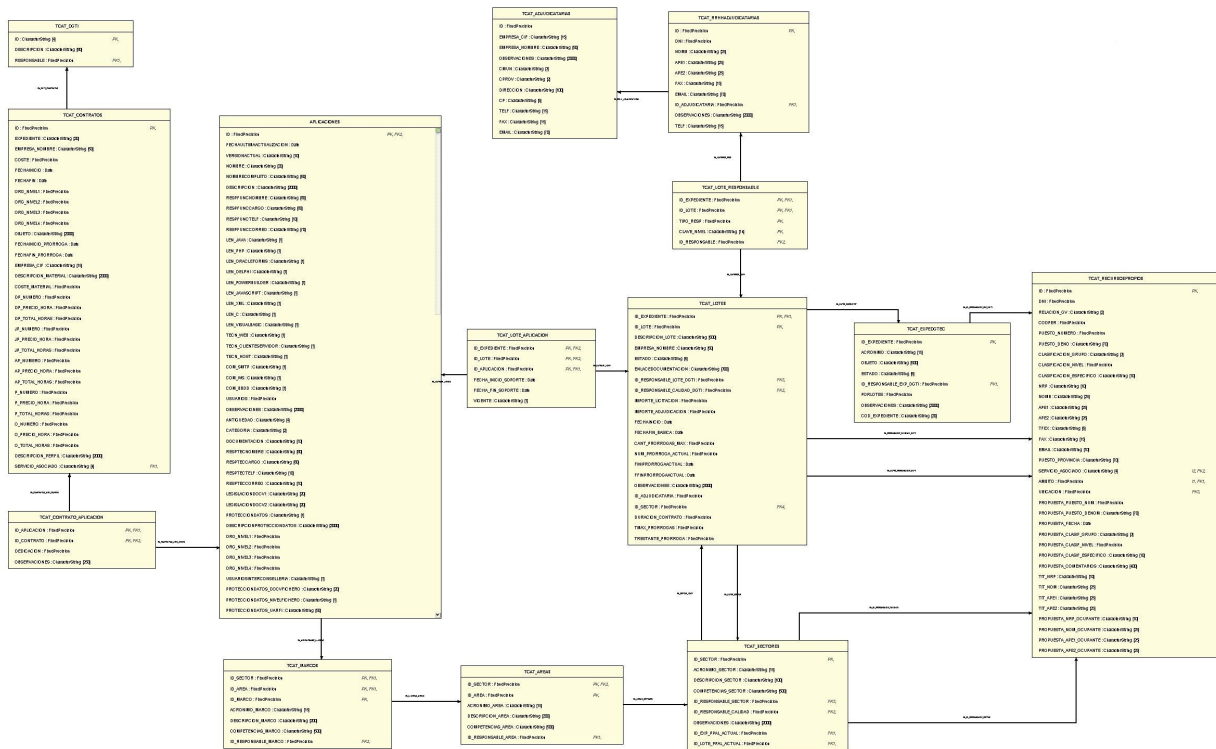


Figura 20: cambios importantes del esquema en relación a Contratos.

Analizamos los cambios más destacados:

- Incorporación de una segunda estructura de contratos: El primer aspecto importante que debemos destacar es la incorporación de la nueva estructura Expediente-Lotes. Esta será la nueva estructura que seguirán los contratos de la DGTI. Se ha respetado la estructura anterior que aparece representada en la parte izquierda del esquema con la entidad Contratos y su relación con los Activos Software (tcats\_contratos\_aplicacion).
- Relación Aplicaciones Lotes: Esta relación nos permitirá agrupar los activos software en los diferentes lotes de los expedientes obteniendo la [Cartera Base](#) de los mismos. Este concepto es importante ya que define el ámbito de mantenimiento de un nuevo contrato.
- Información de los adjudicatarios y sus recursos: Podemos ver como se han creado las entidades necesarias para almacenar esta información.
- Categorización: podemos ver las entidades Sector, Área y Marco correspondientes a la categorización. Destacamos algunos detalles importantes:
  - En primer lugar, al lector le puede extrañar la controversia Categoría/Sector. El primer nivel de de la categorización en un primer momento se llamó sector

- para cambiarse posteriormente por categoría. Este cambio no se ha visto reflejado en el esquema para evitar el impacto que supondría en la planificación. En posteriores fases de *refactoring* se subsanará el problema.
- Todos los activos software están ubicados en un marco (por tanto en una categoría, área y marco) de la estructura funcional de la DGTI.
  - Los lotes de los nuevos expedientes siempre harán referencia, de forma general, a una categoría (tabla `tcat_sectores`) de la clasificación funcional de la DGTI.
  - Responsables. Tal y como se explica en la definición de los pliegos (por ejemplo DESIG2013), tanto la DGTI como la empresa adjudicataria deben aportar representantes en cada uno de los niveles de la categorización a los que el lote de servicio.

Aunque no se vea claramente expresado en el esquema, cabe destacar en este apartado el concepto de contrato vigente para una aplicación. Este contrato será el que se ofrecerá a JIRA para permitir realizar, por ejemplo, la facturación. Los procesos que realizará la aplicación son:

1. Cuando asignamos aplicaciones a un lote, para cada una de ellas realizaremos los siguientes pasos:
  1. Buscaremos el contrato vigente.
    1. Si no tiene, asignaremos como fecha de inicio y fin del soporte a la relación del nuevo lote y la aplicación las fechas del lote.
    2. Si tiene, le asignaremos de fecha inicio del soporte a la relación del nuevo lote y la aplicación la máxima entre la fecha de inicio del lote y la fecha fin del contrato vigente.
  2. Cuando modificamos una fecha de un lote se hará el recálculo para todas las aplicaciones asignadas al lote.
  3. Cuando esporádicamente se requiera, desde la ventana de aplicaciones, se podrá cambiar el contrato vigente a partir de un cambio de las fechas de soporte.

### 6.1.2 Activos Software

En relación a la información de activos los cambios se han centrado en transformar el concepto de Aplicaciones hacia el de *Configured Item*. Para ello, se han creado una serie de tablas maestras que permitirán aumentar la descripción de dicho activo

software. También se han solucionado problemas relativos a campos no normalizados o no utilizados.

En la imagen que adjuntamos a continuación (figura 21) podemos ver los siguientes cambios:

- Creación de tabla maestras: podemos observar que a la tabla de aplicaciones le rodean infinidad de relaciones y nuevas tablas que enriquecen la información existente. Es especialmente interesante el manejo de la información de la tecnología, framework de desarrollo y versión.
- Se han eliminado los campos desrelacionados relativos a lenguajes, tecnología y comunicación.
- Se han eliminado los campos relativos a Responsable Funcional y Técnico ya que sólo se requiere la referencia (acceso a tablas de la Guía de Personas).
- Creación de Campos relativos a la Gestión de la Demanda (comunicación con JIRA y Portal de la Demanda).



## 6.2.-Proceso de migración: ETLs

Para realizar la migración de la base de datos Postgresql a la nueva base de datos Oracle se ha utilizado la herramienta *Pentaho Data Integration (Kettle)* para diseñar unos ETLs. Cabe destacar algunos condicionantes que el lector debe tener en consideración:

1. No se puede por requerimiento del servicio cerrar la aplicación durante un largo periodo de tiempo. Esto significa que la aplicación actual deberá estar funcionando hasta que se ponga en producción la nueva versión. Esto implica que debemos planificar el proceso de migración para poder ser lanzado en un momento concreto del tiempo y, si es posible, de forma desatendida. En todo caso, este requisito es complejo y requerirá de bastante dedicación.
2. Tal y como se ha visto en el punto anterior, los esquemas origen y destino de los datos son distintos. Por ello se necesita realizar una transformación de la información antes de ser ubicada en el destino.
3. La coyuntura actual de la Dirección General ha requerido de la carga de información complementaria a través de hojas de cálculo que han sido elaboradas por responsables de la DGTI. Estas hojas completan la información de los Activos Software y facilitarán la publicación de las carteras base de los expedientes.

Teniendo todo esto en cuenta, a continuación mostramos algunos de los ETLs realizados para migrar la información. En primer lugar vemos el ETL correspondiente a la transformación y carga de la Aplicaciones (figura 22).



