



MPWC: Manage Projects, Workers and Contacts

Máster en Software Libre

Roger Sicart Rams

Gregorio Robles Martínez

Javier Martín Fernández

Versión:	231
Última revisión	16/12/2012

Copyright (C) 2012 Roger Sicart Rams.

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 "GNU Free Documentation License".

Resumen

Sonicon Systems es una startup de Barcelona dedicada al desarrollo de software a medida. Creada en 2009, sus fundadores no han parado de implementar soluciones artesanas, muchas de éstas haciendo uso de nuevas tecnologías como Liferay. Hasta el momento se ha usado su potencial solamente para llevar a cabo proyectos para terceros, pero hace unos meses decidieron empezar a implantar dicha tecnología en la propia empresa para poder explotar todo su potencial internamente. De ahí nace la necesidad de elaborar la base de una herramienta propia que ayude a gestionar diferentes procesos internos, como por ejemplo la gestión de personal, proyectos o clientes y proveedores.

Manage Projects, Workers, and Contacts, también conocido por sus siglas MPWC, pretende empezar a llenar el vacío producido por esa necesidad. El proyecto se divide en cinco partes, que son un *theme* y cuatro *portlets*. El *theme*, llamado Mpwc Cooler, se encarga de la parte estética. El resto se han organizado de la siguiente manera: un *portlet* de gestión de trabajadores -Mpwc Worker-, uno de gestión de proyectos -Mpwc Project-, uno de gestión de clientes y proveedores -Mpwc Contact- y para acabar, un *portlet* que incluye la lógica de acceso a datos necesaria para los tres módulos anteriores. Con esta estructura se consigue un proyecto modular, permitiendo modificar una de las partes sin tener que compilar las tres restantes. Cada uno de los *portlets* permite realizar acciones como altas, bajas, modificaciones, búsquedas o listados de su respectivo ámbito.

Para realizar el proyecto se han utilizado únicamente herramientas de software libre, desde el sistema operativo empleado como la gestión del código fuente, tanto para escribir documentación como código. Adicionalmente se ha seguido una metodología iterativa, haciendo uso de plataformas en línea que ayudan a gestionar proyectos de software libre y proporcionan herramientas de comunicación o gestión de código para la comunidad.

Manage Projects, Workers and Contacts ha sido un proyecto corto, intenso y duro, pues la curva de aprendizaje en Liferay es acentuada y la duración muy limitada, pero a la par interesante e instructivo. Pasen y vean.

Tabla de Contenidos

1.Introducción.....	5
2.Objetivos.....	6
3.Planificación.....	7
3.1 Metodología.....	7
3.2 Listado de tareas.....	7
3.3 Diagrama de Gannt.....	9
4.Tecnologías relacionadas.....	10
5.Herramientas utilizadas para el desarrollo.....	14
6.Análisis.....	18
6.1 Requisitos.....	18
6.2 Casos de uso.....	19
7.Diseño e implementación.....	24
7.1 Arquitectura del sistema.....	24
7.2 interfaz de usuario.....	25
7.3 Base de datos.....	28
7.3.1 Introducción a Service Builder.....	28
7.3.2 Diagrama ER.....	29
7.3.3 Diseño de las tablas.....	31
7.3.4 Archivo de configuración de servicios: service.xml.....	32
7.4 Diagrama de clases.....	36
7.5 Decisiones y cambios en la implementación.....	37
7.6 Estado de la implementación.....	40
8.Conclusiones y trabajo futuro.....	41
9.Bibliografía.....	44
10.Agradecimientos.....	45
11.Anexo.....	46
11.1 Manual de configuración y uso.....	46
11.1.1 Configuración de la organización.....	46
11.1.2 Configuración de los roles.....	47
11.1.3 Instalación de los portlets y el theme.....	51
11.1.4 Asignación de permisos.....	54
11.1.5 Creación de un usuario administrador.....	58
11.1.6 Uso de Mpowc Worker.....	59
11.1.7 Uso de Mpowc Project.....	63
11.1.8 Uso de Mpowc Contact.....	66
11.2 GNU Free Documentation License.....	68

1. Introducción

El proyecto nace de la necesidad de implementar una herramienta de gestión a medida para la empresa Sonicon Systems S.L., dado que la herramienta actual no se adapta completamente a los requerimientos y en algunos casos es compleja o dispone de demasiada información. A grandes rasgos, se quiere desarrollar una herramienta propia para la gestión de varios procesos internos, y para ello, se inicia este proyecto con el objetivo de implementar una base a complementar y extender con el tiempo. Además, se ha optado por trabajar con una tecnología moderna y en expansión como es Liferay, una plataforma que permite desarrollar aplicaciones de empresa de diferentes tipos basados en Java -como son los *portlets*, *themes*, *layouts*, *hooks*, etc.- integrándolas todas ellas en un mismo gestor de aplicaciones.

Como punto adicional cabe destacar que existe una especial motivación por parte de la empresa en empezar a utilizar esta tecnología en el propio negocio, a parte de proporcionar soluciones para terceros basadas en ésta.

A lo largo del documento pretendo explicar el ciclo de vida del proyecto, pero al tratarse de un proyecto de software libre solamente se detallarán los aspectos de las iteraciones iniciales, desde la planificación hasta la primera versión estable. A partir de este punto sólo puede haber previsiones de futuro.

En primer lugar se verán los objetivos que se quieren cumplir, de esta manera podremos entender mejor el alcance del proyecto y hasta dónde se quiere llegar. Una vez definidos, pasaremos a ver la planificación temporal, con todos los hitos fijados en el calendario, a excepción de aquellos puntuales o que puedan surgir a última hora. A continuación se podrá leer una introducción acerca de las tecnologías relacionadas con el proyecto, así como de las herramientas utilizadas para llevarlo a cabo, en el capítulo consecutivo. En el apartado 6 veremos el análisis, con la toma de requisitos y su evolución a los casos de uso. Posteriormente seguiremos con el diseño de la solución, que incluye un esquema de la arquitectura del sistema, además de algunos esbozos relacionadas con la interfaz del usuario, el diagrama de clases o el de entidad-relación. En la última parte del documento, se explican las decisiones y cambios realizados durante el desarrollo del proyecto, las expectativas de futuro y las conclusiones a las que he llegado al acabar estos meses de trabajo intenso.

2. Objetivos

Así pues, el proyecto consiste en desarrollar varios portlets que facilitarán la gestión de datos internos de la empresa. Para concretar más, tenemos:

1. gestión de trabajadores
2. gestión de clientes y proveedores
3. gestión de proyectos

Además, se llevarán a cabo también los siguientes puntos:

- creación de un documento funcional
- configuración del portal Liferay, sus usuarios y roles, y documentación del proceso
- creación de una base de datos -PostgreSQL- para Liferay y los módulos a programar
- creación de un *theme* y un *layout* específicos para el proyecto

A título personal añadiré que mis objetivos son aprender a desarrollar *portlets* para Liferay, adquiriendo una base sólida de conocimientos tanto en lo teórico como en lo práctico.

En lo primero, me baso en el estudio de dos libros referentes en el ámbito, *Portlets in Action* [1] acerca de los *portlets* en general, y *Liferay in Action* [2] acerca del entorno Liferay en particular.

Asimismo, en el campo práctico, me sumergiré en un entorno de desarrollo real como es Sonicon Systems S.L., rodeándome de profesionales ejemplares con una amplia experiencia en la tecnología citada, pudiendo así cuestionar personalmente sus decisiones de diseño e implementación de otros proyectos ya existentes.

Adicionalmente, si en el periodo establecido para llevar a cabo estos *portlets* finalizara muy rápidamente, procedería a ampliar funcionalidades con los *portlets* siguientes:

1. gestión de facturación
2. gestión financiera
3. gestión de presupuestos

Esto se llevaría a cabo únicamente en el caso citado, pues está contemplado a largo plazo como extensión del proyecto original.

3. Planificación

3.1 Metodología

A lo largo de todo el proyecto me basaré en una metodología de desarrollo iterativa [3], llevando a cabo la primera iteración y dejando las siguientes para las futuras versiones del código. Se trata de una metodología flexible, pues permite realizar cambios atendiendo las necesidades de los usuarios o clientes, así como integrar nuevos complementos o técnicas descubiertas por los programadores a lo largo de la vida del proyecto.

Las fases de una metodología iterativa son: planificación, toma de requisitos, análisis, diseño, implementación, pruebas, implantación, evaluación y vuelta a empezar. Lo que se pretende con esta metodología es conseguir una versión funcional cuanto antes, e ir mejorando y refinando poco a poco la base. Dadas estas características, pienso que es la aproximación más adecuada para este proyecto.

Metodología de desarrollo iterativa

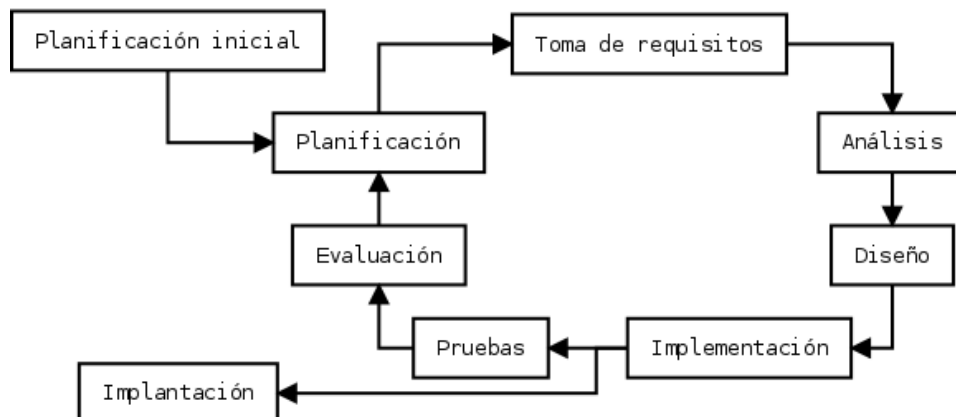


Illustration 1

3.2 Listado de tareas

01-10-2012

- TFM - Plan de trabajo

05-10-2012

- TFM - Documento funcional

07-10-2012

- TFM - Creación de una base de datos -PostgreSQL- para Liferay y los módulos a programar
- TFM - Resumen esquemático

11-10-2012

- TFM - Documento de configuración del portal Liferay, sus usuarios y roles, y documentación del proceso

31-10-2012

- TFM - Final *portlet* gestión trabajadores
- TFM - Final *theme* y *layout*

25-11-2012

- TFM - Final *portlet* gestión proyectos

23-12-2012

- TFM - Final *portlet* gestión clientes y proveedores

06-01-2013

- TFM - Resultado final

13-01-2013

- TFM - Memoria final

3.3 Diagrama de Gannt

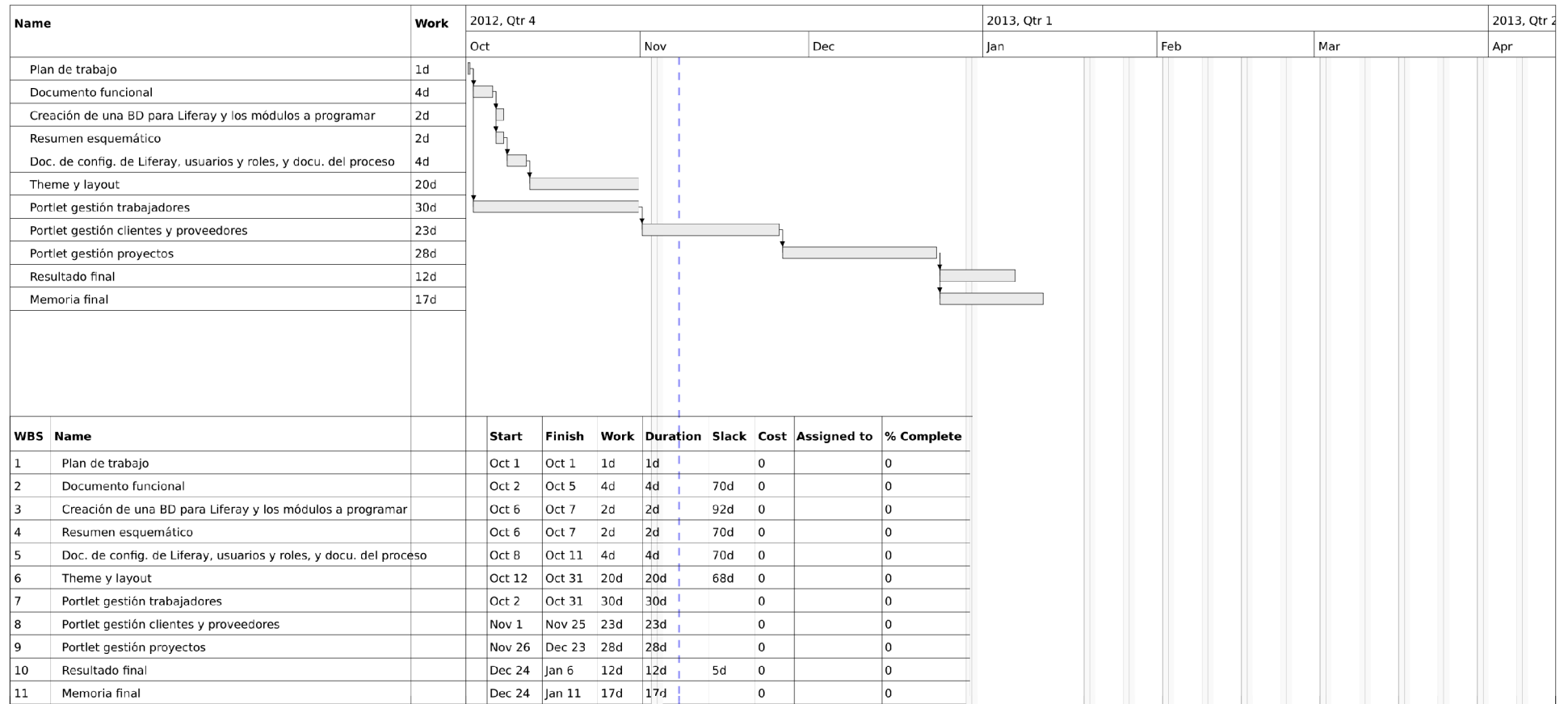


Illustration 2

4. Tecnologías relacionadas

PostgreSQL

PostgreSQL¹ es un sistema gestor de bases de datos relacionales de código abierto multiplataforma. Se trata de un sistema completamente transaccional, licenciado con *PostgreSQL License*, una licencia libre permisiva derivada de la popular *BSD license*.

Permite trabajar con esquemas, desarrollar procedimientos almacenados, extender los tipos de datos, programar *triggers*, índices y secuencias, entre otras cosas, además de la gran cantidad de extensiones con las que se pueden ampliar sus funcionalidades.

La versión utilizada es la 8.4.

Java

Se trata de uno de los lenguajes de programación más populares y más utilizados de la historia. Java² es un lenguaje orientado a objetos (OOP) multiplataforma con el que se pueden desarrollar todo tipo de proyectos: aplicaciones para ordenador, cliente-servidor, aplicaciones en el ámbito web – *applets*, *servlets*, *portlets*, etc – o aplicaciones embebidas para dispositivos son algunos ejemplos.

Este lenguaje de programación se concibió para cumplir con 5 objetivos principales:

- simple, orientado a objetos y familiar
- robusto y seguro
- independiente de la arquitectura del sistema y portable
- alto rendimiento
- interpretado, dinámico y multihilo

En sus inicios fue creado por James Gosling en Sun Microsystems y actualmente pertenece y lo mantiene Oracle.

La versión utilizada en este proyecto es Java SE 6.

1 03/11/2012 PostgreSQL. <<http://www.postgresql.org/>>

2 23/09/2012 Java. <<http://www.oracle.com/technetwork/java/javase/downloads/index.html>>

Portlets

Son componentes modulares de una interfaz de usuario que proporciona contenido específico, el cual puede ser un servicio o datos proporcionados por un sistema de información. Estos generan fragmentos de lenguaje de marcas (*markup language*) para presentar su contenido a los usuarios de un portal, sistema dónde éstos son ejecutados. Son modulares porque cada *portlet* es responsable de generar su propia de interfaz de usuario en un portal.

Existen dos estándares que reúnen las especificaciones: JSF-168³, que fue la primera versión, y JSF-286⁴, segunda y actual versión. Ésta última es una mejora de la primera versión y es la utilizada para este proyecto.

Liferay Portal

Es un portal de código libre y abierto escrito en Java. Tiene una licencia dual, es decir, dispone de una versión *community* con licencia LGPL y una versión *enterprise* con licencia propietaria. Se utiliza para implementar aplicaciones web, gestores de contenido, herramientas de comunicación y otras utilidades enfocadas al mundo profesional. Los portales ejecutan simultáneamente múltiples *portlets*, explicados en el párrafo anterior, los cuales son unidades funcionales con las que los portales se construyen.

Liferay Portal⁵ se distribuye con diferentes servidores de portal, como son Glassfish, Geronimo, Jboss, Jetty o Tomcat, e incluye múltiples *portlets* preinstalados que se pueden adaptar y configurar a medida. Para llevar a cabo el proyecto utilizaré Liferay Portal 6.1 CE GA2, que es la versión *community* anteriormente comentada, con el servidor Tomcat integrado.

Themes

Los *themes* o temas son componentes utilizados para transformar la parte visual y estética de un portal. Éstos se pueden desplegar en caliente y utilizarlos instantáneamente. También permiten modificar estilos y código sin riesgo de romper nada, pues se pueden heredar los estilos de los temas básicos muy fácilmente y retocar

3 06/11/2012 JSF-168. <<http://www.jcp.org/en/jsr/detail?id=168>>

4 06/11/2012 JSF-286. <<http://www.jcp.org/en/jsr/detail?id=286>>

5 03/12/2012 Liferay Portal. <<http://www.liferay.com/>>

solamente las partes que nos interesan.

Layouts

Son componentes con los que podemos editar la distribución de los *portlets* en el portal, también se conocen con el nombre de plantillas o *templates*.

Jquery

Es una librería Javascript que simplifica el desarrollo de código del *frontend*, tanto para lógica como para efectos visuales. Permite vincular elementos a eventos, acceder y modificar de forma fácil los elementos de una página, etc. Jquery⁶ está licenciada con MIT License.

JqGrid

Se trata de una extensión para *jquery* que se utiliza para mostrar tablas con datos, y poder ordenarlos, filtrarlos, etc. JqGrid⁷ es muy práctica a la hora de imprimir resultados de búsqueda de una base de datos. Su licencia es dual, está licenciada con GPL y con MIT, permitiendo así su integración con todo tipo de proyectos.

Alloy UI

Actualmente la mayor parte de proyectos web integran HTML, CSS, Javascript o simplemente los tres combinados, pues juntos nos permiten desarrollar potentes soluciones con formularios, estilos, manejo de datos dinámicos, etc. El lenguaje de marcas HTML proporciona la base o lienzo para las otras dos tecnologías. CSS, por otro nos permite modelar la parte visual con estilos. Para acabar, Javascript nos permite añadir interacción en los elementos de la página. Aunque todas estas herramientas se han usado -y se siguen usando- por separado durante mucho tiempo, desde la comunidad de Liferay se ha desarrollado una API que las combina para facilitar su uso a los desarrolladores. Alloy UI⁸ permite generar código automáticamente, mediante el uso de su

6 14/11/2012 Jquery. <<http://jquery.com/>>

7 14/11/2012 jqGrid. <<http://www.trirand.net/>>

8 05/12/2012 Alloy UI. <<http://www.liferay.com/community/liferay-projects/alloy-ui/overview>>

biblioteca de *tags*, y dinamizar la interacción entre elementos de las páginas mediante su API Javascript (basada en YUI3, el *framework* de Yahoo), añadiendo eventos de manera una similar a jQuery.