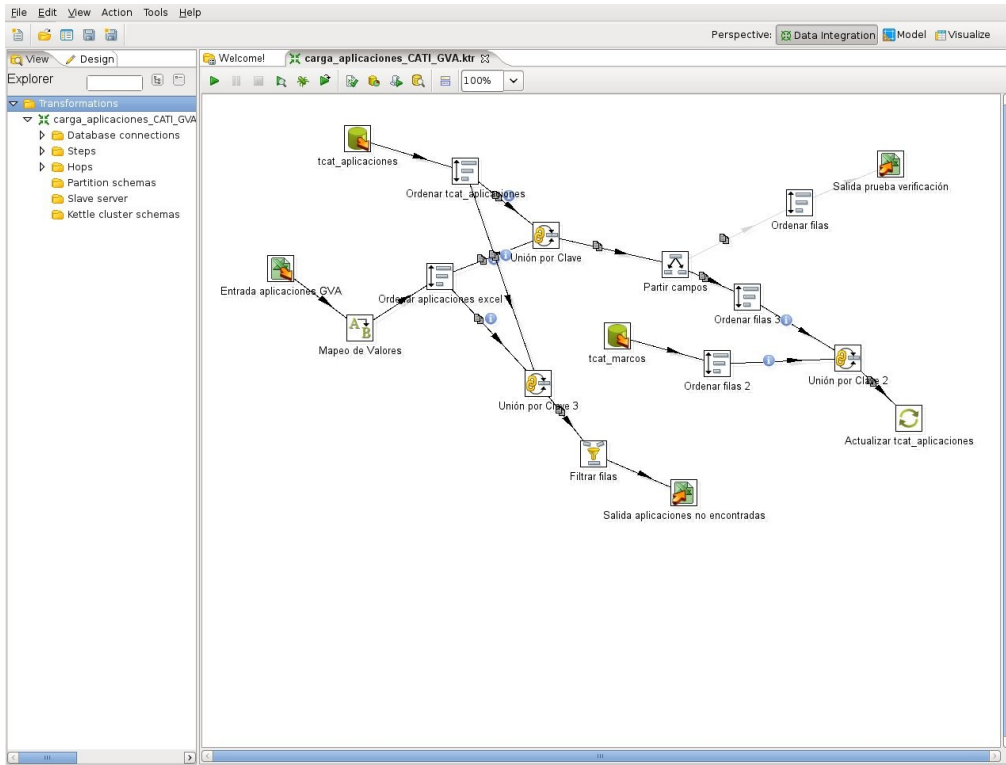


Figura 22: captura de pantalla de la herramienta PDI Keettle con el ETL de migración de Aplicaciones.

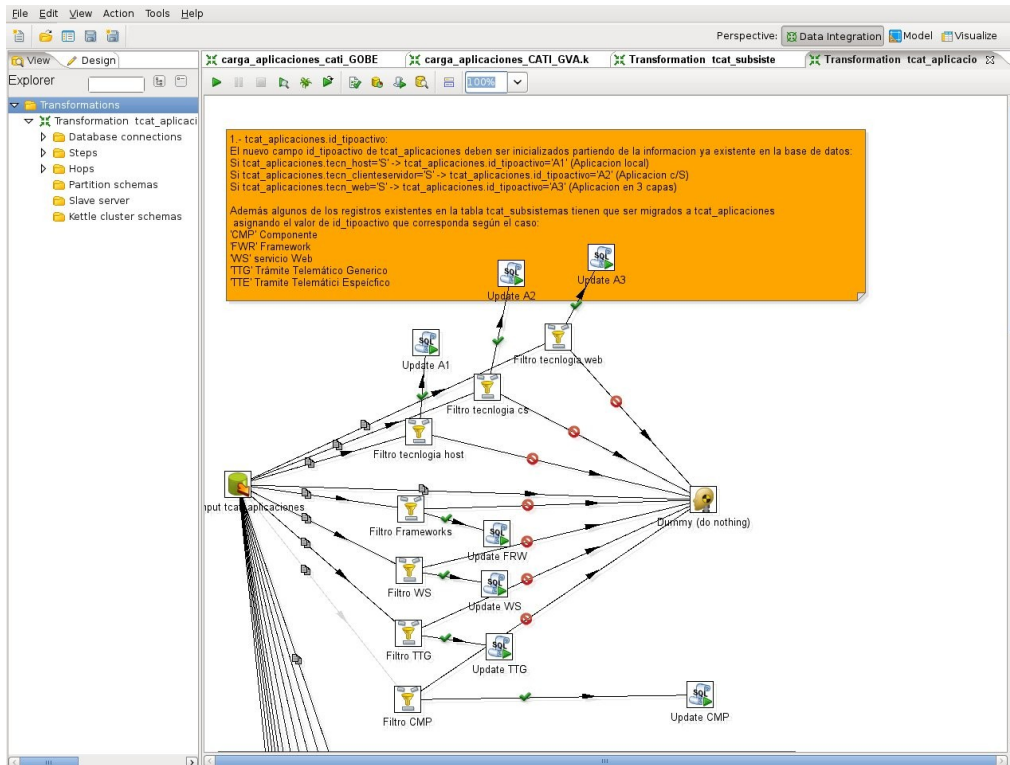


Figura 23: captura de PDI Keettle con el ETL de transformación de Aplicaciones a Activos Software.

El ETL de la figura 23 corresponde a uno de los diseñados para transformar la información existente relativa a los activos software (figura 23). Entre otras cosas, se debe recuperar la información de los lenguajes y *frameworks* anteriormente ubicada en los campos desnormalizados.

Finalmente, presentamos un ejemplo de carga de información a través de una hoja de cálculo (figura 24). Como podemos observar, se realizan varias transformaciones antes de que la información se cargue en la entidad destino.

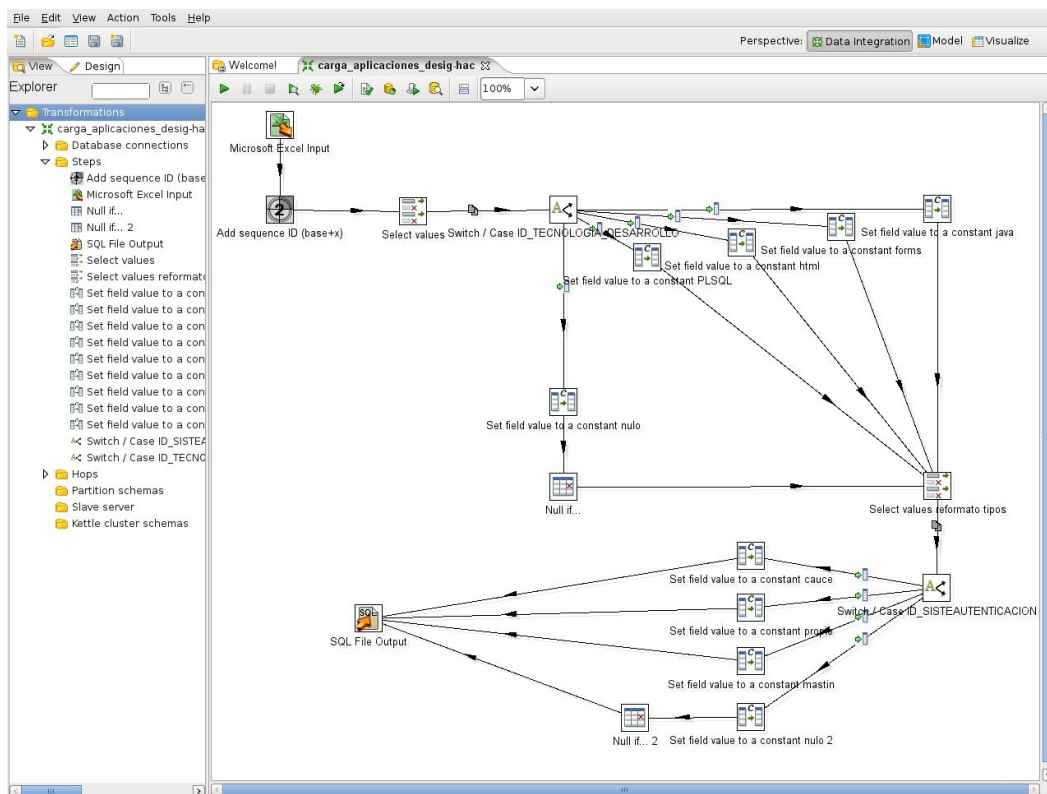


Figura 24: captura de pantalla de *PDI Kettle* con carga desde fuente de hoja de cálculo.

### 6.3.-Decisiones de diseño en la interfaz de usuario

En cuanto al diseño de interfaces se han tomado dos decisiones importantes. En primer lugar, se decidió cambiar el *custom* utilizado para la aplicación. El *custom* hasta ahora era el correspondiente a las aplicaciones de la CITMA. Esta guía de estilo tenía una serie de carencias en cuanto a usabilidad.

La DGTI decidió confeccionar un nuevo estilo para sus aplicaciones más acorde a las

necesidades actuales. Por esta razón, se encargó al equipo de desarrollo de gvHIDRA realizar una nueva propuesta de guía de estilo para las aplicaciones del *framework*. El resultado es un nuevo *custom* con tonos azules y blancos que se distribuye desde la versión 4.0.0 de gvHIDRA. Los responsables de la aplicación decidieron que CATI en su nueva versión tomara este nuevo aspecto.

La segunda decisión importante a efectos de la interfaz era la elección de los patrones para los nuevos mantenimientos a desarrollar. Estos patrones son patrones de presentación. Tal y como dice Roger S. Pressman en su libro [Ingeniería del Software un enfoque práctico](#), “estos patrones abordan como organizar las funciones de control de la interfaz del usuario para una mejor facilidad de uso”. A continuación se muestran los mantenimientos con los patrones de interfaz seleccionados.

Mantenimiento	Patrón	Comentario
Mant. Activos Software	Tabular-Registro	Se mantiene la visualización en formato tabular registro porque es del agrado del usuario funcional.
Mant. Categorías	Tabular-Registro	Se selecciona esta visualización para facilitar la introducción de datos.
Mant. Áreas y Marcos	Maestro (Registro) Detalle (Tabular)	Se selecciona esta visualización para facilitar la introducción de datos. Se facilitará un salto para poder visualizar los activos correspondientes a un marco.
Mant. Marcos y Activos Software	Maestro (Tabular) Detalle (Tabular)	Se selecciona esta visualización para facilitar la introducción de datos correspondientes a la asociación de activos a marcos.
Mant. Contratos DGTI	Maestro (Registro) Detalle (Tabular-Registro)	En el maestro se definirán los expedientes y en el detalle los lotes. Se tendrá que diseñar un salto para poder asignar aplicaciones a los lotes de forma sencilla e intuitiva.
Navegación Jerárquica	Árbol	Se deberá poder modificar para cada nivel los responsables de la DGTI y los responsables de los posibles contratos vinculados con el nodo de la estructura.
Tablas Maestras	Tabular (en su mayor parte)	Se selecciona esta visualización

		para facilitar el mantenimiento por parte de los usuarios administradores
--	--	---

## 6.4.-Esquema de seguridad

Tal y como recogen los requisitos, concretamente el RNF-04, se deben revisar las políticas de seguridad de la aplicación para garantizar un nivel acorde a la información que va a albergar.

Este trabajo se ha dividido en tres fases:

1. Análisis de la seguridad del Framework gvHIDRA.
2. Análisis de la seguridad de la Infraestructura
3. Análisis de la seguridad de la Aplicación

Esta división se debe a que la seguridad es un campo que afecta a todas las capas de un sistema de información y, por tanto, no podemos hablar de ella sin un análisis de las medidas y precauciones que deben tomarse en cada uno de ellas. Una vulnerabilidad en alguna de las capas pondrá en peligro la seguridad de todo el conjunto.

### 6.4.1 Análisis de las medidas de seguridad del Framework gvHIDRA

Como ya sabemos, la aplicación CATI está desarrollada bajo este *framework* de desarrollo libre de la Generalitat Valenciana. Los técnicos responsables del mismo han realizado un estudio de las medidas de seguridad que han aplicado para prevenir los ataques más habituales en las arquitecturas web. En la siguiente tabla podemos ver un resumen de las medidas más importantes que gvHIDRA aporta como prevenciones de seguridad:

Medidas	Descripción	Previene
Escapado de la entrada/salida	Esta medida es básica para la seguridad de los sistemas de información ya que cualquier entrada puede estar contaminada e, incluso, cualquier salida proveniente de una fuente infectada.	Ataques SQL Injection, HTML Injection, Cross-site Scripting (XSS), etc.
Uso de ACL (Access	Las rutas de acceso están gestionadas por el controlador a través de un fichero de rutas válidas.	Ataques por ruta

Control List) o Whitelist	Cualquier acceso fuera de esa lista es denegado.	transversal (.dot-dot-slash).
Protección de la sesión	Los ataques más habituales en el mundo web son los relacionados con la sesión ya que es donde podemos encontrar la información sensible del usuario. Con medidas como la firma con parámetros de la sesión (fingerprint), permitir el acceso a la cookie sólo por http (session.cookie_httponly) o la regeneración del identificador de sesión (session_regenerate_id) se previenen estos ataques	Ataques Session Hijacking o Session Fixation
Evitar accesos indebidos por lenguajes de script	Posiblemente uno de los ataques más de moda en las arquitecturas web. Consiste en envenenar la información contenida en un formulario, bien en la entrada (directo) o en la visualización (diferido) de la información. Con medidas como activar el session.cookie_httponly evitamos que se pueda acceder al identificador de sesión mediante lenguajes de script (típicamente javascript).	Ataques Cross-site Scripting (XSS)

#### 6.4.2.-Análisis de las medidas de seguridad de la Infraestructura

Uno de los requisitos no funcionales que debíamos contemplar en este proyecto es el cambio de la infraestructura de producción de la aplicación (RNF-04). Dado que se ha realizado un traslado de dicha infraestructura, se han realizado una serie de ajustes relativos a la seguridad que a continuación recogemos.

Componente	Medidas	Previene
Ajustes configuración Apache	Forzar SSL	Fuerza que todo el trafico sea a través del puerto 443 cifrado.
	Limitar el acceso a unas IPs	Limita el acceso a la aplicación a un rango de Ips. Esto se hace porque se conoce el rango de Ips que pueden acceder.
	Limitar los directorios de acceso (modo jaula)	Crear una jaula en la aplicación para que, en caso de comprometer su seguridad, no afecte al resto de servicios de la máquina.
	Evitar servir ficheros XML, PDF o ODT	Se evita el peligro de revelar información a través de accesos directos a los ficheros.
	Deshabilitar el listado de los ficheros	Evita revelar información de los archivos a los posibles atacantes.

	Deshabilitar ServerSignature	Evita revelar información del sistema a los posibles atacantes.
	Deshabilitar php en directorios upload	Evita ataques sobre los directorios de upload de una aplicación.
Ajustes configuración PHP	Deshabilitar banner de php	Evita revelar información de la versión de PHP a los posibles atacantes
	No mostrar errores en pantalla	Al deshabilitar el <code>display_errors</code> evitamos que errores de la aplicación aparezcan en pantalla.
	Deshabilitar el registro de variables globales	Aunque viene normalmente deshabilitado esta opción la marcamos. Se evitan ataques por inyección de variables globales.
	Deshabilitar acceso a ficheros externos	Por seguridad conviene deshabilitar el <code>allow_url_fopen</code> y <code>allow_url_include</code> . Previene ataques, por ejemplo, de phishing.
	Forzar canal seguro para el intercambio de la cookie	Al fijar el <code>cookie_secure</code> garantizamos que el tráfico de la <code>cookie</code> se realizará mediante un canal cifrado.
	Evitar que se accede mediante <code>javascript</code> a la <code>cookie</code>	El <code>framework</code> lo activa también. Fijar el parámetro <code>cookie_httponly</code> previene al navegador para que no de el acceso.
Ajustes configuración SGBD	Acceso a la aplicación con usuario no propietario	Para evitar que el impacto de un fallo en la seguridad sea lo menor posible, se proporcionarán a las aplicaciones usuarios sin privilegios de CREATE, DROP o GRANT. La idea será que se cree un usuario que sólo tenga acceso a los objetos del esquema con los permisos SELECT, UPDATE, DELETE e INSERT. Esto supone, tal y como expresan Elmasri y Navathe en su libro <a href="#">Sistemas de Base de Datos</a> un sistema de seguridad a través de mecanismos de seguridad discrecional sencillo y fácil de gestionar por el DBA y el jefe de proyecto.
Ajustes configuración servidor de ficheros	Control de usuarios y grupos	Crear un grupo para la aplicación y a dicho grupo añadir un usuario responsable de la aplicación y el usuario que ejecuta el servidor web .  Los ficheros de la aplicación tendrán estos

		como propietario y grupo. Esto nos permitirá que solo el propietario pueda modificar el código y solo el servidor web pueda ejecutarlo.
--	--	---