5. Herramientas utilizadas para el desarrollo

pgAdmin

Es un proyecto de software libre que sirve para gestionar PostgreSQL de forma gráfica. La versión 3 está licenciada con PostgreSQL license, que es la versión utilizada para gestionar la base de datos de Liferay Portal y del proyecto. PgAdmin⁹ es una herramienta muy completa que permite llevar a cabo operaciones gráficamente o mediante comandos y *scripts*.

Eclipse

Es un entorno de desarrollo de software multilenguaje y abierto, licenciado con su propia licencia llamada Eclipse Public License. Está disponible para múltiples plataformas, entre ellas GNU/Linux y MS Windows. Adicionalmente, Eclipse¹⁰ permite ampliar sus funcionalidades de una forma muy completa mediante extensiones, con las que se puede adaptar el entorno a medida de cualquier equipo de programación.

Para desarrollar los portlets y otros componentes del proyecto utilizo la versión 4.2, con el nombre en clave *Juno*.

Liferay Plugins SDK

Se trata de un kit de desarrollo (SDK) de software creado específicamente para programar complementos de Liferay, como son los *portlets*, *themes* y *layouts*. Este kit está compuesto de *scripts* de línea de comandos que automatizan la creación del esqueleto de cada tipo de complemento, esto es, la estructura de directorios y ficheros así como su contenido básico. Esta herramienta se puede descargar desde la página oficial de Liferay Portal. Para el proyecto se usará la versión 6.1 CE GA2, acorde con la versión del portal.

Liferay IDE

Se trata de un conjunto de extensiones de Eclipse que nos proporciona una interfaz gráfica para el uso de Liferay Plugins SDK. Integra totalmente el desarrollo de

9 18/11/2012 pgAdmin. <<http://www.pgadmin.org/>>

10 30/11/2012 Eclipse. <<http://www.eclipse.org/>>

complementos para Liferay en el entorno de desarrollo Eclipse. Se puede descargar desde la página oficial de Liferay Portal o mediante el asistente de instalación de extensiones de Eclipse.

Git

Es un sistema de gestión de código distribuido. Éste fue creado por Linus Torvalds en un inicio, con el objetivo de proporcionar una nueva herramienta de gestión de código para el desarrollo del kernel de Linux. Los últimos años se ha extendido exponencialmente.

Este tipo de herramientas es especialmente útil, pues permite trabajar simultáneamente a equipos de desarrollo de programas. Además, deben permitir:

- guardar las diferentes versiones de cada fichero para poder volver atrás (*revert*)
- trabajar con diferentes ramas de desarrollo
- automatizar la fusión de las múltiples copias de trabajo (*merge*)

En Git¹¹, cada copia de trabajo es un repositorio completo, olvidando la estructura clásica cliente-servidor. De esta manera, los repositorios se clonan de máquina a máquina con todo el historial de cambios y no dependen de la disponibilidad del servidor: se puede trabajar y hacer *commit* tantas veces como se quiera en la copia local y enviar los cambios a otra máquina o servidor cuando nosotros tengamos disponibilidad. Otra de las principales diferencias con los sistemas de gestión de código son las diferencias entre versiones: Git realiza *snapshots* o fotografías de estado en lugar de guardar un listado de cambios cronológicos para cada archivo.

Para el proyecto utilizo la versión 1.7.2.5. Además también utilizo una extensión para Eclipse llamada EGit, con la que se pueden llevar a cabo las operaciones de forma gráfica, en ocasiones con ayuda de un asistente. Se puede descargar desde la página oficial de Eclipse o mediante el asistente de instalación de extensiones de éste.

Github

Es una plataforma web dedicada al *hosting* de proyectos de software que utiliza Git para gestionar el código de éstos. Ofrece las herramientas siguientes:

11 22/11/2012 Git. <<http://git-scm.com/>>

- *hosting* de código y otros archivos mediante Git, con acceso HTTPS o Git sobre SSH
- gestión y control de código mediante Git
- seguimiento de errores (pestaña *Issues* del repositorio)
- herramientas de colaboración (pestaña *Wiki*)
- estadísticas con gráficos (pestaña *Graphs*)

Con Github¹² disponemos de una plataforma muy completa y moderna, que permite alojar gratuitamente ilimitados proyectos libres -como es el caso de este- y que proporciona servicios adicionales o la posibilidad de disponer de repositorios para proyectos propietarios contratando un plan privado.

Los enlaces a los repositorios del proyecto, desde dónde podemos descargar una copia fresca del código, son los siguientes:

- ➔ Mpwc portlet: <https://github.com/rsicart/mpwc>
- ➔ Mpwc-worker portlet: <https://github.com/rsicart/mpwc-worker>
- ➔ Mpwc-project portlet: <https://github.com/rsicart/mpwc-project>
- ➔ Mpwc-cooler theme: <https://github.com/rsicart/mpwc-cooler>

Google Groups

Es un servicio gratuito proporcionado por Google, donde podemos crear un grupo de comunicación. En el caso de este proyecto es una lista de correo electrónico, una de las herramientas más simples y fáciles de usar para comunicarse con un grupo de personas.

Las características principales de una lista de correo son:

- almacenamiento de suscriptores (personas interesadas en el tema de la lista que dan su dirección de correo electrónico para apuntarse)
- los mensajes enviados por los suscriptores son enviados a todos los miembros de la lista

12 04/12/2012 Github. <<https://github.com/>>

Al tener interfaz web, Google Groups¹³ combina lo mejor de una lista de correo y un foro, pues podemos visualizar los temas de discusión fácilmente, accediendo a la página principal del grupo.

El grupo creado para las posibles colaboraciones del proyecto lo podemos encontrar en:

→ <https://groups.google.com/forum/#!forum/mpwc-dev>

GNU/Linux Debian

Debian¹⁴ es un sistema operativo multiplataforma de código libre, muy popular y utilizado, que no ha parado de crecer desde su fecha de nacimiento en 1993. Está compuesto de paquetes licenciados mayoritariamente con GPL y otras licencias libres compatibles con ésta. Este proyecto está desarrollado y mantenido por un equipo de alrededor de 3000 voluntarios esparcidos por todo el mundo.

13 04/12/2012 Google Groups. <<https://groups.google.com/>>

14 03/12/2012 GNU/Linux Debian. <<http://www.debian.org/>>

6. Análisis

6.1 Requisitos

Requisitos no funcionales

1. Uso de PostgreSQL como sistema gestor de bases de datos.
2. Uso de tecnologías Liferay.
3. Desarrollo en módulos diferentes, en argot Liferay se trataría pues, de *portlets* diferentes (sea en una misma aplicación o en aplicaciones diferentes).
4. Comunicación entre *portlets*, la que se adapte mejor al proyecto

Requisitos funcionales

a) Generales

1. La pantalla principal de cada módulo consistirá en un listado, mediante un componente de Alloy UI llamado "*search-container*", con los datos más relevantes visibles, de la entidad que el *portlet* gestione.
2. En la parte superior de la pantalla principal tendremos un conjunto de campos llamados filtros, para poder realizar búsquedas y filtrar la información del listado con más precisión
3. En la pantalla principal de cada módulo, en un lateral tendremos los botones con los que se podrán realizar diferentes acciones sobre los elementos del listado principal. Algunas de estas acciones se implementarán integrándolas como botones del componente de Alloy UI comentado en el primer punto.
4. Al pulsar un botón de los citados anteriormente, para las acciones que impliquen visualizar o editar datos de un elemento del listado, se abrirá una pantalla con un formulario -cumplimentado o vacío, según el caso- con los campos relacionados a la entidad correspondiente. Dicho formulario se procesará y se llevaran a cabo una o varias acciones que afectarán a la base de datos.

5. Las entidades podrán tener diferentes estados.
 6. Se definirán diferentes roles para los usuarios de la aplicación, así cada rol tendrá unas características específicas y podrá llevar a cabo acciones concretas
- b) Específicos para la gestión de trabajadores
1. Asignar a cada trabajador un coste general, para poder llevar a cabo estimaciones de costes
 2. Asignar un rol, que permitirá generalizar otras características, como por ejemplo tarifas o funciones generales a desempeñar
 3. Introducir la cantidad de horas diarias que un trabajador dedica a un proyecto (imputación de horas)
 4. Se podrán asignar trabajadores a un proyecto concreto
- c) Específicos para la gestión de proyectos
1. Habrá una distinción entre proyectos y servicios (proyectos de duración no determinada)
 2. Será posible crear subproyectos vinculados a proyectos
 3. Se podrán asignar uno o varios proyectos a los trabajadores

6.2 Casos de uso

Generales

Todos los módulos nos permiten realizar las acciones siguientes:

- alta: creación de un nuevo elemento
- baja: eliminación de un elemento existente, efectivamente se marcará como elemento eliminado
- modificación: edición de las características o propiedades de un elemento existente

- listado: visualización de una lista de elementos
- búsqueda: visualización de una lista de resultados en función de unos filtros

Gestión de trabajadores

Los trabajadores, adicionalmente, deben poder ver los proyectos a los que están asignados e imputarles horas invertidas en ellos. También podrán buscar otros trabajadores para visualizar sus datos -de contacto, por ejemplo- o modificar su propios datos.

En el caso del rol administrador, este tendrá más opciones para gestionar los trabajadores: baja de trabajadores, asignación y desvinculación de proyectos, además de todas las funciones disponibles para los trabajadores.

Se pueden observar los casos de uso en el siguiente diagrama:

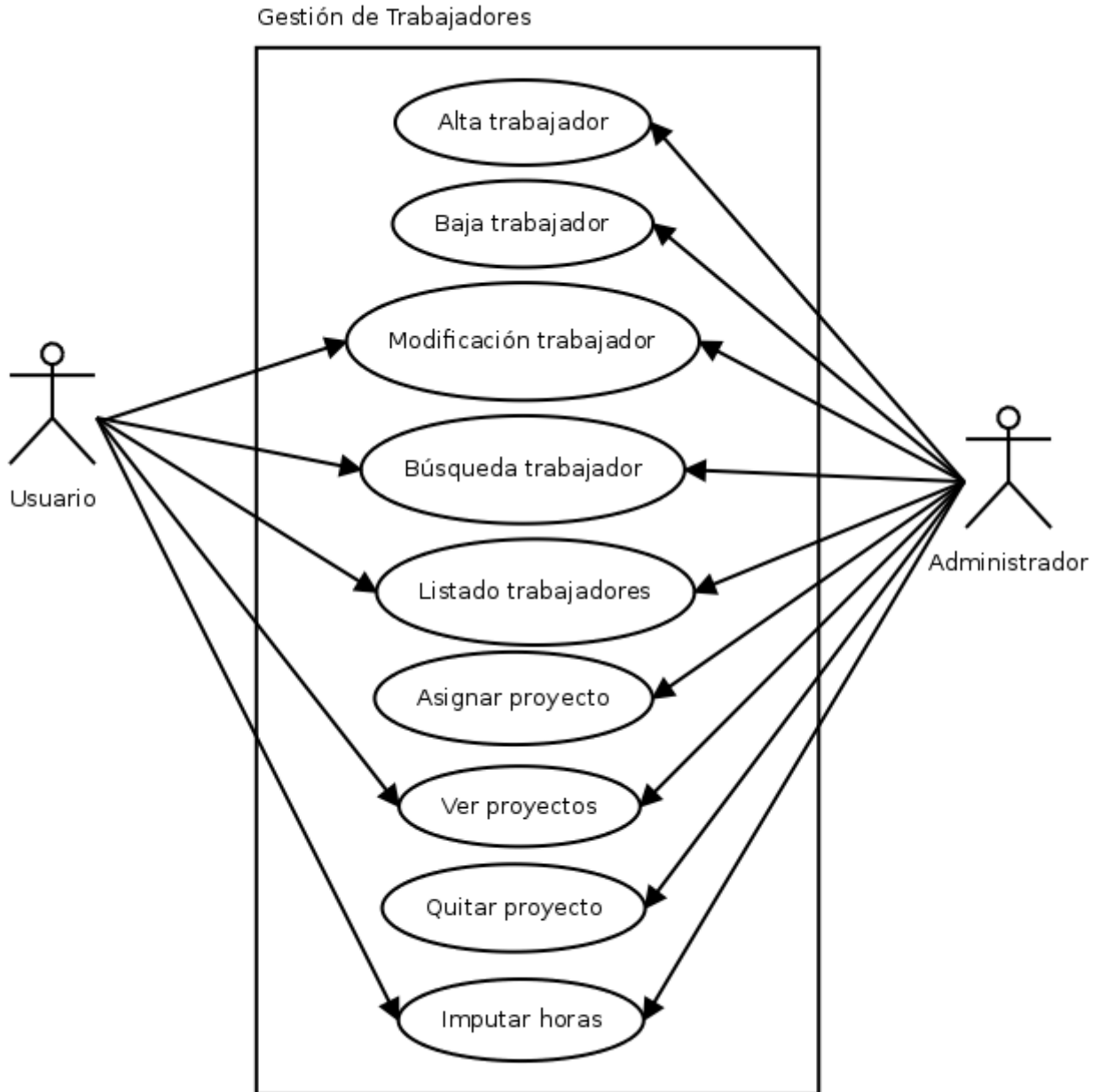


Illustration 3

Gestión de contactos (clientes y proveedores)

Para los contactos tenemos los casos de uso generales, especificados anteriormente, además de la vinculación o desvinculación con los proyectos con los que están relacionados. Estas últimas acciones, conjuntamente con las bajas, se reservan para el rol de administrador. Veamos el diagrama en la siguiente imagen.

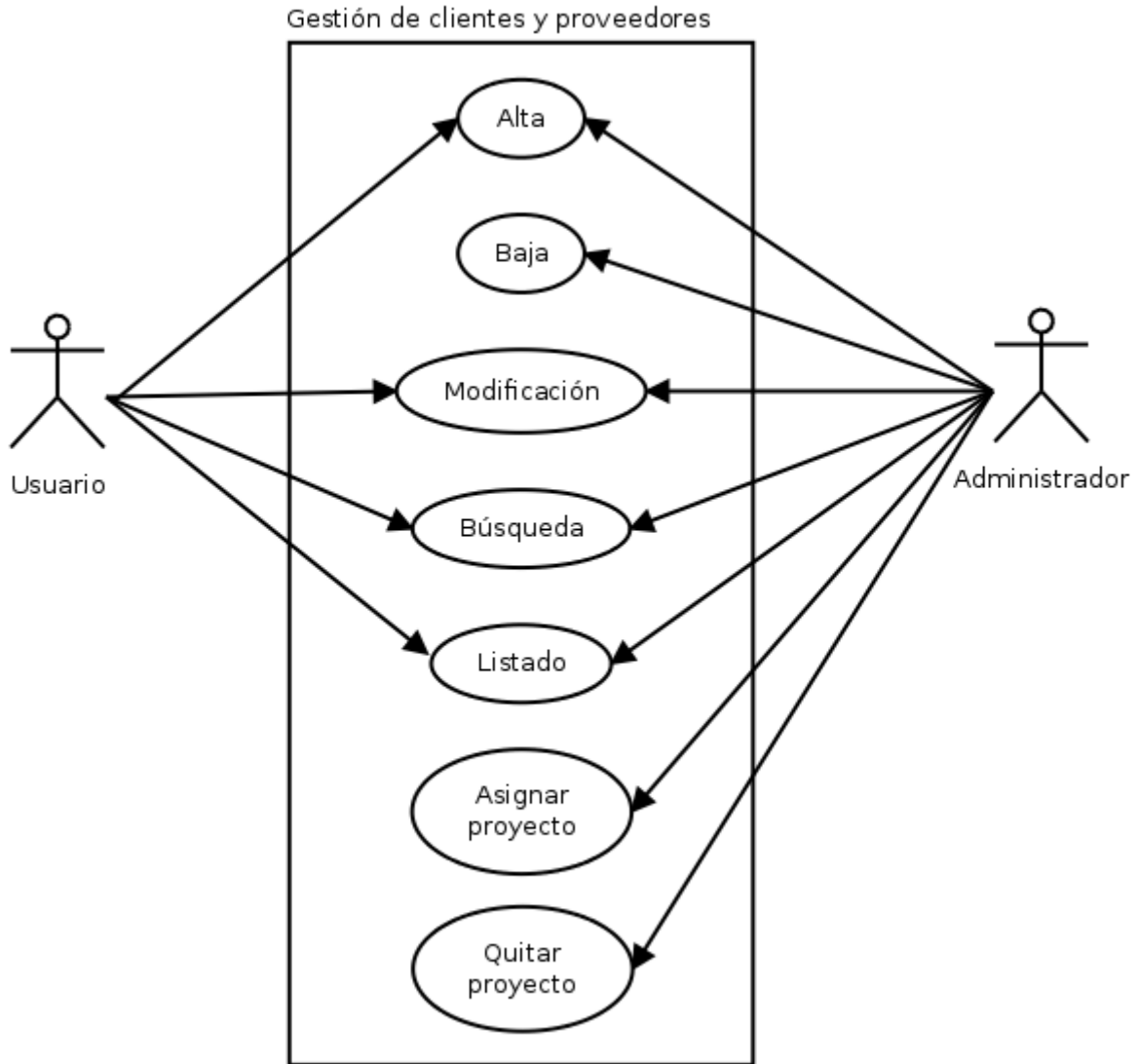


Illustration 4

Gestión de proyectos

En el caso del *portlet* de proyectos disponemos de las operaciones habituales -alta, baja, modificación, listado, búsqueda- pero adicionalmente el rol de administrador podrá asignar trabajadores y contactos a los proyectos. En el siguiente diagrama de casos de uso podemos verlo más fácilmente.