### 6.4.3.-Diseño de las medidas de seguridad de a nivel de aplicación

Para completar las medidas anteriormente citadas, tanto a nivel de *framework* como de infraestructura, hemos diseñado algunas revisiones y cambios que deberemos ejecutar en la aplicación. Para garantizar que la calidad del trabajo, aconsejados por el equipo de desarrollo del *framework*, hemos acudido a la página de la [OWASP](#) (*Open Web Application Security Project*) como base de la planificación. También se han consultado fuentes como [Zend PHP 5 Certification Study Guide](#) (en su capítulo de seguridad en PHP) o la documentación de la asignatura [Administración Avanzada de la UOC](#) en su tema relativo a la administración de la seguridad.

#### Debilidades del sistema de validación

Una de las mayores vulnerabilidades de las aplicaciones suele estar en su sistema de validación. Para solventar que esto pueda afectar a la aplicación *CATI* hemos decidido migrar a un sistema de autenticación más seguro. Concretamente, vamos a utilizar la Plataforma de Intermediación de eSirca para acceder al servicio SAFE a través de *WSSecure*. Se ha seleccionado este sistema por dos razones fundamentales:

1. Se trata de un sistema seguro gracias a que utiliza un canal cifrado para la comunicación de los mensajes SOAP.
2. Se trata de un sistema que autentica los usuarios contra el LDAP de la Generalitat Valenciana. Esto permite delegar la autenticación en este sistema ya que todos los usuarios tienen cuenta de correo con las credenciales del LDAP.

#### Lectura de información de Sesión

Se ha concienciado a todos los programadores del peligro que supone realizar una gestión de los valores de autorización a través de campos de pantalla. Se corre el riesgo de ataques por *spoofed form*.

Por ello, se realizará una revisión completa de todas las clases manejadoras y las *views* de todos los mantenimientos de la aplicación para validar que no puede haber ningún

ataque por delegación de autorización a campos de pantalla.

### Ficheros de Documentos Servidos por URL

En la versión anterior, el manual de usuario se servía directamente mediante un acceso HTTP. Esto supone un riesgo de seguridad por revelación de información a un posible atacante. Por esta razón, se va cambiar el acceso a estos documentos para que sólo puedan acceder a ellos los usuarios con sesión activa (usuarios validados).

Del mismo modo, se ha revisado la metainformación que aparece en los documentos ofrecidos a los usuarios para que no se puedan deducir cuentas o usuarios privilegiados.

### Tipado de los campos de pantalla

Aunque como hemos visto gvHIDRA realiza un escapado de las entradas y salidas que recibe; es importante facilitar el trabajo tipando los campos de las pantallas.

El trabajo en este aspecto se centrará en la ventana que más se va a ver afectada por los cambios, la ventana de Aplicaciones (o activos software). Se revisará la clase manejadora para que todos los campos cuenten con su correspondiente tipo gvHIDRA.

### Debilidades del control de acceso a rutas

Como hemos visto anteriormente, una de las fortalezas del *framework* en cuanto a seguridad es limitar los flujos de acceso a dos ficheros. Esto no tiene efecto si se crean *scripts* autónomos para la aplicación.

En la versión anterior estos *scripts* existían para sincronizar las copias de las tablas comunes. Al cambiar de ubicación la infraestructura y centralizar los datos ya no serán necesarios por lo que se eliminarán estos *scripts* evitando problemas de seguridad.



## 7.-IMPLEMENTACIÓN

En este punto vamos a detallar los trabajos realizados en la implementación de la nueva versión de la aplicación *CATI*. Veremos como se han desarrollado los cambios en los mantenimientos principales para recoger las necesidades de nueva información. Del mismo modo, veremos una ventana de las nuevas tablas maestras que, como veíamos en el apartado anterior, se han creado para alimentar el concepto de activo software.

También repasaremos las nuevas ventanas realizadas para cumplir las necesidades de navegación de los usuarios por la nueva estructura funcional. Adicionalmente, revisaremos algunos de los filtros de listados y sus salidas.

Finalmente, haremos referencia a dos trabajos de peso en esta fase. En primer lugar las labores de implementación realizadas para poder mantener las relaciones con el resto de componentes que forman parte de la gestión del servicio. Concretamente, revisaremos las implementaciones realizadas para conectar con JIRA. En segundo lugar, analizaremos los trabajos relativos al *refactoring* del código generado.

### **7.1.-Mantenimientos**

Este apartado ha supuesto la mayor parte de los esfuerzos en desarrollo. Se han tenido que realizar trabajos de limpieza de código para eliminar campos y relaciones que ya no existían así como incorporar las nuevas.

#### 7.1.1.-Activos Software

El mantenimiento más afectado en cuanto a trabajos de transformación ha sido el mantenimiento de aplicaciones que ha pasado a ser mantenimiento de Activos Software. Caben destacar los siguientes trabajos:

- Creación de nuevos componentes. Listas, campos, ventanas de selección relativas a la nueva información que se requiere.
- Eliminación de campos no existentes y de toda la lógica vinculada a ellos. Por ejemplo la de los lenguajes de programación
- Relación con tablas comunes. Este punto ha sido especialmente delicado ya que

se han tenido que eliminar referencias textuales por el código de los elementos en las tablas comunes.

A continuación, en la figura 25, mostramos como ejemplo la ventana relativa a una aplicación y a los datos de Gestión de la demanda.