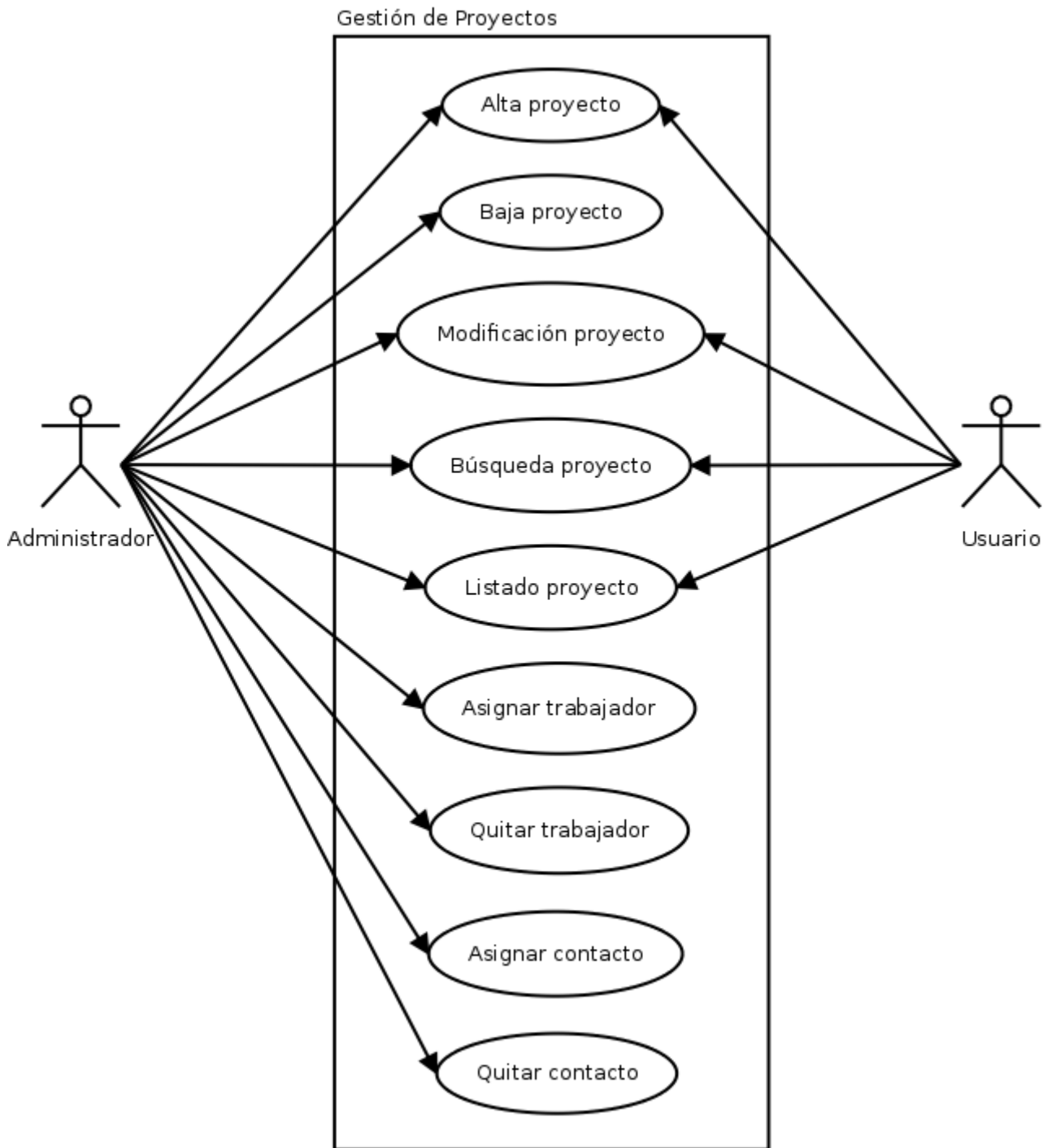


Illustration 5

7. Diseño e implementación

7.1 Arquitectura del sistema

Los principales componentes del sistema son 3: servidor, base de datos y aplicación(es).

El servidor, como ya se ha comentado anteriormente, es Liferay Portal, el cual integra Tomcat, pero esto es transparente.

En segundo lugar tenemos el sistema gestor de bases de datos, en nuestro caso PostgreSQL.

Para terminar, tenemos los *portlets* implementados para cumplir con los objetivos de este proyecto, que en un principio son 4:

- **Mpwc portlet:** integra todas las clases de acceso a la base de datos, utilizadas por el resto de *portlets*. Se ha desarrollado como por separado para hacer la aplicación modular
- **Mpwc-worker portlet:** aplicación de gestión de trabajadores
- **Mpwc-contact portlet:** aplicación de gestión de clientes y proveedores
- **Mpwc-project portlet:** aplicación de gestión de proyectos

Podemos ver un esquema de como interactúan entre sí en la imagen a continuación.

Arquitectura general del sistema

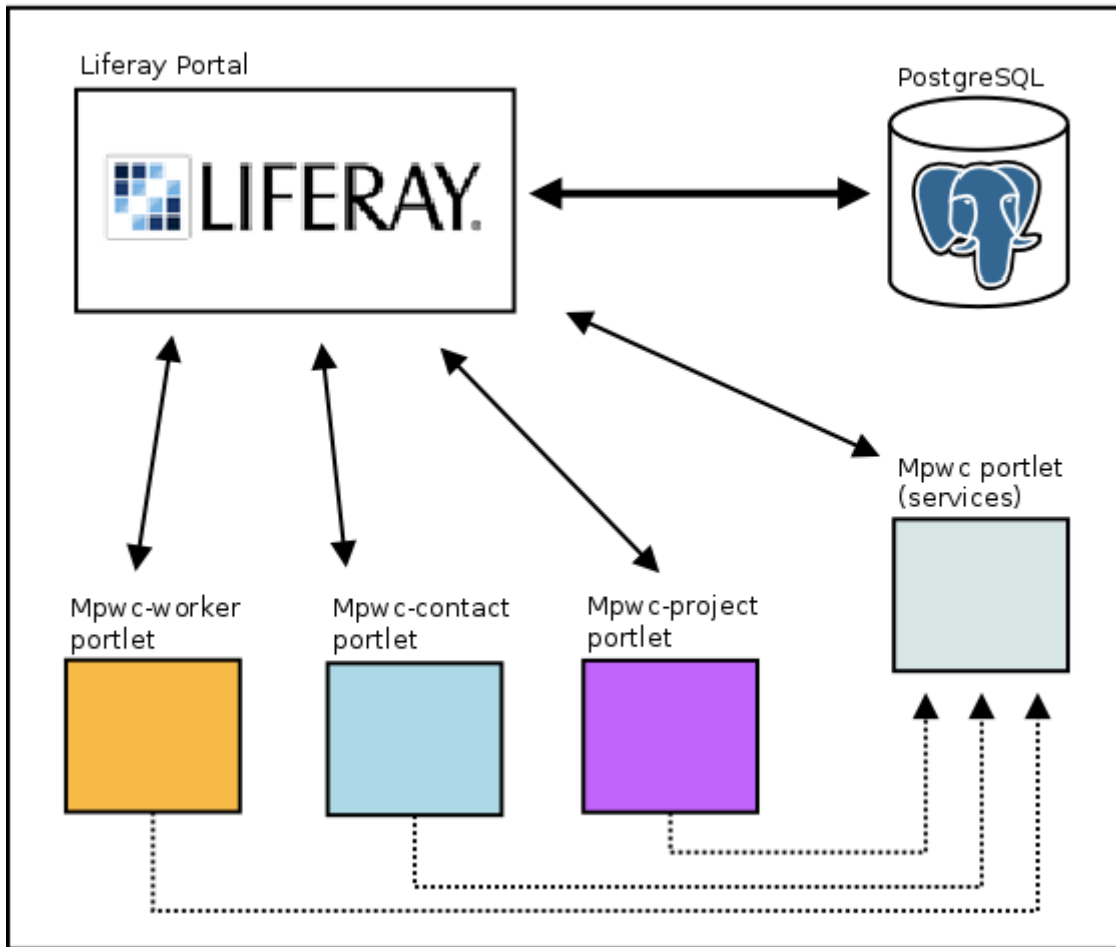


Illustration 6

7.2 interfaz de usuario

A continuación se muestran unos esbozos del aspecto general de la aplicación, desarrollados a partir de los requisitos recogidos. Cabe mencionar que al ser esbozos pueden variar al gusto del cliente.

El primero pretende mostrar la distribución general de las pantallas principales de los *portlets*:

Diagrama general de distribución de los portlets

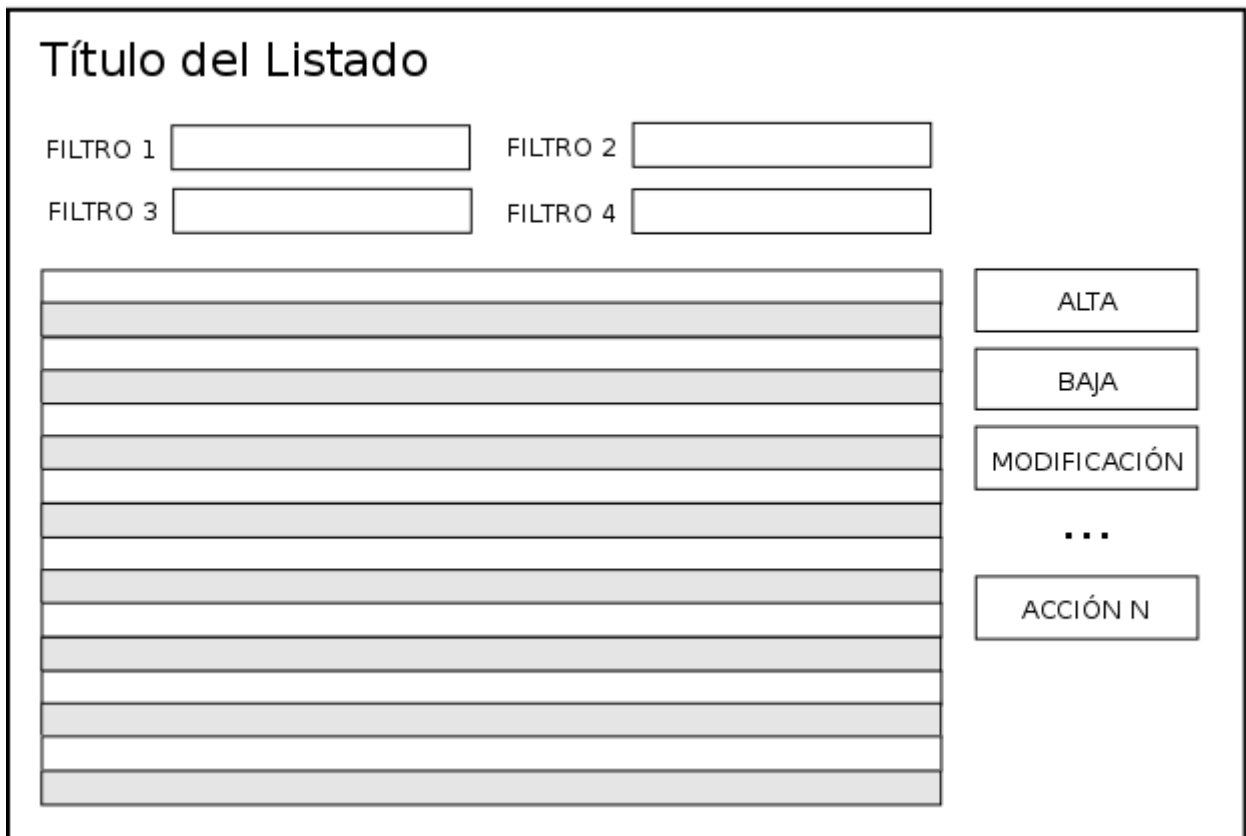
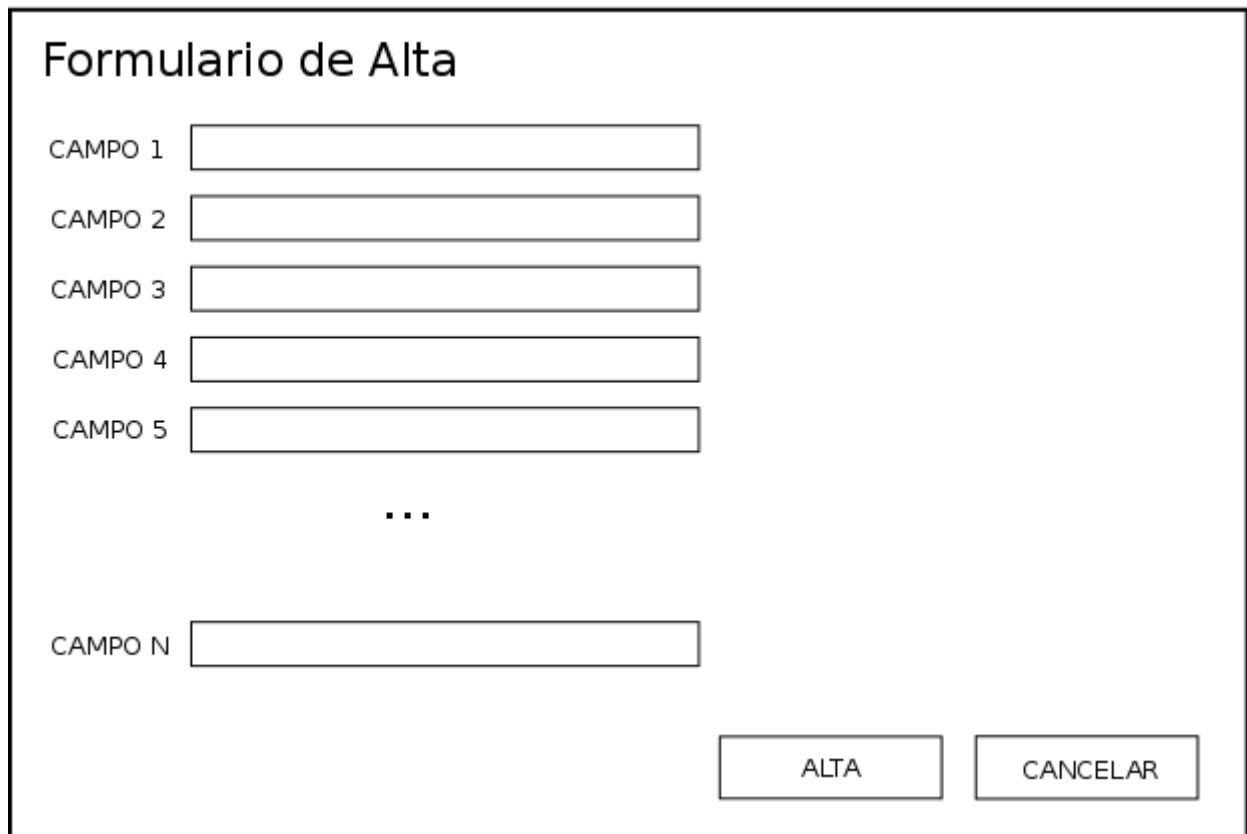


Illustration 7

En segundo lugar podemos observar un ejemplo de formulario de alta:

Diagrama de un formulario de alta genérico de los portlets



Formulario de Alta

CAMPO 1

CAMPO 2

CAMPO 3

CAMPO 4

CAMPO 5

...

CAMPO N

ALTA CANCELAR

Illustration 8

Para acabar, se muestra un mensaje de confirmación de baja con la pantalla principal de fondo:

Diagrama genérico de confirmación de baja

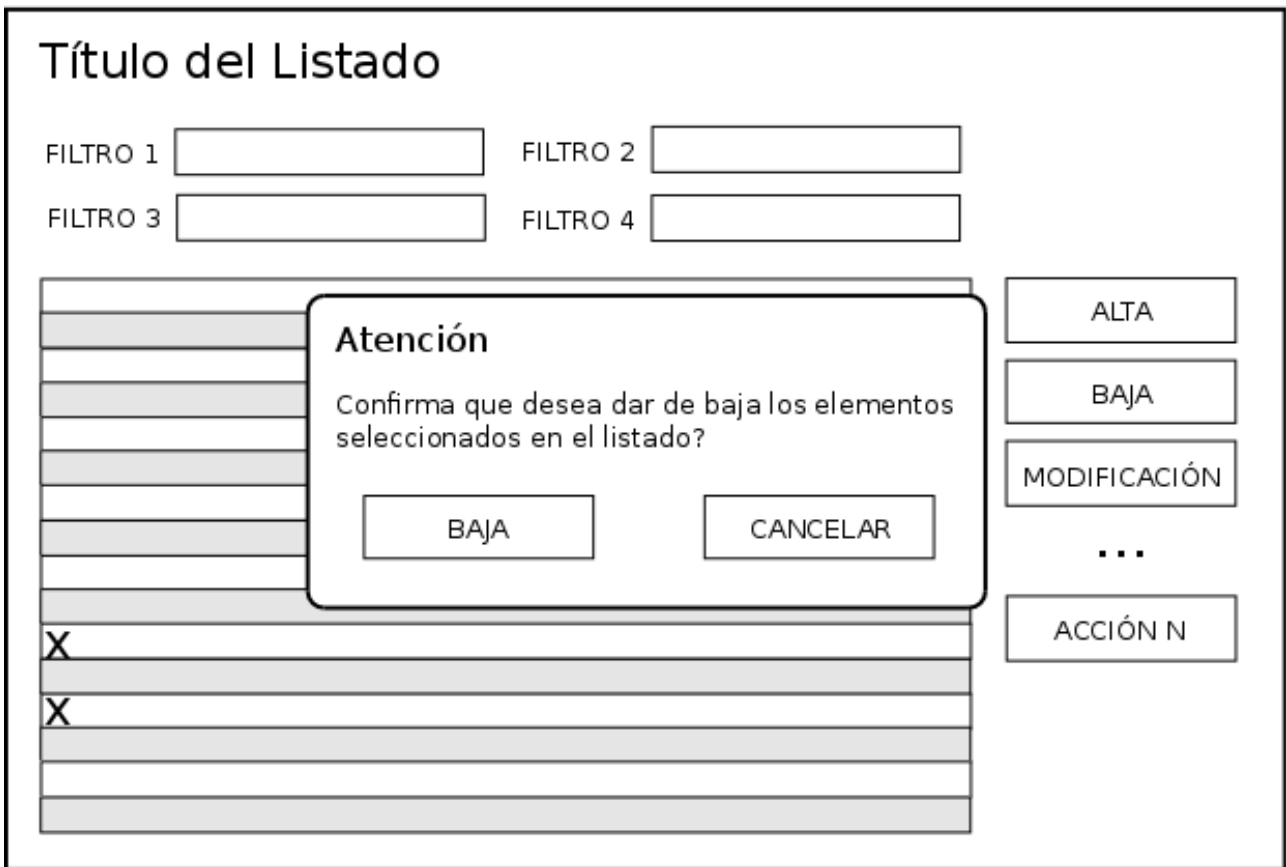


Illustration 9

7.3 Base de datos

7.3.1 Introducción a *Service Builder*

En muchos escenarios aún se continúa implementando la lógica de acceso a base de datos artesanalmente utilizando algún conector, como por ejemplo JDBC, y programado métodos que ejecutan sentencias SQL. Siguiendo esta metodología, para obtener una capa de acceso a los datos sólida es necesario que los desarrolladores inviertan mucho tiempo en la construcción de consultas SQL y el tratamiento de sus parámetros, para evitar errores en las transacciones con la base de datos o, en el peor de los casos, inyección de código por parte de usuarios maliciosos.

Para evitar estos problemas y establecer una base estable desde un principio, apareció el proyecto Hibernate, el cual ofrece una API de acceso a los datos basándose en el mapeo de objetos a tablas, es decir, se pueden enlazar objetos Java con tablas de nuestra base

de datos y de esta manera es Hibernate quien se encarga de la persistencia de los datos. Esta API también se complementa con Spring, tecnología con la que se lleva a cabo la Inyección de Dependencia (*Dependency Injection*), que nos permite inyectar objetos ya instanciados dentro de otros objetos. Un ejemplo es la conexión con la base de datos, anteriormente se tenía que pasar entre objetos de alguna manera, creándola cada vez con ciertos parámetros. Pues bien con Spring la conexión ya está incluida en los objetos mapeados por Hibernate, y por lo tanto, a nivel de programación, no hace falta preocuparse este aspecto. Éste es sólo un pequeño ejemplo de inyección de dependencia.