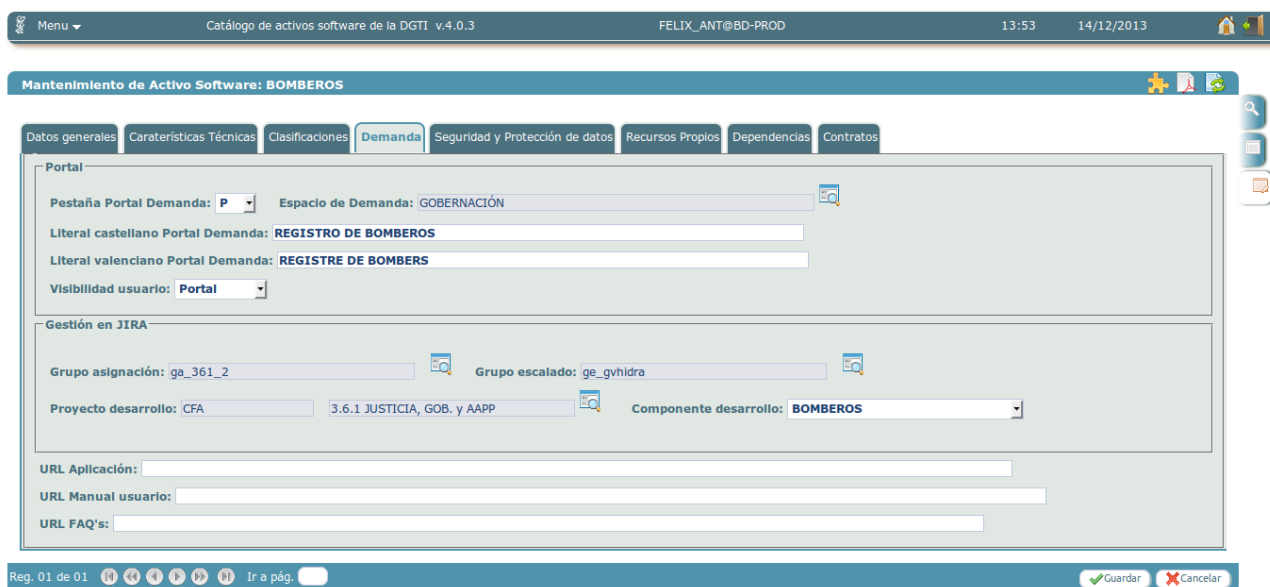


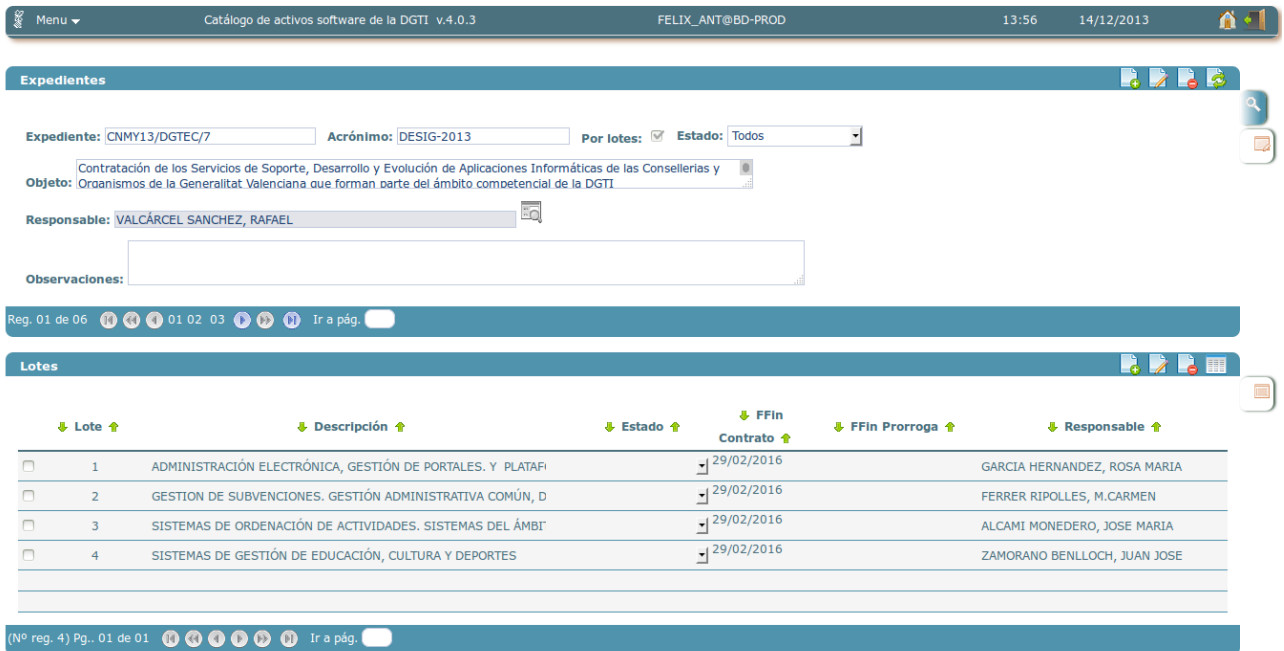
Figura 25: captura de pantalla de Activos Software, la nueva solapa de Gestión de la Demanda.

### 7.1.2.-Contratos

Dada la decisión de crear una estructura nueva para los contratos, en la fase de implementación esto ha supuesto la creación de nuevos mantenimientos. Para estos mantenimientos, vista la distribución de la información, se ha optado por la creación de mantenimientos siguiendo el patrón de diseño Maestro-Detalle. A continuación, en la figura 26, mostramos una captura con la ventana del mantenimiento para los nuevos contratos DGTI.

Como se aprecia en la imagen, podemos comprobar como en el panel superior se podrán mantener los Expedientes (vemos en la captura el expediente correspondiente a DESIG2013) y en el detalle los distintos lotes del expediente.

Cabe destacar en esta ventana la implementación de un salto para poder asociar activos software a cada uno de los lotes. Podemos apreciar en la imagen como en la parte superior del panel detalle tenemos un *botontooltip* que parametriza esta invocación.



Expediente: CNMY13/DGTEC/7    Acrónimo: DESIG-2013    Por lotes:     Estado: Todos

Objeto: Contratación de los Servicios de Soporte, Desarrollo y Evolución de Aplicaciones Informáticas de las Consellerías y Organismos de la Generalitat Valenciana que forman parte del ámbito competencial de la DGTI

Responsable: VALCÁRCEL SANCHEZ, RAFAEL

Observaciones:

---

Reg. 01 de 06    Ir a pág.

**Lotes**

Lote	Descripción	Estado	FFin Contrato	FFin Prorroga	Responsable
1	ADMINISTRACIÓN ELECTRÓNICA, GESTIÓN DE PORTALES. Y PLATAFI	✓	29/02/2016		GARCIA HERNANDEZ, ROSA MARIA
2	GESTION DE SUBVENCIONES. GESTIÓN ADMINISTRATIVA COMÚN, D	✓	29/02/2016		FERRER RIPOLLES, M.CARMEN
3	SISTEMAS DE ORDENACIÓN DE ACTIVIDADES. SISTEMAS DEL ÁMBI	✓	29/02/2016		ALCAMI MONEDERO, JOSE MARIA
4	SISTEMAS DE GESTIÓN DE EDUCACIÓN, CULTURA Y DEPORTES	✓	29/02/2016		ZAMORANO BENLLOCH, JUAN JOSE

(Nº reg. 4) Pg. 01 de 01    Ir a pág.

Figura 26: captura de pantalla de nuevos contratos de la DGTI por Expediente-Lote.

### 7.1.3.-Tablas maestras

Para facilitar la labor de los administradores, se han creado una serie de mantenimiento sobre las nuevas tablas maestras que enriquecen los conceptos de CIs. En su mayor parte, se han implementado en un patrón tabular para facilitar el mantenimiento de la información por parte de los administradores. En la siguiente figura podemos ver el mantenimiento de Tipos de Activos Software y el menú con las diferentes entradas correspondientes al resto de tablas maestras.



Activos Sw

- Personal
- Contratos
- Ámbitos
- Mantenimiento de Ámbitos.
- Gestores Documentales
- Sistemas de Autenticación
- Sistemas de Autorización
- Sistemas Gestores BD
- Subsistemas
- Tecnologías de Desarrollo
- Tipo Activo Sw
- Funcionalidades Geomáticas

Descripción	Acrónimo
Aplicaci	A2
Aplicaci	A3
Aplicaci	A1
Componente	CMP
CUADRO DE MANDO	CMD
Framework	FWR
Geomática	GIS
Intranet	INT
Portal Institucional	PTI
Servicio Web	WS

(Nº reg. 12) Pag. 01 de 02    Ir a pág.

Figura 27: captura de pantalla de la nueva ventana de tipos de Activo Software.

## 7.2.-Categorización

La implementación de este requerimiento en lo que al diseño de la ventana se requiere ha sido uno de los pasos más complicados. Los usuarios requerían un acceso a través de la estructura con diferentes visiones dependiendo del nivel en el que se encontraran.

Por ello, se ha utilizado un patrón de diseño en árbol que, en la estructura de gvHIDRA, permite definir el nodo mostrado en cada nivel. Esto ha permitido que los usuarios puedan visualizar/modificar por un lado la información de los responsables de cada nivel de la estructura funcional y los contratos que la afectan. En el último nodo, se puede ver la información relativa a los activos software del marco. En la figura 28 podemos observar el árbol haciendo referencia al activo software *gvHIDRA*.



Figura 28: captura de pantalla de la pantalla de navegación jerárquica.

## 7.3.-Listados

Como es evidente, una vez cargada la información de los activos y contratos, los usuarios finales necesitan explotarla. Para ello se han creado una serie de listados que permiten obtener la información por diferentes criterios.