Actualmente Liferay dispone de una herramienta llamada “*Service Builder*”, con la que se puede delegar la creación de clases y métodos necesarios para desarrollar toda la lógica de acceso a bases de datos de forma automática. Consiste en definir un documento XML que contiene las entidades y sus propiedades, los tipos de datos y el nombre de la tabla de la base de datos. Al compilar este documento se generan todos los archivos de configuración de Hibernate, Spring, los *scripts* SQL y todas las clases java necesarias. Al desplegar el *portlet* en el portal, este se encarga de leer estos archivos de configuración y ejecutar los *scripts* generados para reflejar el contenido del documento XML definido en la base de datos.

Todo esto es posible gracias a que Liferay integra todas las tecnologías citadas anteriormente. La integración de estas tecnologías es compleja y no es mi objetivo explicar su funcionamiento interno, sino su uso en este proyecto.

7.3.2 Diagrama ER

Con esta explicación, a continuación presento las relaciones teóricas que debe tener la base de datos de la aplicación. Cabe destacar que no se muestra la entidad “Rol” porque finalmente se utiliza el sistema de permisos de Liferay Portal, solo tendré que mapearlos a los que se definan en la aplicación mediante otro archivo XML de configuración, y añadir cierto código en los métodos de inserción y borrado de datos, así como a la hora de mostrar las acciones que cada usuario puede realizar.

A continuación se muestra el **diagrama ER** teórico con las entidades principales y las relaciones entre ellas.

Modelo Entidad-Relación

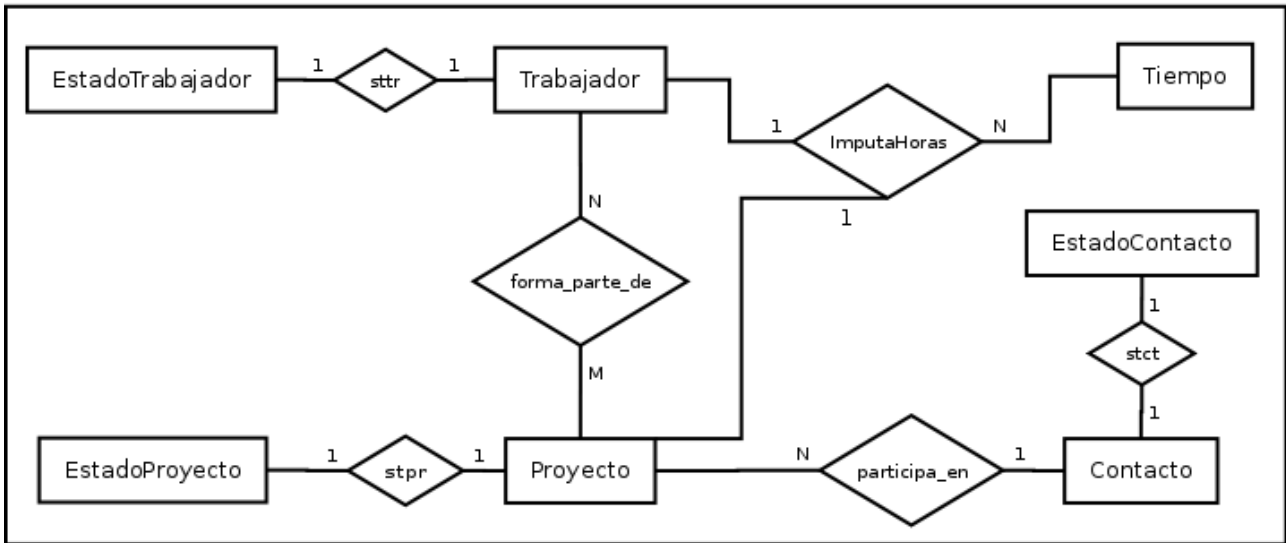


Illustration 10

7.3.3 Diseño de las tablas

A continuación se muestran las tablas que se crearán, con una pequeña descripción de qué es cada campo. Los nombres están en inglés, pues para el desarrollo de un proyecto de software libre, teniendo en cuenta que se le quiere dar continuidad, se ha establecido este criterio para facilitar el apoyo de la comunidad.

tbl_mpw_c_workers <i>Trabajadores</i>	tbl_mpw_c_projects <i>Proyectos</i>	tbl_mpw_c_contacts <i>Contactos</i>	tbl_mpw_c_workerstatus <i>EstadoTrabajador</i>	tbl_mpw_c_timebox <i>Tiempos</i>
<ul style="list-style-type: none"> *workerid bigint <i>Clave primaria</i> °groupid bigint <i>Identificador de grupo de Liferay</i> °companyid <i>Identificador de Compañía de Liferay</i> °username character varying <i>Nombre de usuario de Liferay</i> °createdate timestamp <i>Fecha de creación de Liferay</i> °modifieddate timestamp <i>Fecha de última modificación de Liferay</i> °nif character varying <i>NIF del trabajador</i> °name character varying <i>Nombre</i> °surname character varying <i>Apellidos</i> °email character varying <i>Dirección de correo electrónico</i> °phone character varying <i>Teléfono</i> °comments character varying <i>Comentarios</i> °statusid bigint <i>Clave foránea a EstadoTrabajador</i> 	<ul style="list-style-type: none"> *projectid bigint <i>Clave primaria</i> °groupid bigint <i>Identificador de grupo de Liferay</i> °companyid <i>Identificador de Compañía de Liferay</i> °createdate timestamp <i>Fecha de creación de Liferay</i> °modifieddate timestamp <i>Fecha de última modificación de Liferay</i> °name character varying <i>Nombre</i> °type character varying <i>Tipo de proyecto (proyecto o servicio)</i> °descshort character varying <i>Descripción corta</i> °descfull character varying <i>Descripción completa</i> °costestimate euros double precision <i>Coste estimado en euros</i> °timeestimatedhours bigint <i>Timepo estimado en horas</i> °cansetworkerhours boolean <i>Flag que indica si los trabajadores pueden imputar tiempo</i> °startdate timestamp <i>Fecha de inicio</i> °enddate timestamp <i>Fecha de fin</i> °comments character varying <i>Comentarios</i> °projectstatusid bigint <i>Clave foránea a EstadoProyecto</i> °contactid bigint <i>Clave foránea a Contactos</i> 	<ul style="list-style-type: none"> *contactid bigint <i>Clave primaria</i> °groupid bigint <i>Identificador de grupo de Liferay</i> °companyid <i>Identificador de Compañía de Liferay</i> °username character varying <i>Nombre de usuario de Liferay</i> °createdate timestamp <i>Fecha de creación de Liferay</i> °modifieddate timestamp <i>Fecha de última modificación de Liferay</i> °nif character varying <i>NIF del contacto</i> °firmname character varying <i>Nombre fiscal</i> °address character varying <i>Dirección</i> °zipcode character varying <i>Código postal</i> °city character varying <i>Ciudad</i> °country character varying <i>País</i> °email character varying <i>Dirección de correo electrónico</i> °phone character varying <i>Teléfono</i> °phone2 character varying <i>Teléfono alternativo</i> °comments character varying <i>Comentarios</i> °image bytea <i>Imagen</i> °type character varying <i>Tipo de contacto (cliente o proveedor)</i> °contactstatusid bigint <i>Clave foránea a EstadoContacto</i> 	<ul style="list-style-type: none"> *statusid bigint <i>Clave primaria</i> °desc_en_us character varying <i>Descripción en inglés</i> °desc_es_es character varying <i>Descripción en castellano</i> °desc_es_ca character varying <i>Descripción en catalán</i> 	<ul style="list-style-type: none"> *timeboxid bigint <i>Clave primaria</i> °groupid bigint <i>Identificador de grupo de Liferay</i> °companyid <i>Identificador de Compañía de Liferay</i> °createdate timestamp <i>Fecha de creación de Liferay</i> °modifieddate timestamp <i>Fecha de última modificación de Liferay</i> °dedicationdate timestamp <i>Fecha en que se ha trabajado en el proyecto</i> °minutes bigint <i>Minutos dedicados, tiempo a imputar</i> °comments character varying <i>Comentarios</i> °projectid bigint <i>Clave foránea a Proyectos</i> °workerid bigint <i>Clave foránea a Trabajadores</i>
			<ul style="list-style-type: none"> *projectstatusid bigint <i>Clave primaria</i> °desc_en_us character varying <i>Descripción en inglés</i> °desc_es_es character varying <i>Descripción en castellano</i> °desc_es_ca character varying <i>Descripción en catalán</i> 	
			<ul style="list-style-type: none"> *contactstatusid bigint <i>Clave primaria</i> °desc_en_us character varying <i>Descripción en inglés</i> °desc_es_es character varying <i>Descripción en castellano</i> °desc_es_ca character varying <i>Descripción en catalán</i> 	

Una vez definidas las características que necesitamos para cada entidad y cómo deben estar relacionadas entre ellas, ya podemos crear el archivo de configuración para la herramienta Service Builder, el cual creará todos los archivos necesarios para crear las tablas en la base de datos.

7.3.4 Archivo de configuración de servicios: service.xml

Se puede consultar la última versión del archivo en el repositorio¹⁵ de código del proyecto.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE service-builder PUBLIC "-//Liferay//DTD Service Builder 6.1.0//EN" "http://www.liferay.com/dtd/liferay-
service-builder_6_1_0.dtd">
<service-builder package-path="com.mpsc">
  <author>rsicart</author>
  <namespace>tools</namespace>

  <!-- Begin MPWC Worker -->

  <entity name="Worker" table="tbl_mpsc_workers" local-service="true" remote-service="true">

    <!-- PK fields -->

    <column name="workerId" type="long" primary="true" id-type="sequence" id-
param="seq_worker_workerid"/>

    <!-- Group instance -->

    <column name="groupId" type="long" />

    <!-- Audit fields -->

    <column name="companyId" type="long" />
    <column name="userId" type="long" />
    <column name="userName" type="String" />
    <column name="createDate" type="Date" />
    <column name="modifiedDate" type="Date" />

    <!-- Other fields -->

    <column name="nif" type="String" />
    <column name="name" type="String" />
    <column name="surname" type="String" />
    <column name="email" type="String" />
    <column name="phone" type="String" />
    <column name="comments" type="String" />

    <!-- FK fields -->
    <column name="statusId" type="long" />

    <!-- Relation m:n -->
    <column
name="projectList"
type="Collection"
entity="Project"
mapping-table="tbl_mpsc_worker_project"
/>

    <!-- Relation 1:n -->
    <column name="timeBoxList" type="Collection" entity="TimeBox" mapping-key="workerId" />

    <!-- Order -->
    <order by="asc">
      <order-column name="workerId" />
    </order>
  </entity>
</service-builder>
```

15 06/12/2012 Archivo service.xml. <<https://github.com/rsicart/mpwc/blob/master/docroot/WEB-INF/service.xml>>


```

<!-- Finder methods -->
<finder name="Nif" return-type="Collection">
  <finder-column name="nif" />
</finder>

<finder name="Email" return-type="Collection">
  <finder-column name="email" />
</finder>

<finder name="Name" return-type="Collection">
  <finder-column name="name" />
</finder>

<finder name="Surname" return-type="Collection">
  <finder-column name="surname" />
</finder>

<finder name="G_Nif" return-type="Collection">
  <finder-column name="groupId" />
  <finder-column name="nif" />
</finder>

<finder name="G_Email" return-type="Collection">
  <finder-column name="groupId" />
  <finder-column name="email" />
</finder>

<finder name="G_Name" return-type="Collection">
  <finder-column name="groupId" />
  <finder-column name="name" />
</finder>

<finder name="G_Surname" return-type="Collection">
  <finder-column name="groupId" />
  <finder-column name="surname" />
</finder>

<finder name="G_U" return-type="Collection">
  <finder-column name="groupId"></finder-column>
  <finder-column name="userId"></finder-column>
</finder>
</entity>

<entity name="Status" local-service="true" table="tbl_mpw_c_workerstatus">
  <column name="statusId" type="long" primary="true"></column>
  <column name="desc_en_US" type="String"></column>
  <column name="desc_es_ES" type="String"></column>
  <column name="desc_ca_ES" type="String"></column>
  <column
    name="workerStatus"
    type="Collection"
    entity="Worker"
    mapping-key="statusId" />
</entity>

<!-- End MPWC Worker -->

<!-- Begin MPWC Project -->

  <entity name="Project" table="tbl_mpw_c_projects" local-service="true" remote-service="true">

    <!-- PK fields -->

    <column name="projectId" type="long" primary="true" id-type="sequence" id-
param="seq_project_projectid"/>

    <!-- Group instance -->

    <column name="groupId" type="long" />

    <!-- Audit fields -->

    <column name="companyId" type="long" />
    <column name="createDate" type="Date" />
    <column name="modifiedDate" type="Date" />

    <!-- Other fields -->

    <column name="name" type="String" />
    <column name="type" type="String" />

```

```

<column name="descShort" type="String" />
<column name="descFull" type="String" />
<column name="costEstimatedEuros" type="double" />
<column name="timeEstimatedHours" type="long" />
<column name="canSetWorkerHours" type="boolean" />
<column name="startDate" type="Date" />
<column name="endDate" type="Date" />
<column name="comments" type="String" />

<!-- FK fields -->
<column name="projectStatusId" type="long" />
<column name="workerId" type="long" /> <!-- project manager -->
<column name="contactId" type="long" /> <!-- contact (customer) -->

<!-- Relation m:n -->
<column name="workerList" type="Collection" entity="Worker" mapping-table="tbl_mpw_worker_project" />

<!-- Relation 1:n -->
<column name="timeBoxList" type="Collection" entity="TimeBox" mapping-key="projectId" />

<!-- Order -->

<order by="asc">
  <order-column name="projectId" />
</order>

<!-- Finder Methods -->

  <finder name="G_Name" return-type="Collection">
    <finder-column name="groupId"></finder-column>
    <finder-column name="name"></finder-column>
  </finder>

  <finder name="G_Type" return-type="Collection">
    <finder-column name="groupId"></finder-column>
    <finder-column name="type"></finder-column>
  </finder>

  <finder name="GroupId" return-type="Collection">
    <finder-column name="groupId"></finder-column>
  </finder>

  <finder name="CompanyId" return-type="Collection">
    <finder-column name="companyId"></finder-column>
  </finder>

</entity>

<entity name="ProjectStatus" local-service="true" table="tbl_mpw_projectstatus">
  <column name="projectStatusId" type="long" primary="true"></column>
  <column name="desc_en_US" type="String"></column>
  <column name="desc_es_ES" type="String"></column>
  <column name="desc_ca_ES" type="String"></column>
  <column
    name="projectStatus"
    type="Collection"
    entity="Project"
    mapping-key="projectStatusId" />
</entity>

<entity name="TimeBox" local-service="true" table="tbl_mpw_timebox">
  <column name="timeboxId" type="long" primary="true" id-type="sequence" id-
param="seq_timebox_timeboxid"></column>
  <column name="projectId" type="long"></column>
  <column name="workerId" type="long"></column>

  <!-- Group instance -->
  <column name="groupId" type="long" />

  <!-- Audit fields -->
  <column name="companyId" type="long" />
  <column name="createDate" type="Date" />
  <column name="modifiedDate" type="Date" />

  <column name="dedicationDate" type="Date" />
  <column name="minutes" type="long" />
  <column name="comments" type="String" />

  <!-- Finders -->
  <finder name="P" return-type="Collection">
    <finder-column name="projectId"></finder-column>
  </finder>

```

```

        <finder name="W" return-type="Collection">
        <finder-column name="workerId"></finder-column>
</finder>

        <finder name="P_W" return-type="Collection">
        <finder-column name="projectId"></finder-column>
        <finder-column name="workerId"></finder-column>
</finder>

<finder name="P_W_DD" return-type="Collection">
        <finder-column name="projectId"></finder-column>
        <finder-column name="workerId"></finder-column>
        <finder-column name="dedicationDate"></finder-column>
</finder>

        <finder name="G_P" return-type="Collection">
        <finder-column name="groupId"></finder-column>
        <finder-column name="projectId"></finder-column>
</finder>

        <finder name="G_W" return-type="Collection">
        <finder-column name="groupId"></finder-column>
        <finder-column name="workerId"></finder-column>
</finder>
</entity>

<!-- End MPWC Project -->

<!-- Begin MPWC Contact -->

<entity name="Contact" table="tbl_mpwc_contacts" local-service="true" remote-service="true">

    <!-- PK fields -->

    <column name="contactId" type="long" primary="true" id-type="sequence" id-
param="seq_contact_contactid"/>

    <!-- Group instance -->

    <column name="groupId" type="long" />

    <!-- Audit fields -->

    <column name="companyId" type="long" />
    <column name="userId" type="long" />
    <column name="userName" type="String" />
    <column name="createDate" type="Date" />
    <column name="modifiedDate" type="Date" />

    <!-- Other fields -->

    <column name="ctype" type="String" />
    <column name="nif" type="String" />
    <column name="firmname" type="String" />
    <column name="address" type="String" />
    <column name="zipcode" type="String" />
    <column name="city" type="String" />
    <column name="country" type="String" />
    <column name="email" type="String" />
    <column name="phone" type="String" />
    <column name="phone2" type="String" />
    <column name="comments" type="String" />

    <!-- FK fields -->
    <column name="contactStatusId" type="long" />

<!-- Relation 1:n -->
<column name="projectList" type="Collection" entity="Project" mapping-key="contactId" />

    <!-- Order -->
    <order by="asc">
        <order-column name="contactId" />
    </order>

    <!-- Finder methods -->

    <finder name="Ctype" return-type="Collection">
        <finder-column name="ctype" />
    </finder>

    <finder name="Nif" return-type="Collection">

```

```

        <finder-column name="nif" />
    </finder>

    <finder name="Email" return-type="Collection">
        <finder-column name="email" />
    </finder>

    <finder name="Firmname" return-type="Collection">
        <finder-column name="firmname" />
    </finder>

    <finder name="Zipcode" return-type="Collection">
        <finder-column name="zipcode" />
    </finder>

    <finder name="C_C" return-type="Collection">
        <finder-column name="city" />
        <finder-column name="country" />
    </finder>

</entity>

<entity name="ContactStatus" local-service="true" table="tbl_mpwc_contactstatus">
    <column name="contactStatusId" type="long" primary="true"></column>
    <column name="desc_en_US" type="String"></column>
    <column name="desc_es_ES" type="String"></column>
    <column name="desc_ca_ES" type="String"></column>
    <column
        name="contactStatus"
        type="Collection"
        entity="Contact"
        mapping-key="contactStatusId" />