Este punto ha sido especialmente crítico dadas las necesidades del servicio de obtener los listados de las carteras base para publicar los nuevos expedientes. A continuación mostramos la ventana de listados con los diferentes formatos de salida creados para satisfacer las necesidades.



Figura 29: captura de pantalla de los filtros de listados para Activos Software. Vemos las distintas salidas.

El resultado de las salidas son listados PDF como el que mostramos en la figura 30. Concretamente, este listado corresponde a la cartera base del Lote 1 del expediente DESIG2013.

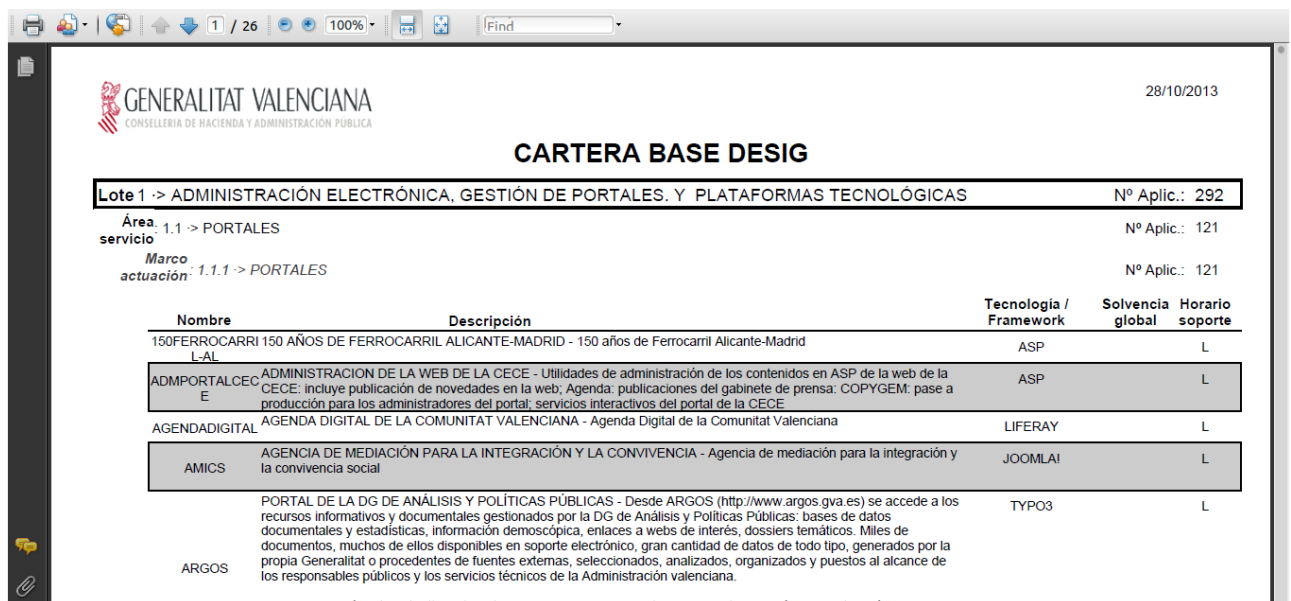


Figura 30: captura de pantalla del PDF resultante del listado de Cartera Base.

### 7.4.-Sinergias: JIRA, Portal y Gestión de la Demanda

Para poder relacionarse con otros subsistemas, CATI ha preparado una serie de vistas que permiten a dichos subsistemas obtener la información. Concretamente hemos dispuesto:

- VCAT\_ACTIVO\_SW\_JIRA: Vista que ofrece la información que requiere JIRA de un activo software.
- VCAT\_ACTIVO\_SW\_PORTAL: Vista que ofrece la información que requiere el Portal de gestión de la demanda de un activo software.
- VCAT\_CONTRATOS: Vista utilizada por JIRA como por el propio catálogo para mostrar la información de los contratos antiguos y nuevos.

Por otro lado, ha surgido la necesidad de obtener información de los grupos de escalado y asignación para los activos software de JIRA. Para ello se han implementado una serie de clases que haciendo uso de los servicios API REST que provee JIRA obtienen la información.

En el anexo 1 adjuntamos dos clases que permiten obtener los grupos de asignación de JIRA en una ventana de selección de gvHIDRA. La primera (Clase Jira.php) muestra como se conecta el sistema al servicios REST de JIRA. La segunda (Clase WS\_JiraGrupoAsignacion.php) integra la información en una *SelectionWindow* de gvHIDRA.

## 7.5.-Refactoring

Una vez concluida la implementación de los factores más críticos y tras la validación y puesta en producción de las primeras versiones, se inició un proceso de *refactoring*. Este proceso, traducido como “refabricación” en algunos textos, se puede definir como la técnica de reorganización que simplifica el diseño (o código) de un componente sin cambiar su función o comportamiento (Roger S. Pressman en su libro [Ingeniería del Software](#)).

Al ser el desarrollo de este proyecto tan crítico en tiempo y con un “modelo de desarrollo en bazar”, se tomaron algunas determinaciones que han requerido de la realización de este proceso de mejora continua.

La primera decisión que se tomó en la fase de diseño y que ha hecho necesario que se efectúe a posteriori un proceso de esta naturaleza es la correspondiente al proceso de migración al nuevo esquema. Debido a la importancia de la información existente, la complejidad de la transformación y al cambio de SGBD; se decidió mantener algunos

campos antiguos antes de ser borrados. El ejemplo más evidente era el de los campos pertenecientes a los lenguajes de programación.

Este proceso se ha destinado a limpiar todas las referencias a dichos campos y clarificar las clases manejadoras. No cabe duda de que la decisión de mantener los campos fue correcta dada la magnitud del cambio; pero no podemos eludir que también ha supuesto un foco de pequeños problemas. El más trascendente se produjo al no eliminar los campos relativos a los responsables funcionales y técnicos. El nuevo sistema obtiene dicha información de las tablas comunes a través de un identificador; pero algunas ventanas y listados seguían leyendo la información de los campos antiguos. Este proceso sirvió para revisar y solucionar todos estos problemas.

Otra de las tareas que se acometió en este proceso fue la de estandarización de componentes y comportamientos de ventanas. Los usuarios finales detectaron que componentes que hacían referencia al mismo conjunto de datos se comportaban de diferente manera en diferentes partes de la aplicación. Este problema fue debido a la presión en los tiempos de desarrollo y una pequeña descoordinación entre el equipo debida a la deslocalización de sus miembros. Una vez detectado, fue sencillo homogeneizar el comportamiento de dichos componentes. En el caso de los mantenimientos el problema se ha resuelto de forma similar.

Finalmente, este proceso ha concluido con una revisión completa del código de todas las clases manejadoras para validar que se ha seguido un estilo homogéneo a la hora de codificar. Se ha hecho uso de algunas de las recomendaciones que para ello ofrece el libro [Clean Code](#) de *Robert C. Martin*. En forma resumida, citamos algunas de las tareas que se han realizado:

- Se ha repasado el nombre de las variables, clases, métodos y funciones. En este caso lo más importante ha sido validar que el nombre correspondía con su destino. Como notación, se ha seguido la norma marcada en la mayor parte de las aplicaciones gvHIDRA.
- Uso homogéneo de la indentación y las llaves. Esto facilita la lectura de código.
- Método cortos y concretos. Sólo realizan una tarea para que sea más sencillo saber que es lo que hacen.
- Se ha revisado la estructura de los bloques condicionales. Se ha eliminado el uso de los bloques condicionales con ?.

- Revisión de los comentarios. Sólo se añaden comentarios cuando es necesario, no como norma general. Un comentario debe aportar algo que con la simple lectura del código no podamos entender.
- Refinamiento sucesivo. Dada la naturaleza del proyecto, versión tras versión se ha realizado un refinamiento sucesivo del código resultado.



## 8.-CONCLUSIONES Y TRABAJOS FUTUROS

En este apartado vamos a hablar de las conclusiones a las que hemos llegado tras la puesta en producción, de la importancia del software libre en este proyecto y de los nuevos retos que se debe acometer la aplicación en un futuro próximo.