</entity>

<!-- End MPWC Contact -->
</service-builder>

```

7.4 Diagrama de clases

Para concretar más la información con la que se trabajará con cada entidad, así como sus relaciones entre ellas, se muestra un diagrama UML de clases a continuación.

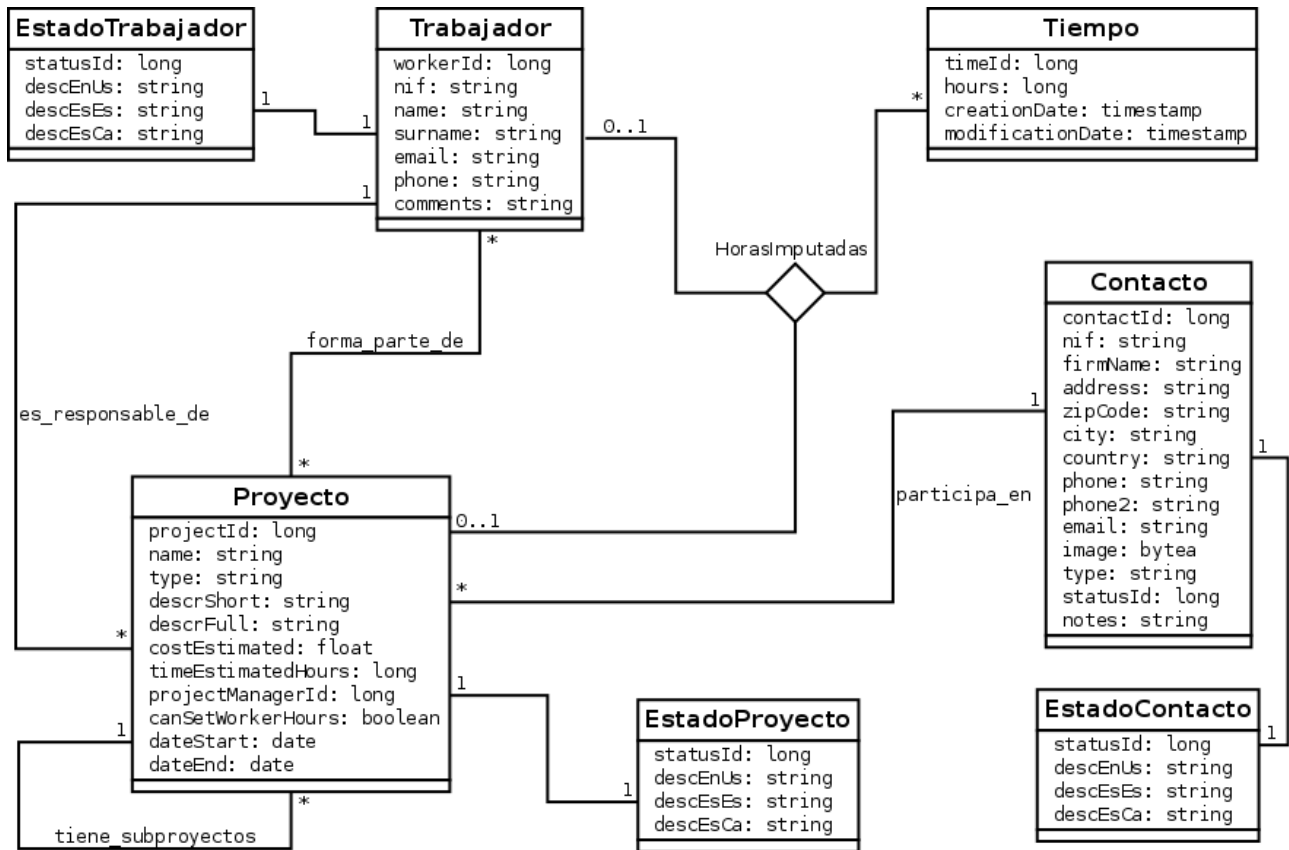


Illustration 12

Este diagrama de clases también es teórico y tiene la finalidad de ayudar a la comprensión de las clases utilizadas en el proyecto, así como las relaciones entre ellas. De esta manera se puede entender mejor el contenido del archivo *service.xml*. Realmente se generan automáticamente muchas más clases, entre ellas las principales que aparecen en el diagrama, que nos proporcionan un diseño por capas, separando lo que son interfaz de usuario, modelo y persistencia de datos.

7.5 Decisiones y cambios en la implementación

A continuación se presentan una serie de decisiones y cambios que se han producido durante el desarrollo de este proyecto, sea por decisión del cliente, en este caso Sonicon Systems, o como soluciones alternativas a problemas encontrados.

- a) Se ha decidido, por parte del cliente, **utilizar al máximo las herramientas que proporciona Liferay** siempre que sea posible. Esto ha implicado cambiar los listados de jqGrid y la manera de funcionar de los *portlets* programados hasta el momento. Los portlets tienen un ciclo de vida definido, que va acorde al patrón de

diseño MVC, por el cual su naturaleza es llamar a alguna de las acciones que nos proporciona el controlador (en el caso de un portlet de este proyecto, es la clase principal del portlet que extiende la clase MVCPortlet), conocida como fase de acción o *action phase*, y luego se genera el resultado en la fase de renderizado (*render phase*) conjuntamente con la vista y el modelo. JqGrid, muestra los datos de una base de datos mediante AJAX, de esta manera se pueden recargar dinámicamente los datos. Para poder integrar esta tecnología en portlets, hay que programar todas las acciones que proporcionan estos datos dentro de una acción llamada *serveResource* de la clase principal de cada portlet. Haciendo esto se consigue acceder a los datos de la base de datos sin tener que pasar por todo el ciclo de vida del portlet, y por consiguiente, sin tener que recargar la página (pues al renderizarla se crea de nuevo). Este comportamiento, como podéis imaginar, es ideal para AJAX.

Al decidir usar las herramientas de Liferay, y concretamente Alloy UI, que es muy potente, se han cambiado todas las tablas jqGrid por elementos *search-container* de Alloy UI, los cuales vienen perfectamente integrados con el portal de Liferay y los objetos creados por *service-builder*, además de ser visualmente atractivos y fáciles de implementar. El componente *search-container* obtiene los datos siguiendo el ciclo de vida natural del portlet, no usa AJAX. Esta decisión ha implicado que las acciones definidas dentro de *serveResource* pasen a ser acciones del controlador.

- b) Tratándose de un proyecto de software libre, personalmente he tomado la decisión de **internacionalizar el proyecto** de tres maneras:
- la totalidad de los comentarios en los repositorios de código, incluyendo los mensajes en los *commits*, en inglés
 - soportar tres idiomas desde un inicio: catalán, castellano e inglés. En función del idioma configurado en el sistema del usuario se muestran las etiquetas y títulos
 - los comentarios del código en inglés
- c) **Sistema de permisos basado en roles** de Liferay: en lugar de implementar un sistema de permisos propio basado en roles, y conociendo de la existencia del

sistema de permisos que Liferay proporciona, se ha decidido integrar los portlets con dicho sistema. De esta manera se pueden configurar los permisos de cada rol, de manera gráfica mediante el panel de control, para cada módulo del proyecto. Se han definido, para empezar, dos roles de portal para todos los módulos:

- **MpwcManager**: rol para los administradores de los módulos. Este rol tiene permisos para llevar a cabo todas las acciones posibles en todos los portlets del proyecto
- **MpwcUser**: rol para los trabajadores, con el cual pueden realizar tareas de consulta y en algunos casos de modificación

De esta manera, cuando un usuario se registra al portal, adquiere el rol de portal *User*, por defecto, y cuando se da de alta en la aplicación Mpwc se le asigna automáticamente el rol *MpwcUser*. Por el momento con estos dos roles ya es suficiente, pero es posible que en un futuro se deban segmentar más.

- d) Encontrado **bug LPS-29668¹⁶ en service-builder** para Liferay Portal 6.1: para los métodos que no devuelven nada (*void*) esta herramienta genera código erróneo para una entidad en concreto, pues en una de las clases auto-generadas encargadas de llamar a dicha entidad se fuerza a devolver un objeto o lanzar una excepción de tipo *UnsupportedOperationException*. Esta clase, al no devolver nada los métodos, lanzaba la excepción, cancelando de esta manera la transacción con la base de datos.

Después de investigar varios días y dar con este problema, en primer lugar opté por crear lo que se conoce en el mundo de Liferay como *Custom-sql¹⁷*, que consiste en crear una nueva entidad en la capa más baja de los servicios de acceso a la base de datos, con la que es posible configurar un archivo XML de consultas SQL utilizando *prepared statements*.

Finalmente, como solución alternativa se ha decidido cambiar los métodos que no devuelven nada por métodos que devuelven un objeto, por norma general la clave primaria de dicha entidad. Cabe destacar que dada la potencia de *custom-sql*, se

16 28/11/2012 LPS-29668 <<http://issues.liferay.com/browse/LPS-29668>>

17 06/12/2012 Custom-sql <<http://www.liferay.com/community/wiki/-/wiki/Main/Custom+queries+in+Liferay>>

sigue utilizando para otras situaciones.

7.6 Estado de la implementación

A modo de resumen o fotografía de estado, incluyo un cuadro para tener claros los puntos acabados y los que se encuentran por hacer. El *portlet* de servicios (mpwc-portlet) no se incluye en el resumen, pues no tiene definidas funcionalidades concretas ya que éstas vienen establecidas por el resto de módulos.

	Mpwc-worker	Mpwc-project	Mpwc-contact
Añadir trab. / proy. / contacto			
Editar trab. / proy. / contacto			
Eliminar trab. / proy. / contacto			
Listar/Filtrar trab. / proy. / cont.			
Ver proy. / trab. / proyectos			
Asignar proy. / trab. / proy.			
Quitar proy. / trab. / proy.			

Como se puede observar, todas las funcionalidades básicas están finalizadas.

8. Conclusiones y trabajo futuro

Liferay es una tecnología actual y muy potente. El desarrollo de software con Java aplicado al ámbito web, juntamente con esta tecnología, permite a la gran comunidad de desarrolladores Java poder empezar a desarrollar portlets rápidamente. No es fácil, personalmente me estoy encontrando con muchos problemas cotidianos, en los que pierdo muchísimo tiempo y me hacen sentir como un principiante. Al comienzo me sentía muy perdido y con demasiada información por procesar, la verdad es que ha habido momentos abrumadores. Pero poco a poco me estoy dando cuenta de que hasta el momento lo que veo es la punta del iceberg. Me queda mucho camino por hacer, muchas horas que invertir y trabajar duro, soy consciente de ello, pero estoy contento de haber metido las manos en el barro y haberme ensuciado de verdad. Hace un par de meses para mí Liferay era otra palabra tecnológica de esas que suenan tan bien y dejan la boca llena al decirlas. Hoy puedo decir que empiezo a conocerlo, y espero continuar aprendiendo de todos los errores y problemas con los que me estoy encontrando para poder desarrollar portlets potentes y de calidad.

Como objetivos cumplidos, de los marcados en el apartado segundo de este documento, puedo comentar que se han cumplido el 100% de los objetivos básicos, que son principalmente la creación de los portlets de gestión de trabajadores, proyectos y contactos, el tema para suplir la parte estética, así como todos los documentos de soporte asociados: manuales de usuario y configuración. Los objetivos personales, relacionados con el estudio de los dos libros, incluidos en la bibliografía, se han cumplido en la medida de las funcionalidades programadas por el proyecto. Esto significa que no ha habido día que no se abrieran para consultar aspectos o dudas, o también, para avanzar en la lectura de sus capítulos, no llegando a los capítulos de contenidos más avanzados o técnicos, es decir, los del final de cada libro, pues por el momento sus datos no han sido relevantes para las funcionalidades requeridas. De toda manera, esto no quiere decir que a día de hoy pare de utilizarlos, pues continuo con su lectura y estoy seguro que para seguir desarrollando otros proyectos del mismo ámbito son los libros a tener como referencia.

Como objetivos no conseguidos, pienso que hubiera sido interesante alargar el proyecto para empezar a programar lo que son las funcionalidades avanzadas de los tres portlets desarrollados, por ejemplo, para extraer datos estadísticos o informes. Como he podido

comprobar, no he tenido tiempo de desarrollar ninguno de los tres portlets adicionales, comentados en el apartado de objetivos, que como veremos se incluyen en los siguientes párrafos como trabajo por hacer.

Trabajo futuro falta mucho. Personalmente, pienso que hay que realizar múltiples iteraciones en el ciclo de vida del proyecto para poder mejorar la base de lo que será la plataforma de gestión de procesos internos de la empresa. Pienso que la mejor manera de mejorar es, como se hace en el paradigma de desarrollo de software libre, iterando infinitas veces para refinar el proyecto en sí. Por ello en la primera oración del párrafo me refiero a mejorar la base, y podemos empezar por estos dos aspectos:

- corrigiendo errores que puedan aparecer después de la finalización de este proyecto, pues como pasa en todos los proyectos, ningún código está libre de errores
- añadir funcionalidades nuevas y mejorar las existentes

Una vez hechas unas cuantas iteraciones, se puede continuar con esta metodología y empezar a desarrollar los módulos de gestión financiera, gestión de facturación y gestión de presupuestos. Los módulos actuales -trabajadores, proyectos, contactos- son útiles para tareas de gestión básica, pero pienso que es muy interesante desarrollar estas nuevas partes e integrarlas siguiendo las bases marcadas en este proyecto, pues en ellos residen procesos internos y de análisis de datos que pueden ayudar mucho a la gestión de un negocio, además de facilitar algunas tareas periódicas que pueden resultar un tanto pesadas, como es por ejemplo la introducción y contabilización de facturas.

Para acabar, quiero aportar lo que ha significado para mí la realización de este proyecto en concreto, y la realización del máster en general. Por si no lo sabéis, soy Ingeniero Técnico en Informática de Sistemas, y siempre me ha interesado la gestión de sistemas operativos y redes, además de la programación, que siempre ha estado en un segundo plano. Mi carrera profesional se ha basado en el mantenimiento de redes y la gestión de equipos y servidores. En toda empresa en la que he trabajado he intentado aportar mi granito de arena haciendo uso de software libre.

Si observáis mi trayectoria curricular, no he realizado demasiadas asignaturas de programación o desarrollo de software. De hecho, muchas de las asignaturas cursadas son de administración sistemas operativos y redes, campos en los que como he

comentado, tengo alguna experiencia. A lo largo de todo el máster, he estudiado los paradigmas de desarrollo de proyectos de software libre, pero siempre desde un punto de vista teórico. Entendiendo como trabaja la comunidad de desarrolladores, poco a poco he empezado a tener ganas de poner en práctica los conocimientos adquiridos, y como consecuencia di un giro a mi carrera profesional, hace ya casi un año, para empezar a programar en Sonicon Systems. En esta empresa he podido poner en práctica toda la teoría aprendida en el máster, y ver, desarrollando y participando tanto en estas prácticas como en otros proyectos, todo el proceso: desde la planificación hasta la entrega del producto.

Para concluir, puedo afirmar que gracias a este máster ahora formo parte de la comunidad de software libre y estoy creciendo como programador, pues con las prácticas he empezado mi primer proyecto de software libre y mis planes de futuro pasan por seguir colaborando con la comunidad.

9. Bibliografía

- [1] SARIN, Ashish. *Portlets in Action*. Primera edición. United States of America: Manning Publications Co., 2012. 612 p. ISBN: 9781935182542
- [2] SEZOV JR., Richard. *Liferay in Action*. Primera edición. United States of America: Manning Publications Co., 2012. 351 p. ISBN: 9781935182825
- [3] AYCART Pérez, David. Gibert Ginestà, Marc. Hernández Matías, Martín. Mas Hernández, Jordi. *Enginyeria del programari en entorns de programari lliure*. Primera edición. España: Fundació per a la Universitat Oberta de Catalunya, 2006. 311 p. ISBN: 8497884116