En el transcurso de esta memoria, hemos revisado todos los aspectos concernientes al proyecto. En el capítulo dos revisamos los objetivos que nos marcábamos como meta en esta travesía. Tras presentar el entorno tecnológico en el tercer capítulo, revisamos los requisitos y el análisis del impacto de los mismos en los dos capítulos siguientes. En los capítulos sexto y séptimo hemos desglosado los trabajos relativos al diseño e implementación de la solución. Finalmente, en este capítulo octavo vamos a hablar de conclusiones.

En cuanto a conclusiones, no cabe duda de que el balance global del proyecto ha sido un éxito. Se han cumplido las expectativas de los usuarios funcionales y, lo más importante, se ha llegado en plazo para satisfacer las necesidades de la Dirección General. La aplicación ha estado operativa al 100% cumpliendo todos los requisitos solicitados en el plazo esperado. Actualmente, es una de las piezas activas que componen el nuevo sistema de gestión del servicio de la Dirección General. Todo ello se ha conseguido gracias al gran esfuerzo realizado por el equipo de trabajo. No ha sido sencillo, y el proyecto ha tenido dificultades durante su ejecución que se han superado satisfactoriamente gracias a una excelente labor.

Una de los principales dificultades ha sido la eterna problemática en los proyectos TIC: los requisitos cambiantes. Es normal que se produjera este efecto tratándose de un proyecto que debe ser una de las piezas angulares de una organización que está en fase de continua evolución en estos momentos. El equipo de trabajo tomó la determinación de utilizar una estrategia de desarrollo evolutivo siguiendo las pautas de la metodología SCRUM. Se han realizado *sprints* cortos, consensuados con las necesidades del servicio y contando con un alto grado de implicación del usuario final. De esta forma, se ha logrado satisfacer las necesidades de la Dirección General a tiempo y se han detectado de forma rápida cambios en los requisitos limitando su impacto en la planificación. Cabe destacar en este punto la gran ayuda que ha supuesto contar con un responsable funcional como M<sup>a</sup> Arrate Baquedano que en todo momento se ha prestado a colaborar

y ha sido, sin duda, un miembro más en el equipo de desarrollo.

Otro problema, consecuencia del anterior, son los cambios en las planificaciones. Este inconveniente se ha resuelto gracias al extraordinario equipo de desarrollo con el que se ha contado y la coordinación del mismo a través de la forja de desarrollo. Desde un primer momento la elección fue un equipo reducido, deslocalizado, de personal altamente cualificado y con un perfil multidisciplinar. En los momentos de dificultad, las dificultades se han resuelto gracias a una buena gestión del problema, el apoyo del equipo y el esfuerzo colectivo. Es aquí donde decisiones atípicas como la de no incorporar más recursos en cuellos de botella han resultado cruciales. Tal y como anunciaba *Frederick Brooks* en su famoso libro [\*The Mythical Man-Month\*](#) añadir más recursos en un cuello de botella puede suponer retrasar más el proyecto. Por esta razón, cuando se han producido estos problemas de planificación ha sido la implicación del equipo la que los ha resuelto mediante un sobreesfuerzo encomiable.

Sobre como ha estado presente el software libre en este proyecto no cabe duda de que ha sido el motor principal del mismo. Evidentemente, la estructura de desarrollo centralizada en la forja de gvHIDRA y sus servicios a favorecido en gran medida a que los esfuerzos de todos los miembros del proyecto se hayan encauzado en una misma dirección. El modelo de trabajo colaborativo ha sido fundamental para el éxito del proyecto. La comunidad de desarrolladores gvHIDRA también ha supuesto una fuente de apoyo y ayuda permanente; hubiera sido imposible realizar cambios como el de la validación o el estudio de seguridad sin su ayuda. También han formado parte importante del proyecto herramientas como *Pentaho Data Integration (Keettle)* que ha sido fundamental para poder realizar todo el trabajo de transformación.

Definitivamente, el software libre ha sido el protagonista principal de este proyecto para alegría de todos los participantes. Es evidente, tal y como aparece en el anexo de prescripciones técnicas del pliego DESIG2013, que la DGTI quiere fortalecer el uso de software libre y/o de fuentes abiertas para el desarrollo de sistemas de información en su ámbito competencial. Pero cabe destacar, y es de vital importancia este aspecto, que en este proyecto no se ha seleccionado esta opción por una oportunidad coyuntural ni política; se ha optado por estas herramientas por ser las más adecuadas para alcanzar los objetivos con éxito. El resultado final del proyecto así lo demuestra y supone un espaldarazo para el uso de software libre en el seno de la Generalitat

Valenciana.

En cuanto a los trabajos futuros, son muchos los retos que le quedan al proyecto de la aplicación *CATI*. Uno de los principales retos supone la mejora de la clasificación de los activos software y la incorporación en su definición del entorno tecnológico. Es fundamental poder conocer la infraestructura que necesita cada *CI* para poder realizar una gestión eficiente desde la *CMDB*. Para ello, se debe hacer un esfuerzo de análisis no sólo en la definición de la estructura sino también en el estudio de las existentes ya que repositorios como *gvCESTA* se deberán enlazar con esta información.

Otro de los trabajos futuros que está pendiente de la disponibilidad de recursos es relativo a la seguridad. La seguridad, es un campo dinámico y no puede descuidarse a pesar de los esfuerzos que se han realizado para mejorarla. El concepto seguridad 100% no existe, por ello se tiene planificada la realización de una auditoría por parte del *CSIRT-CV* que detecte puntos débiles y zonas de mejora.

No podemos olvidar en esta lista un nuevo proceso de *refactoring*. Se han detectado debilidades en el diseño y hay una serie de problemas de nomenclatura. Uno de los requisitos futuros es aumentar la jerarquía con un cuarto nivel y será, cuando se acometa este cambio, cuando se podrá realizar la labor de actualización de nomenclaturas.

En conclusión, el proyecto ha sido un éxito por ser capaz de satisfacer las necesidades funcionales y las restricciones de plazo de la Dirección General. Gran parte del mérito de este éxito se lo lleva la buena gestión de un excepcional equipo humano; concepto que se debe conservar como la principal lección aprendida.

En cuanto a mi recompensa personal, he podido aplicar gran parte de los conocimientos adquiridos durante los dos años de realización del Master de Software Libre de la UOC. Desde la la definición y replanificación de un plan de trabajo visto en la asignatura Implantación de Sistemas de Software Libre, a la programación de scripts shell y configuración de Apache visto en las asignaturas de Administración de Sistemas pasando por las recomendaciones y enseñanzas vistas en la asignatura de Software Libre en la Administración Pública. Ha sido una experiencia altamente gratificante ya que gracias a estos conocimientos hoy me siento mejor profesional.

## 9.-ANEXOS

### 9.1.-Clases de Acceso a JIRA

#### Clase Jira.php

```

<?php
/* gvHIDRA. Herramienta Integral de Desarrollo Rápido de Aplicaciones de la Generalitat Valenciana
 *
 * Copyright (C) 2006 Generalitat Valenciana.
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License
 * as published by the Free Software Foundation; either version 2
 * of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307,USA.
 *
 * For more information, contact:
 *
 * Generalitat Valenciana
 * Conselleria d'Infraestructures i Transport
 * Av. Blasco Ibáñez, 50
 * 46010 VALENCIA
 * SPAIN
 * +34 96386 24 83
 * gvhidra@gva.es
 * www.gvhidra.org
 */

/**
 * Clase Manejadora Jira
 *
 * Clase para conectarse con JIRA para interactuar con este sistema
 *
 * @autor Toni Félix
 * @version 1.0
 * @license http://opensource.org/licenses/gpl-2.0.php GNU Public License v.2
 */

class Jira {

    private $username = '';
    private $password = '';
    private $credential = null;
    private $urlBase = '';
    private $header;

    public function __construct() {

        $conf = ConfigFramework::getConfig();
        $g_jira = $conf->getDSN('g_jira');

        $this->urlBase = $g_jira['uriWSDL'];
        $this->username = $g_jira['username'];
        $this->password = $g_jira['password'];

        $this->credentials = base64_encode($this->username.':'.$this->password);
        $this->headers = array(
            'Accept: application/json',
            'Content-Type: application/json',
            'Authorization: Basic '.$this->credentials
        );
    }

    public function getProjects() {

        $url = $this->urlBase.'/rest/api/2/project';
        $curl = curl_init();

        curl_setopt($curl, CURLOPT_USERPWD, "$this->username:$this->password");
        curl_setopt($curl, CURLOPT_URL, $url);
        curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($curl, CURLOPT_HTTPHEADER, $this->headers);
        curl_setopt($curl, CURLOPT_FOLLOWLOCATION, 1);
        curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);
        curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);
    }
}