- [4] HUNT, Andrew. Thomas, David. *The Pragmatic Programmer: From Journeyman to Master*. Primera edición. Addison-Wesley, 1999. ISBN: 020161622X. ISBN-13: 9780201616224
- [5] ECKEL, Bruce. *Piensa en Java*. Segunda edición. Prentice Hall, 2002. 958 p. ISBN: 8420531928.

10. Agradecimientos

Os voy a confesar que había olvidado este capítulo. A lo largo de mi vida laboral me he encontrado con una amplia variedad de profesionales, de los cuales no todos han sabido expresar sus ideas u opiniones, o dar una palmadita en la espalda en el momento adecuado. Yo no quiero cometer ese error, he aprendido que para avanzar hay que dialogar, y que es muy importante reconocer tanto el trabajo propio como el ajeno.

Por estos motivos quiero dar las gracias a todas las personas que me han apoyado durante estos meses. Gracias Cécile, Carmen y Ramon por todo vuestro soporte, espero algún día poder corresponderos, pues sin vosotros no lo hubiera logrado. Igualmente agradezco todo el apoyo recibido por mis amigos y amigas más cercanos, vuestras muestras de afecto y consejos son muy apreciados.

Me gustaría agradecer también el soporte de los compañeros y compañeras de Sonicon Systems por su confianza, comprensión, experiencia y sobretodo por su carácter, es muy difícil encontrar un equipo humano tan bueno. En especial, gracias J. Martín, S. Mingueza, J. Torres y A. Laparra.

Para finalizar, deseo agradecer el buen trabajo y el saber hacer de Gregorio Robles, por tu compromiso en este proyecto, así como tus buenos consejos.

Gracias.

Roger Sicart Rams

11. Anexo

11.1 Manual de configuración y uso

11.1.1 Configuración de la organización

Para empezar hay que acceder al portal con las credenciales creadas después de la instalación, en mi caso el usuario “test”. Una vez dentro, hay que abrir el Panel de Control. Podemos encontrar el acceso directo dentro del menú “Ir a” en la barra superior.

En el menú de la izquierda de la pantalla, clicamos sobre “Usuarios y Organizaciones”.

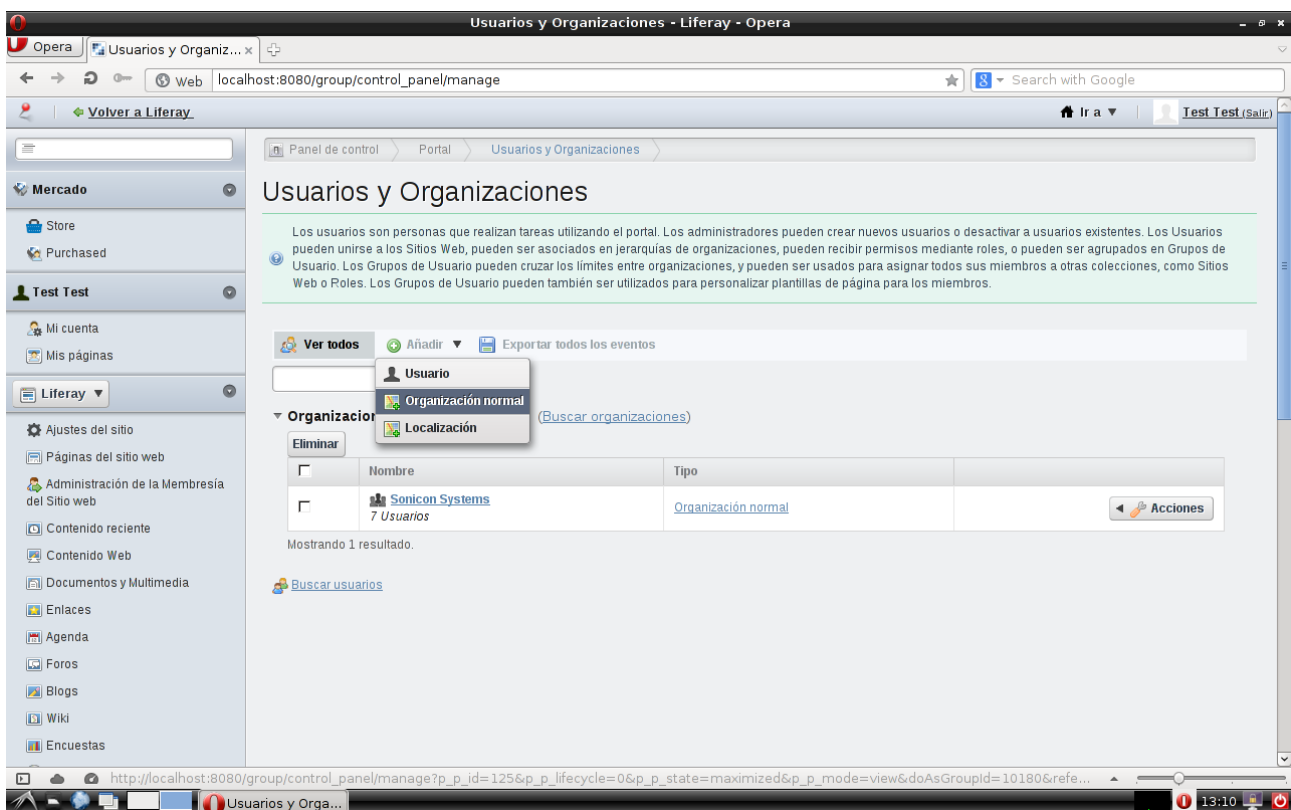


Illustration 13

Una vez dentro, rellenamos el campo nombre. Después clicamos sobre el enlace “Sitio de la organización” en el menú de la parte derecha, y marcamos la opción “Crear sitio”. Para finalizar clicamos sobre el botón “Guardar”.

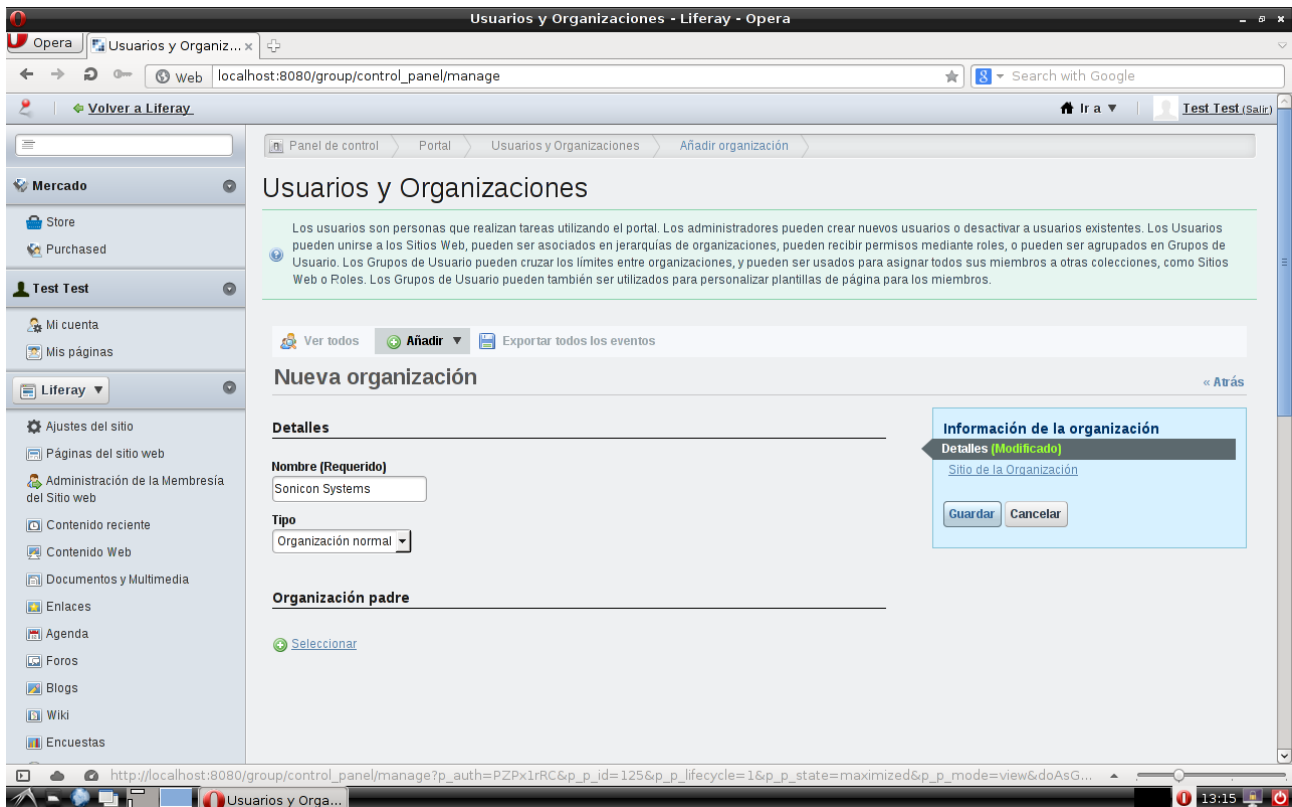


Illustration 14

La organización y su sitio ya están creados, ahora clicando sobre "Usuarios y Organizaciones" nos aparecerá en el listado de la pantalla la organización que acabamos de crear.

11.1.2 Configuración de los roles

Para el proyecto crearemos dos roles, uno para los administradores y otro para los trabajadores, con menos permisos. Para empezar, dentro del Panel de Control, clicamos sobre el "Rol" en el menú de la izquierda de la pantalla. Acto seguido podremos ver un listado de roles que vienen configurados por defecto con la instalación de Liferay.

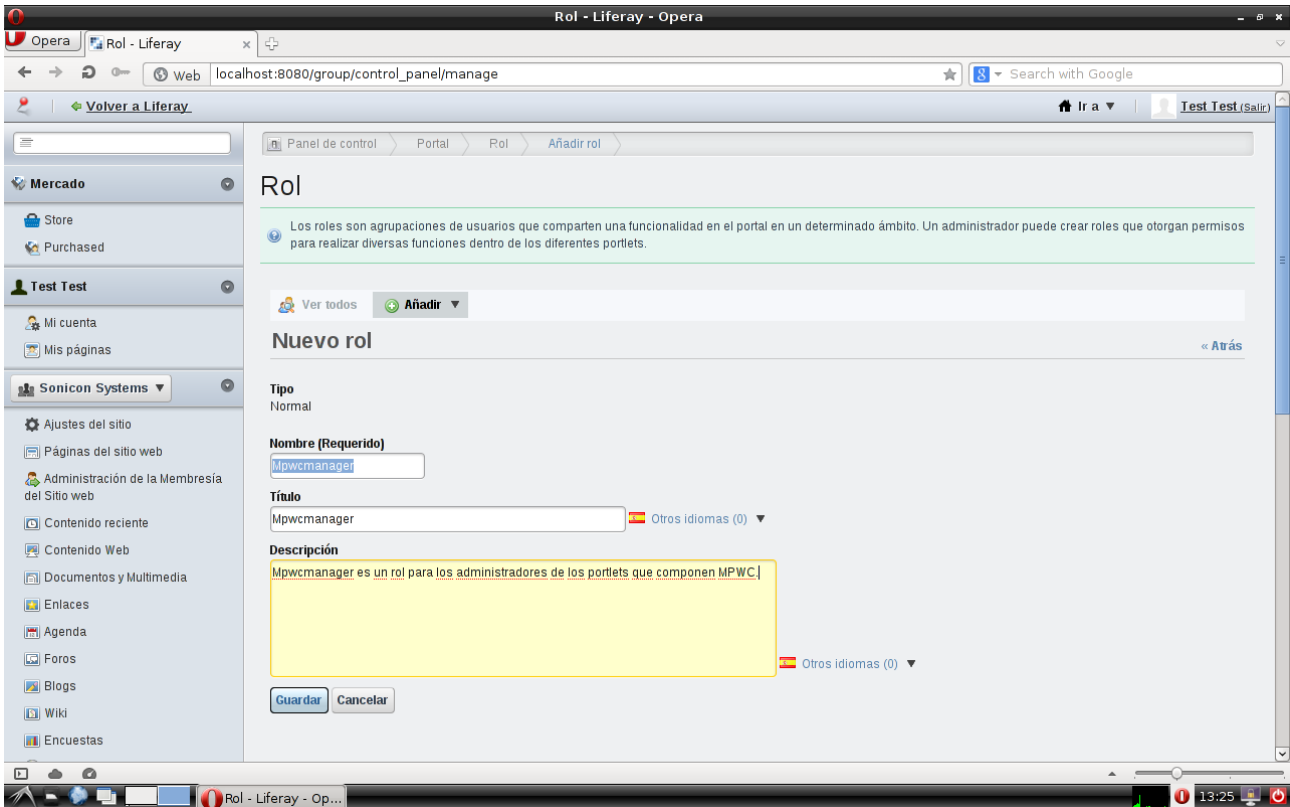


Illustration 15

Clicamos sobre “Añadir” y seleccionamos la opción “Rol de portal”. Rellenamos los campos “nombre”, “título” y “descripción”. Guardamos los cambios y repetimos el proceso para el segundo rol. Los roles creados para este proyecto tienen por nombre “MpwManager” y “MpwUser”.

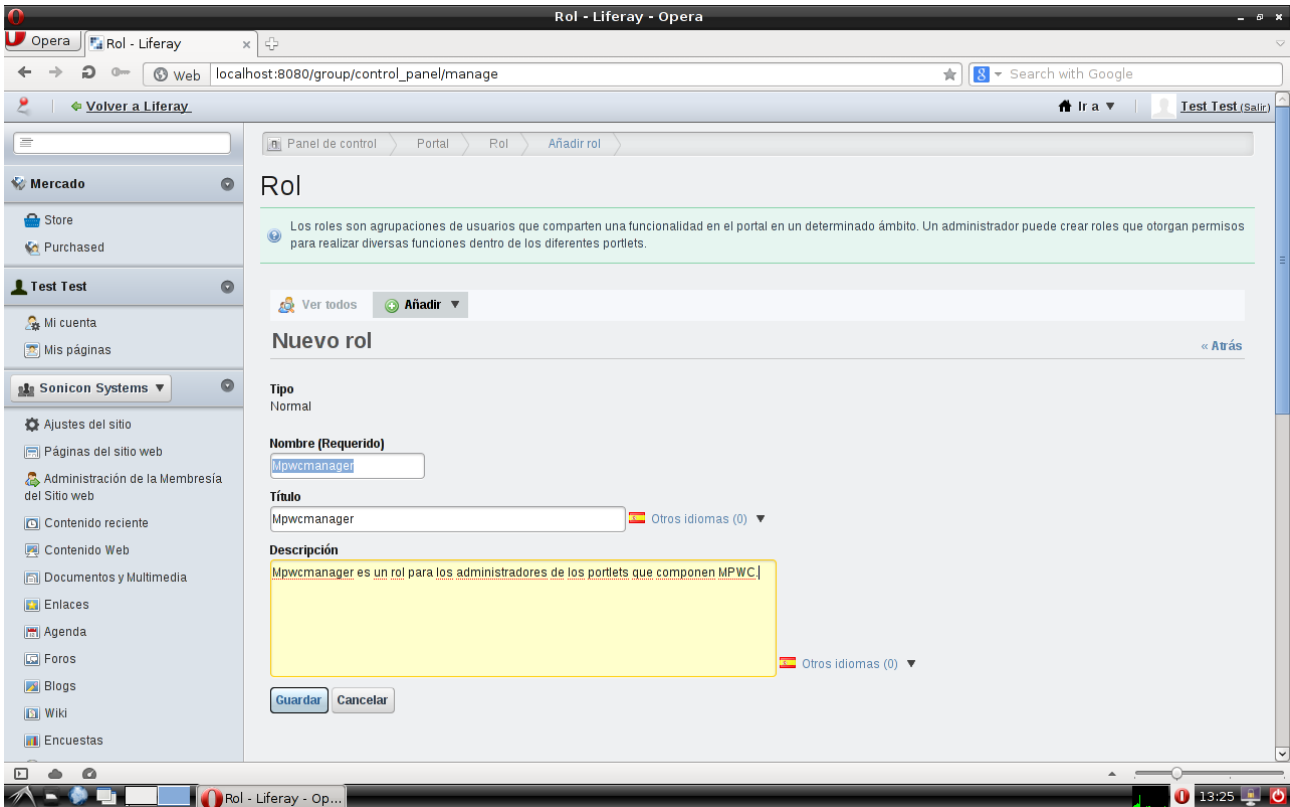


Illustration 16

En el listado de roles, para el rol “MpwUser” realizaremos una acción adicional: vincularemos el rol a la organización creada para limitarlo al ámbito de la organización de nuestro proyecto. Para hacerlo, clicamos sobre el botón “Acciones” de dicho rol, y seleccionamos “Asignar miembros”. En la pantalla siguiente clicamos sobre “Organizaciones”, luego clicamos sobre “Disponible” y seleccionamos nuestra organización y presionamos el botón “Modificar asignaciones”.

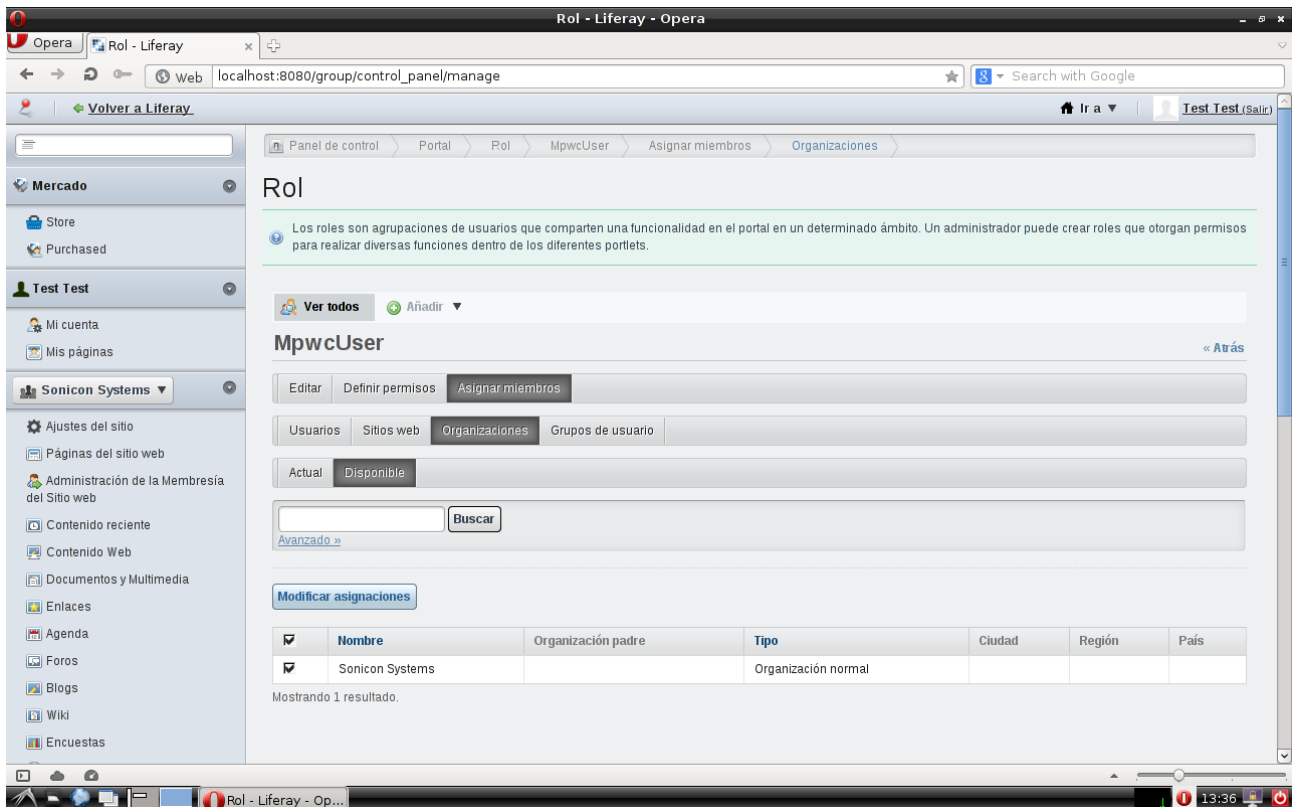


Illustration 17

Finalmente, haremos que cuando un usuario se registre, por defecto adquiera automáticamente el rol “User” del portal. De esta manera, cuando un administrador de la aplicación quiera dar de alta a un trabajador, podremos mostrar un listado de usuarios del portal y seleccionar el deseado. Así, éste pasará a ser “MpwcUser”, pues el portlet de trabajadores así lo gestiona.

En el menú de la izquierda del Panel de Control, clicamos sobre “Configuración”. Después, en el menú de la derecha clicamos sobre el enlace “Usuarios” y luego sobre el botón “Asociaciones por defecto de los usuarios”. Escribimos el nombre del rol “User” en el cuadro de texto de “Rol” y hacemos clic sobre el botón “Guardar”.

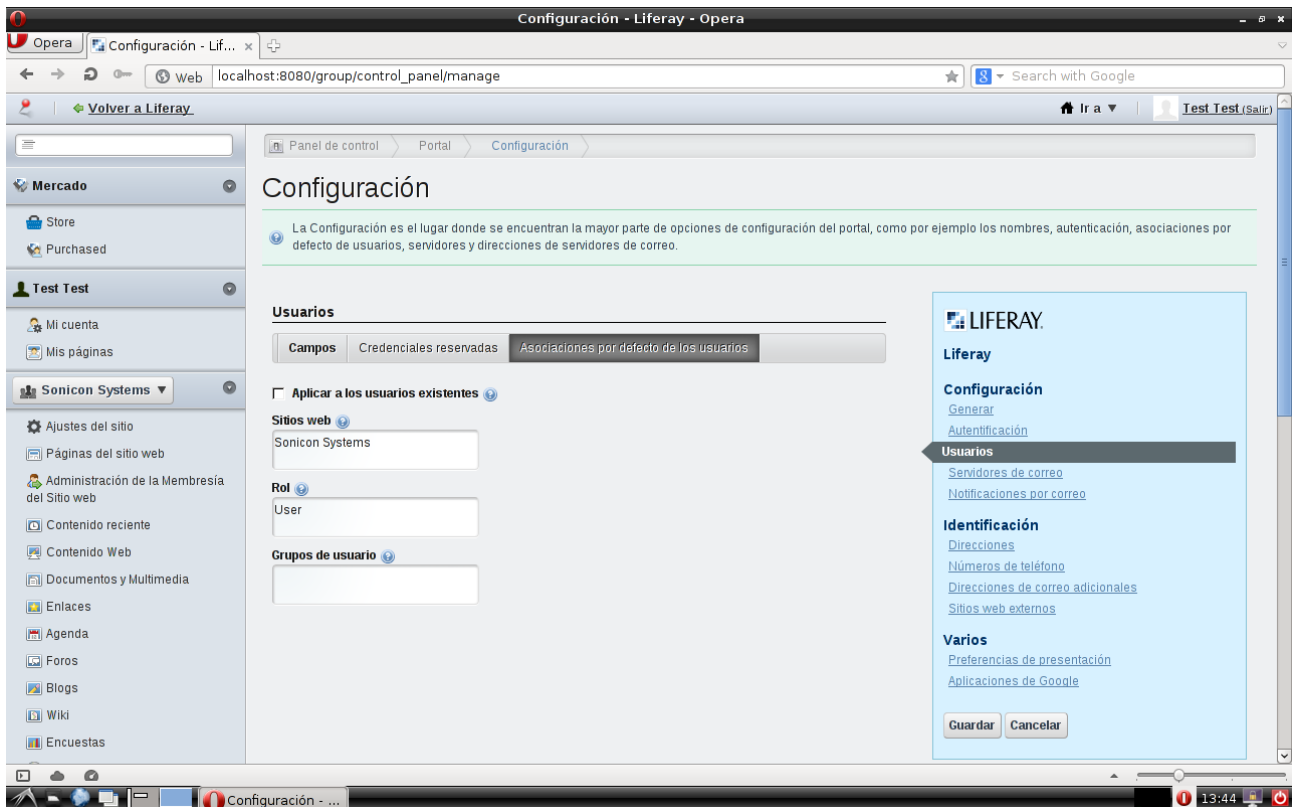


Illustration 18

11.1.3 Instalación de los portlets y el theme

Disponemos de dos métodos para instalar los plugins en el portal: mediante el Panel de Control, o copiando directamente los archivos WAR en el directorio “deploy” del portal. El primer método es más visual, y hay que añadir los módulos uno por uno. En esta guía voy a explicar cómo hacerlo por la vía rápida, con el segundo método.

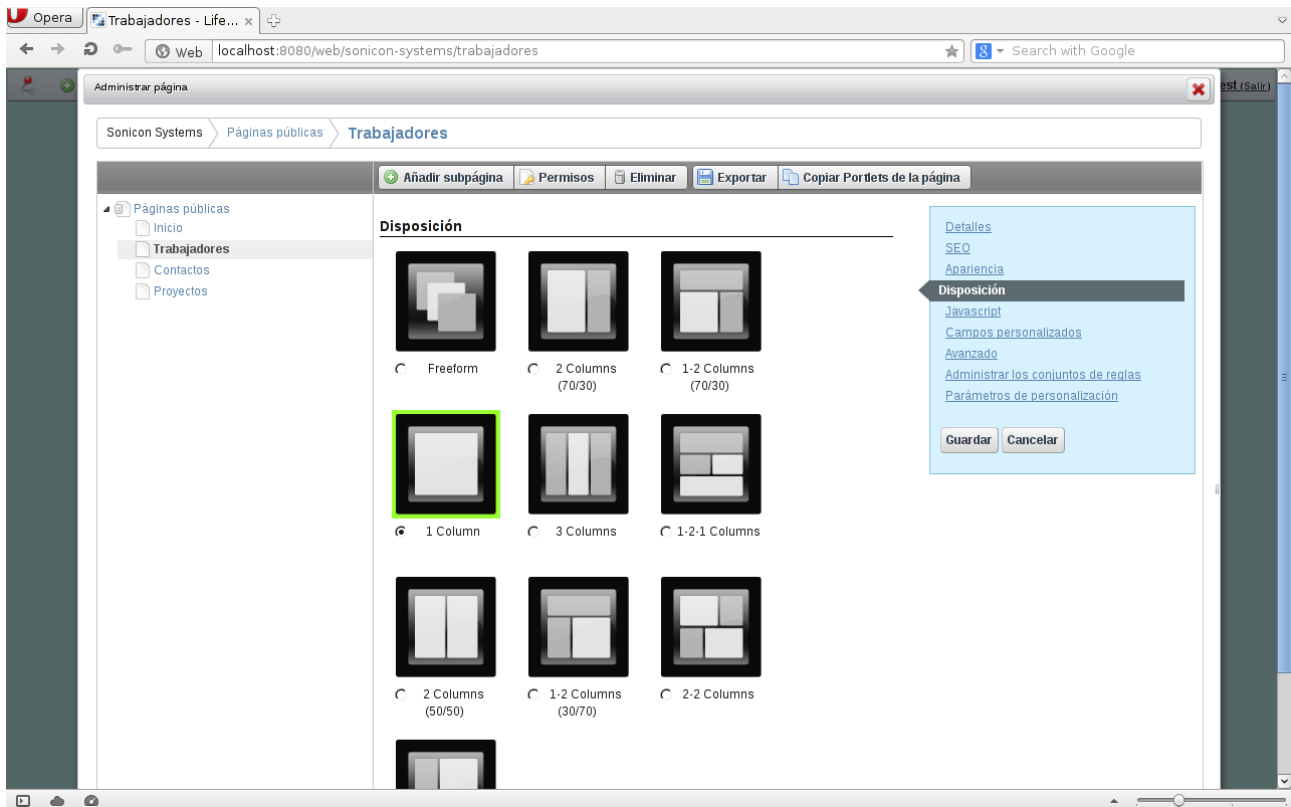
En primer lugar, debemos copiar los 5 archivos con extensión WAR que forman el proyecto al directorio “deploy” de la instalación del portal, que son:

- mpwc-cooler-theme-6.1.1.1.war
- Mpwc-portlet-6.1.1.1.war
- mpwc-contact-portlet-6.1.1.1.war
- mpwc-project-portlet-6.1.1.1.war
- mpwc-worker-portlet-6.1.1.1.war

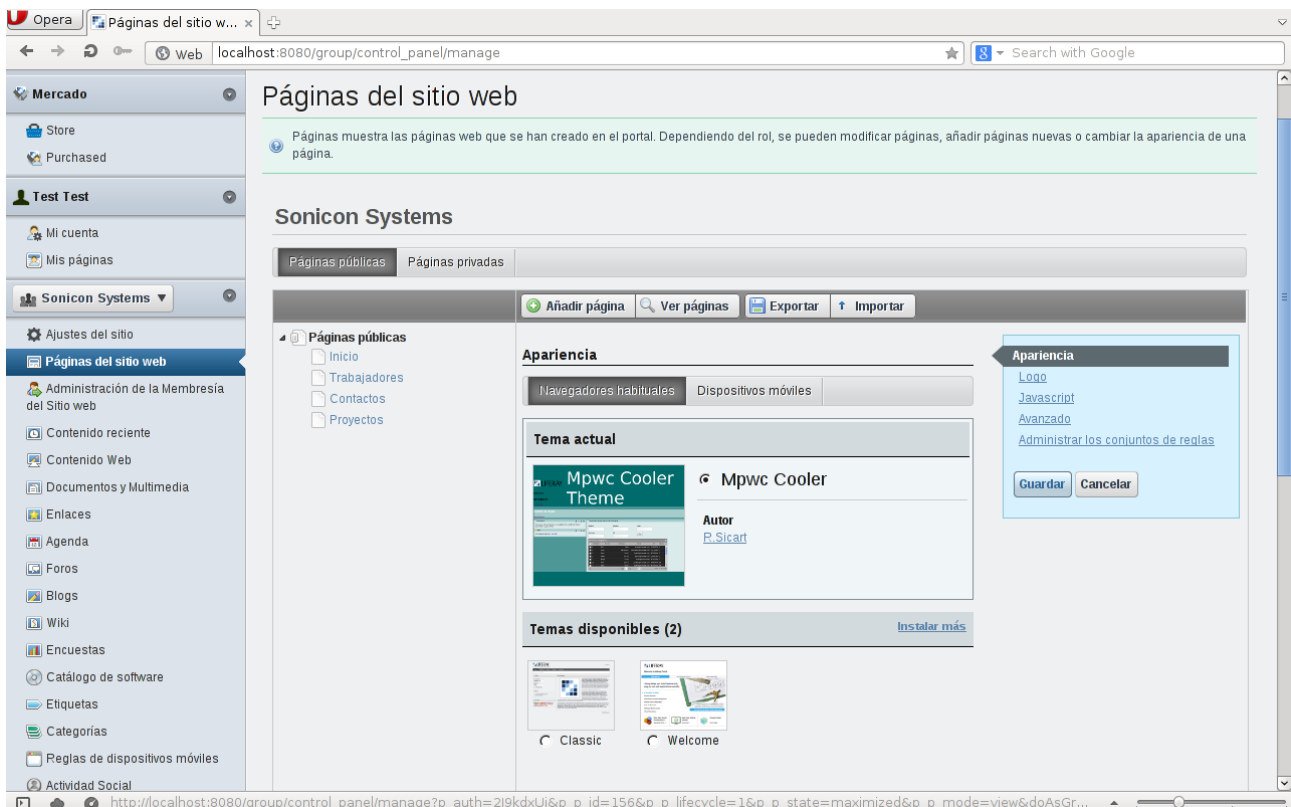
Automáticamente el portal instalará los portlets y aparecerán como disponibles para usar en el menú superior izquierdo, llamado “Añadir”. Antes de nada, hay que abrir el sitio público de Sonicon Systems, pues añadiremos los portlets para esta organización. Para hacerlo, vamos al menú “Ir a” y marcamos la opción “Sonicon Systems”, o abrimos la URL <http://localhost:8080/web/sonicon-systems>. Veremos que solamente hay una página vacía, con un enlace en el menú dónde hay escrito “Inicio”. Pues bien, añadiremos tres páginas más, con los nombres “Trabajadores”, “Contactos” y “Proyectos”. Para hacerlo, debemos clicar sobre el menú “Añadir”, en la parte superior izquierda, y seleccionar “Añadir página”. Hay que hacerlo un total de tres veces, una para cada uno de los nombres comentados. Después veremos que en el menú ya tenemos todos los enlaces necesarios, sólo falta añadir el portlet necesario a cada página.

Primero, abrimos la página de “Trabajadores”, y una vez dentro, vamos al menú “Añadir” , clicamos sobre la opción “Más...”, clicamos sobre la categoría “Desarrollo”; veremos que aparecen nuestros 4 portlets. Para instalar el portlet Mpwc Worker en la página de “Trabajadores” (que teníamos abierta), simplemente arrastramos la opción del menú hacia la página. Automáticamente aparecerá la ventana principal del portlet en nuestra página de “Trabajadores”. Debemos repetir este proceso -arrastrar cada portlet a su página- una vez para cada portlet.