```

```

    try {
        $result = curl_exec($curl);
        $projects = json_decode($result);
    }
    catch(Exception $e) {
        return -1;
    }
    return $projects;
}

public function getComponents($project) {
    if(empty($project))
        return array();

    $url = $this->urlBase.'/rest/api/2/project/'.$project.'/components';
    $curl = curl_init();

    curl_setopt($curl, CURLOPT_USERPWD, "$this->username:$this->password");
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $this->headers);
    curl_setopt($curl, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);

    try {
        $result = curl_exec($curl);
        $components = json_decode($result);
    }
    catch(Exception $e) {
        return -1;
    }
    return $components;
}

public function getGroups($query) {
    $url = $this->urlBase.'/rest/api/2/groups/picker?query='.$query;
    $curl = curl_init();

    curl_setopt($curl, CURLOPT_USERPWD, "$this->username:$this->password");
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $this->headers);
    curl_setopt($curl, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);

    try {
        $result = curl_exec($curl);
        $aux = json_decode($result);
        $groups = $aux->groups;
    }
    catch(Exception $e) {
        return -1;
    }
    return $groups;
}

}
?>

```

## Clase WS\_JiraGrupoAsignacion.php

```

<?php
/**
 * Fuente de datos de grupos
 *
 * Esta fuente de SelectionWindow permite obtener los grupos de JIRA
 *
 * @author Toni Félix
 * $Revision: 1.1.4.1 $
 */

class WS_JiraGrupoAsignacion implements gvHidraSelectionWindow_Source {
    private $jira;

    public function __construct() {
        $this->jira = new Jira();
    }

    public function search($value,$dependence,$dependenceType) {
        //Los grupos de asignación se identifican con ga_
    }
}

```

```
$query='ga_';

$groups = $this->jira->getGroups($query);
$res = array();
foreach($groups as $group) {
    //Busqueda: si hay value buscamos en name
    if(empty($value) OR (strpos(strtoupper($group->name),strtoupper($value))!==false))
        $res[] = array('name'=>$group->name);
}
return $res;
} //Metodo search
} //Fin WS_JiraGruposAsignacion
?>
```

## 10.-GLOSARIO

- Cartera Base: en relación a los nuevos contratos de la DGTI se refiere a las aplicaciones que se deberán de mantener con un lote concreto.
- Categorización: estructura jerárquica funcional de los activos software de la DGTI. Servirá para gestionar los activos software dentro de la Dirección General.
- CATI: Acrónimo de la aplicación de Catálogo de Activos Software. Es la base central de este Proyecto Final de Master.
- CI: Configured Item. Término que proviene del mundo ITIL. A groso modo hace referencia a todo componente que ha de ser gestionado por la organización TI.
- CIT: Conselleria de Infraestructuras y Transporte. Conselleria que inicia un proceso de migración a software de fuentes abiertas y, entre sus proyectos está gvHIDRA.
- CMDB: Término ITIL para referirse a la Base de Datos de Gestión de la Configuración. Es un repositorio de información donde se relacionan todos los componentes de un sistema de información, ya sean hardware, software, documentación, etc.
- CRUD: Create, Read, Update y Delete. Término que se refiere a la realización de las cuatro operaciones de forma atómica.
- Custom: Referido a gvHIDRA se refiere a la capa funcional correspondiente a la organización. En esta capa se caracteriza por la posibilidad de configurar el aspecto de la interfaz de usuario.
- DESIG: *Desarrollo de los Sistemas de Información de la Generalitat*. Expediente tramitado por la Dirección General de Tecnologías de la Información para la contratación de los Servicios de Soporte, Desarrollo y Evolución de las Aplicaciones Informáticas bajo su ámbito competencial.
- DGTI: Dirección General de Tecnologías de la Información de la Generalitat Valenciana.
- ETL: Extract, Transform and Load: procesos que permiten extraer información de diferentes soportes y cargarlas en otros. Puede realizar transformaciones en mitad del proceso para facilitar la carga, muy habitual en procesos de normalización de datos.
- gvPONTIS: Proyecto de migración a Sistemas Abiertos de la Conselleria de Infraestructuras y Transporte. (<http://www.gvpontis.gva.es>)

- ITIL: *Information Technology Infrastructure Library*. Es un conjunto de conceptos y prácticas para la gestión de servicios de tecnologías de la información, el desarrollo de tecnologías de la información y las operaciones relacionadas con la misma en general.
- JIRA: se trata de una software de la compañía Atlassian basada en web para el seguimiento de errores, de incidentes y para la gestión operativa de proyectos. Es la herramienta seleccionada por la DGTI para realizar estas funciones.
- MVC: arquitectura Modelo Vista Controlador. Esta arquitectura se caracteriza por la separación de capas de negocio y presentación a través de un controlador que realiza las funciones de *dispatcher*.
- PDI Kettle: Pentaho Data Integration. Se trata de un componente de la compañía Pentaho que permite la implementación de forma gráfica de ETL.
- RA: Responsable de Ámbito. Personas encargadas de trasladar el conocimiento de los activos software, contratos y RRHH de cada una de las Consellerias a la primera versión de CATI. Normalmente eran ex jefes de Servicio de dichas Consellerias.
- RAD: Rapid Application Development
- SOI: Servicio de Organización e Informática de la CIT. Servicio responsable de la creación de gvHIDRA.



## 11.-BIBLIOGRAFÍA

- ¿Qué es la DGTI?: <http://www.dgti.gva.es/que-es>
- ¿Qué es gvHIDRA?: Dossier.
- Manual Técnico Infraestructura de Desarrollo v1. Autor: Software Colaborativo.
- Manual de Bienvenida a la Forja de gvHIDRA.
- DECRETO 119/2011, de 9 de septiembre, del Consell de la Generalitat Valenciana: [http://www.docv.gva.es/portal/ficha\\_disposicion\\_pc.jsp?sig=009549/2011&L=1](http://www.docv.gva.es/portal/ficha_disposicion_pc.jsp?sig=009549/2011&L=1)
- DECRETO 20/2013, de 25 de enero, del Consell de la Generalitat Valenciana: [http://www.docv.gva.es/portal/ficha\\_disposicion\\_pc.jsp?sig=000865/2013&L=1](http://www.docv.gva.es/portal/ficha_disposicion_pc.jsp?sig=000865/2013&L=1)
- Expediente CNMY13/DGTEC/7: “SERVICIO EN INFORMÁTICA: PROYECTO DESIG-2013 (DESARROLLO DE SISTEMAS DE INFORMACIÓN DE LA GENERALITAT)”
- La Catedral y el Bazar de Eric S. Raymond. <http://biblioweb.sindominio.net/telematica/catedral.html>
- Ingeniería del Software, enfoque práctico. Roger S. Pressman. Editorial McGrawHill. ISBN 970-10-5473-3
- Clean Code, a handbook of agile software craftsmanship de Robert C. Martin. Editorial Prentice Hall. ISBN 0-13-235088-2
- Recomendaciones de Seguridad. Toni Félix y Nacho Santos. Documento de recomendaciones de seguridad para las aplicaciones desarrolladas con gvHIDRA.
- Sistemas de Bases de Datos, conceptos fundamentales de Elmasri y Editorial Navathe. Addison-Wesley Iberoamericana
- The Mythical Man-Month. Frederick P. Brooks, JR. Editorial Addison Wesley. ISBN 0-201-83595-9
- Zend PHP 5 Certification Study Guide de Davey Shafik y Ben Ramsey. Editorial php|architect's. ISBN 0-9738621-4-9
- Administración de Seguridad: Josep Jorba Esteve (PID\_00174429). UOC.
- Open Web Application Security Project (OWASP) <https://www.owasp.org>

## 12.-LICENCIA

### GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

#### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title

Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate,

this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.