Una vez metido cada portlet en su página, vamos a configurar la disposición o *layout*. Vamos al menú superior “Administrar” y marcamos “Disposición de página”. Se abrirá una ventana donde debemos seleccionar “1 Column” y pulsar el botón “Guardar”. Ahora clicamos sobre “Contactos”, en el menú de la izquierda, luego clicamos sobre “Disposición”, en el menú de la derecha, y para acabar seleccionamos “1 Column” y pulsamos el botón “Guardar”. Repetimos los mismos pasos para “Proyectos”.



Una vez configurados los portlets, vamos a configurar el *theme*. Debemos abrir el Panel de Control, en el apartado de “Sonicon Systems” clicar sobre el enlace “Páginas del Sitio”. En la ventana que se abre, debemos seleccionar el tema “Mpwco Cooler” y pulsar el botón “Guardar”.



11.1.4 Asignación de permisos

Accedemos al Panel de Control, hacemos clic en “Rol”, luego en el listado de roles, presionamos el botón de “Acciones” del rol “MpwcUser” y clicamos sobre la opción “Definir permisos”. En el desplegable que tiene por título “Añadir permisos”, dentro del subapartado “Contenido del Sitio Web”, seleccionamos la opción “Manage Workers”. Aparecerá una lista de acciones que se pueden llevar a cabo sobre la entidad “Worker”, por parte de cualquier usuario que tenga el rol que estamos configurando. Para el caso de “MpwcUser”, queremos que solamente puedan ver los trabajadores, que no puedan ni eliminar, ni modificar. Seleccionamos el *checkbox* de “Ver” y presionamos el botón “Guardar”.

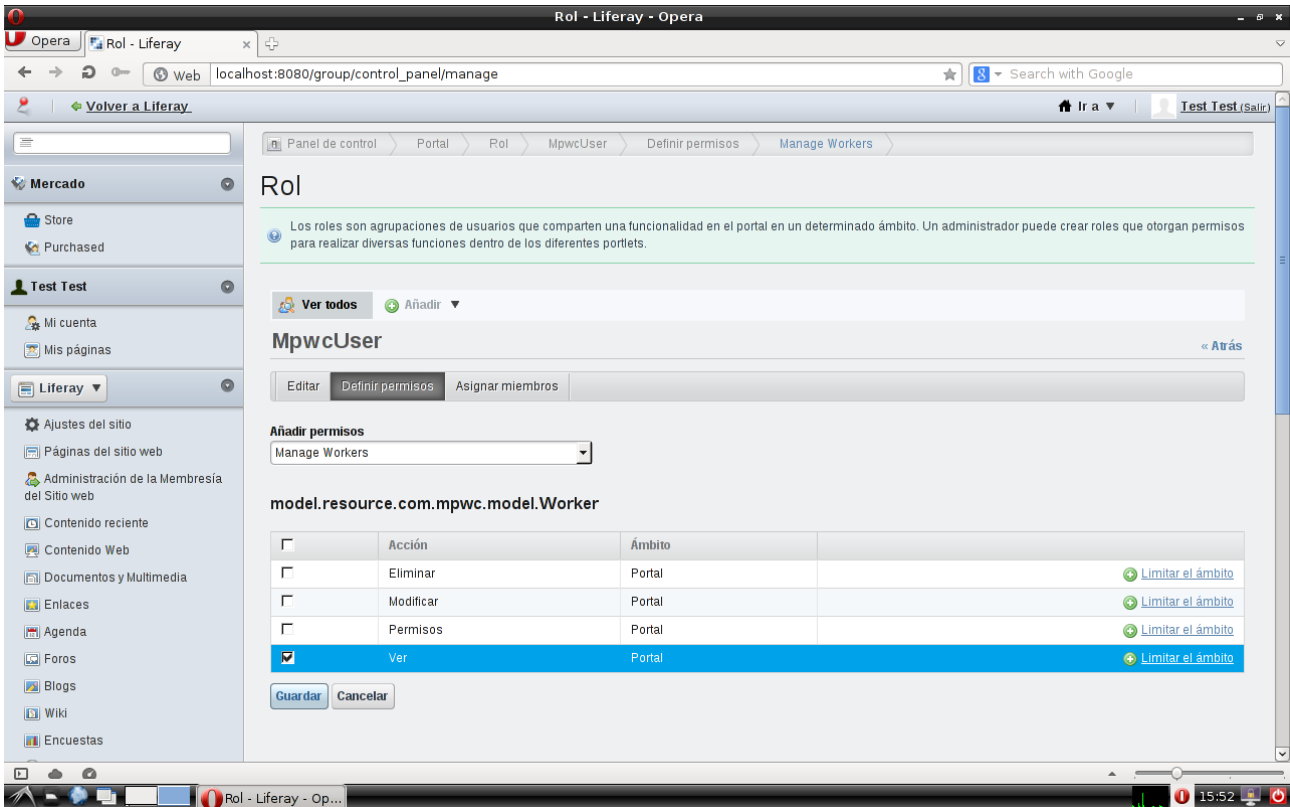


Illustration 19

Después, en la misma pantalla de definición de permisos del rol, volvemos a abrir el desplegable con nombre “Añadir permisos”, y dentro del subapartado “Aplicaciones del Sitio”, clicamos sobre la opción “Manage Workers”. En el listado que aparece, marcamos solamente el *checkbox* de “Ver” y clicamos sobre el botón de “Guardar”.

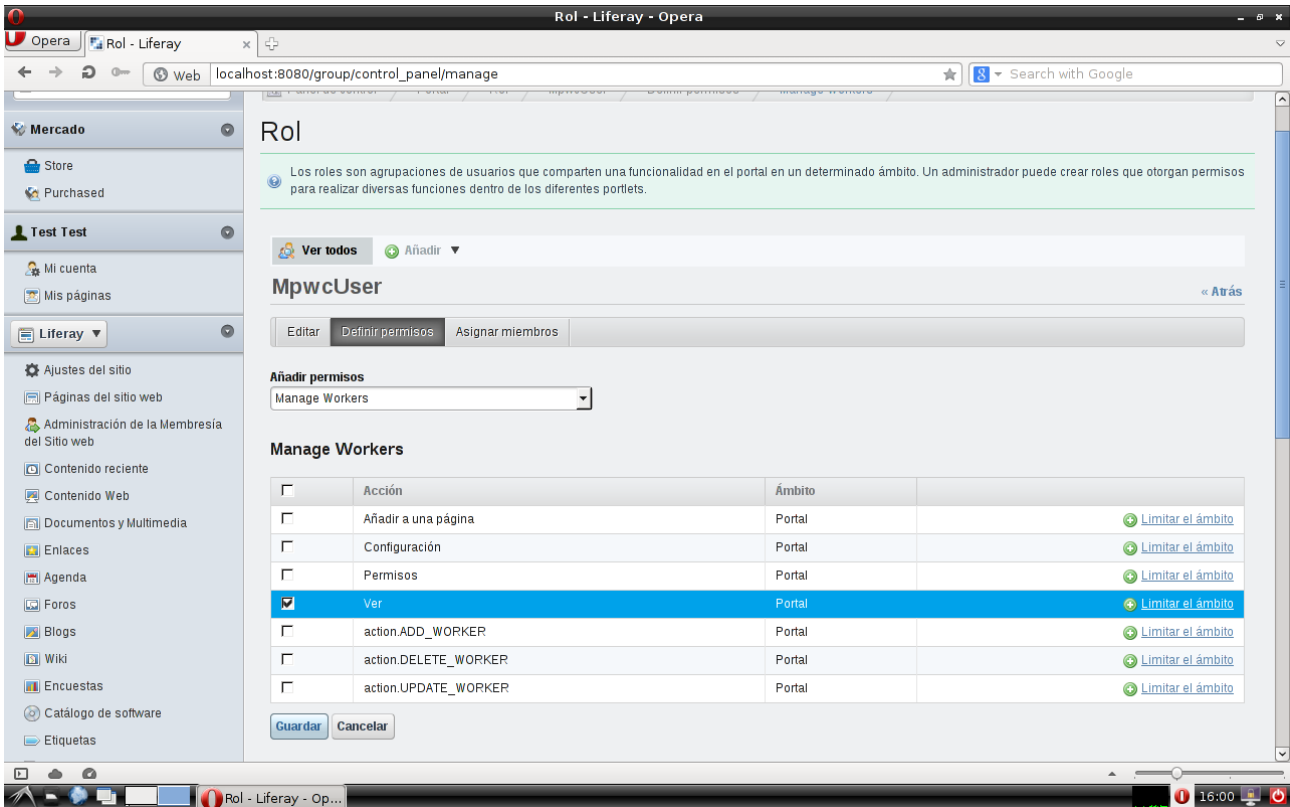


Illustration 20

Para el resto de portlets de gestión (items Mpowc Project y Mpowc Contact del desplegable), debemos repetir el proceso anterior. No hay que olvidar de definir los permisos del desplegable para los dos subapartados, como se ha comentado anteriormente, con los nombres “Contenido del Sitio” y “Aplicaciones del Sitio”. Al final tendremos un listado similar de permisos -asignados al rol “MpowcUser”- como el que aparece en la siguiente imagen:

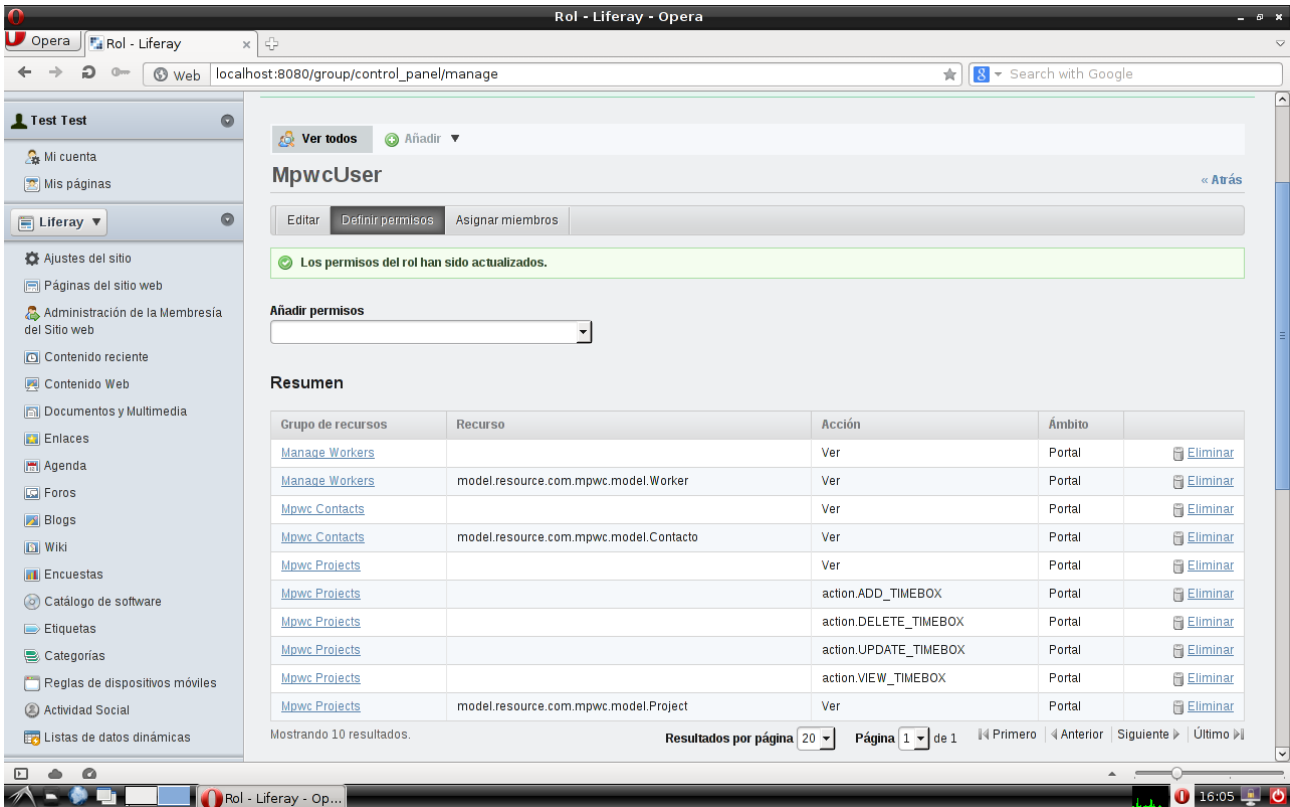


Illustration 21

Bien, hemos acabado con el rol de usuarios, ahora hay que repetir el proceso entero con el rol “MpwManager”. A este rol, para el subapartado de “Contenido del Sitio”, le podemos asignar, además del permiso de “Ver”, los permisos de “Modificar”, “Eliminar”. Para el subapartado de “Contenido del Sitio” le añadiremos los propios de cada portlet. Por ejemplo, en el caso del portlet de gestión de proyectos, seleccionaremos los permisos “action.ADD_PROJECT”, “action.UPDATE_PROJECT”, “action.DELETE_PROJECT”, “action.ADD_WORKER_PROJECT” y “action.DELETE_WORKER_PROJECT”. También seleccionaremos todos los llamados “action.XXXXXX_TIMEBOX”. Concretamente, los permisos para los proyectos deberían quedar similares al listado de la siguiente imagen:

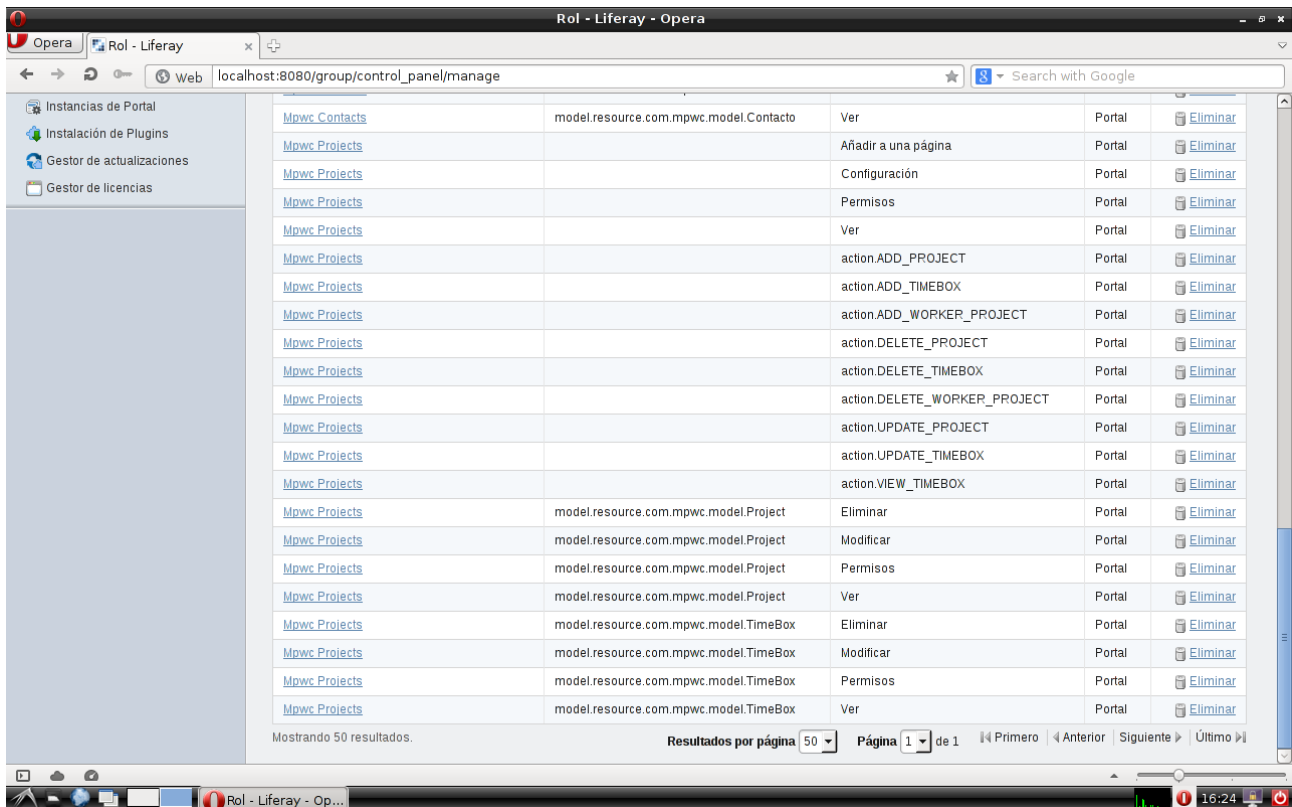


Illustration 22

En general, a los administradores podemos añadirles todos los permisos disponibles para cada portlet.

11.1.5 Creación de un usuario administrador

Para utilizar la aplicación debidamente, sólo nos queda crear un usuario administrador, al que le asignaremos el rol “MpowcManager”. Con este usuario, que en mi caso lo he llamado “mpwcproject”, llevaremos a cabo todas las operaciones de los capítulos siguientes, acerca del uso de los diferentes portlets. Para crearlo, en la página inicial del portal haremos clic sobre “Crear cuenta”, en la parte inferior de la pantalla. Rellenaremos los datos del formulario y la pregunta y respuesta secreta. Luego aceptaremos los términos y condiciones, y por último introduciremos una nueva contraseña para la cuenta. Una vez hecho esto, cerraremos la sesión y entraremos con el usuario administrador del portal que se crea por defecto con la instalación, en mi caso el usuario Test. Accedemos al Panel de Control, hacemos clic en la opción “Rol” del menú izquierdo. En el listado de roles que aparece, pulsamos el botón “Acciones” del rol “MpowcManager”, y seleccionamos

la opción “Asignar miembros”. En la página que se abre, clicamos sobre “Disponible”, seleccionamos el usuario que acabamos de crear y pulsamos el botón “Modificar asignaciones”. Opcionalmente también podemos asignarle el rol de administrador de portal, repitiendo el procedimiento anterior. Acto seguido debemos salir del Panel de Control, e ir a la página de “Trabajadores”. Pulsamos el botón “Añade”, hacemos clic sobre el enlace “Añade” del usuario anterior, rellenamos el formulario que aparece con los campos necesarios y pulsamos “Guardar”. Ahora ya disponemos de un usuario administrador de la aplicación, miembro y trabajador de Sonicon Systems.

Remarcar que es importante la creación del usuario anterior, y su asignación como trabajador de la empresa, pues hay pantallas donde se muestra información en la que se requiere este estatus (ser trabajador de la empresa), y si se accediera con un usuario que no lo es nos aparecería un error y el comportamiento no sería el esperado.

11.1.6 Uso de Mpwc Worker

En la página principal de este módulo aparece un listado con todos los trabajadores de la empresa. En la parte superior de esta lista, podemos ver unos campos, por los cuales podemos filtrar para obtener listados más concretos.

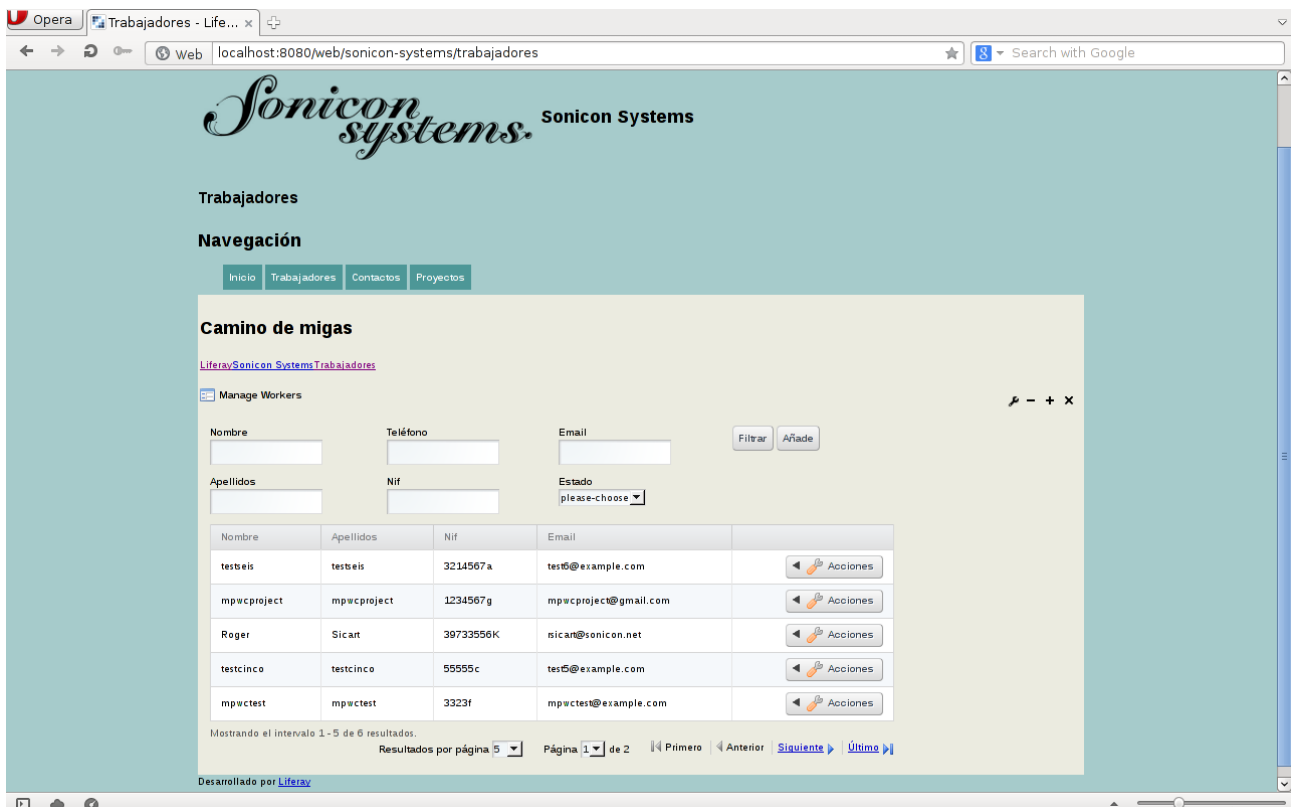


Illustration 23

Así para obtener un listado de todos los trabajadores con una cuenta de correo que acabe con la cadena “@sonicon.net”, introduciremos dicha cadena en el filtro “Email” y pulsaremos el botón “Filtrar”.

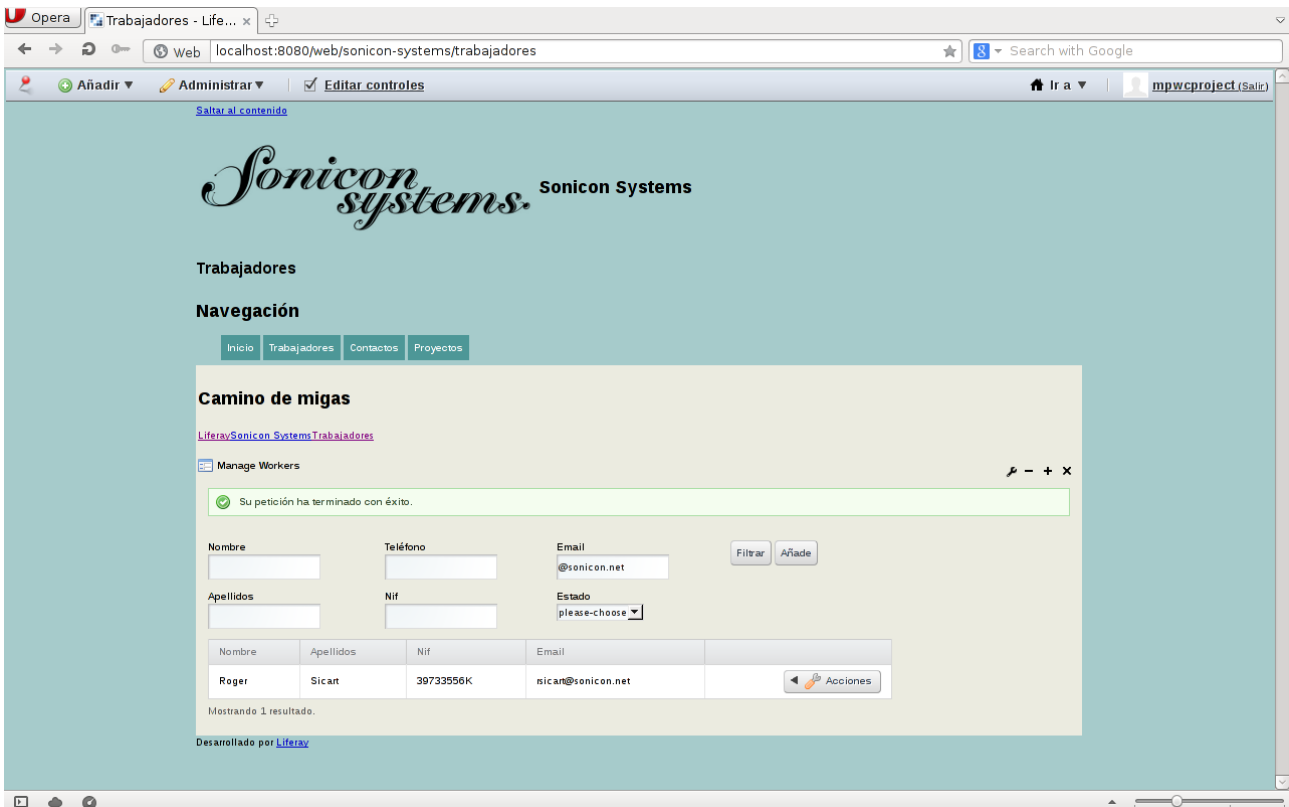


Illustration 24

Para eliminar un trabajador, clicamos sobre el botón “Acciones”, y seleccionamos la opción “Eliminar”. De la misma manera, para modificar la información de un trabajador, seleccionaremos la opción “Editar”. En la pantalla de edición, aparece la información del trabajador, además de un listado de proyectos en los que está trabajando.

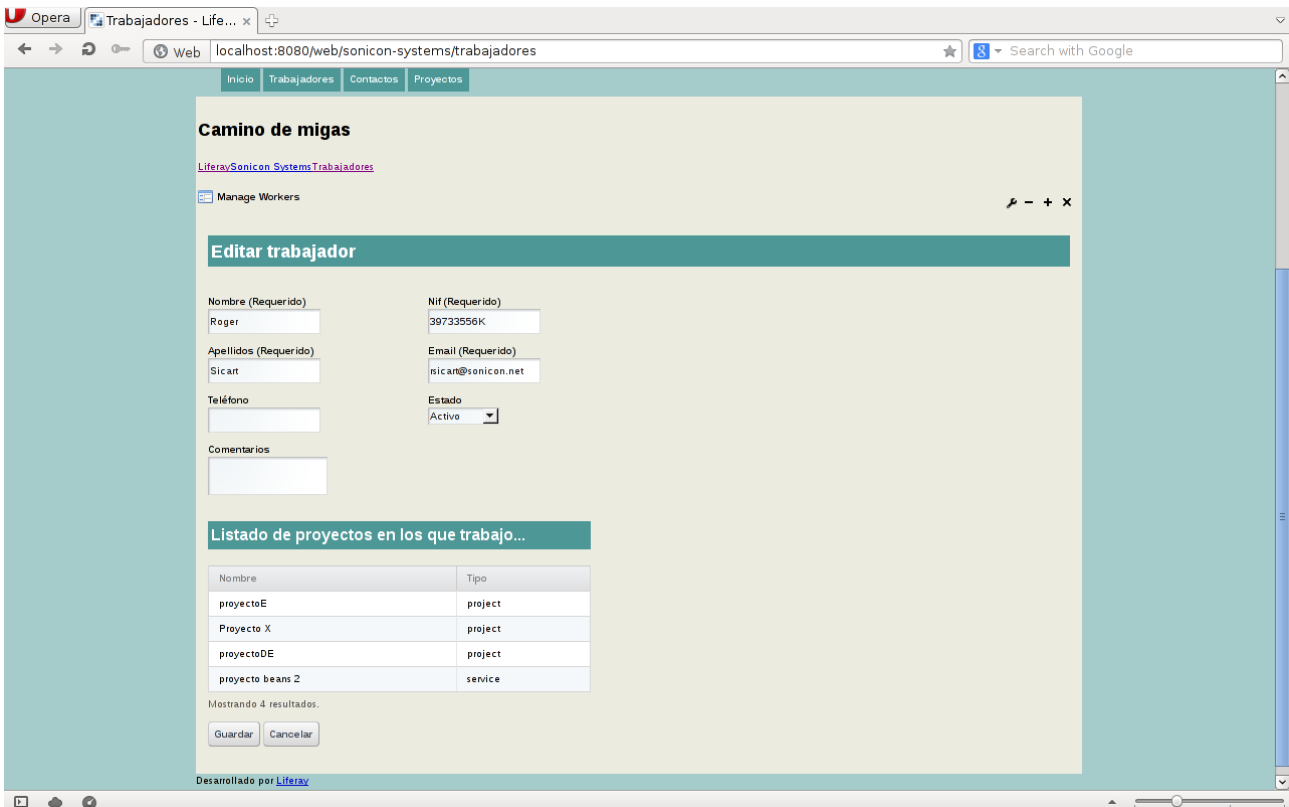


Illustration 25

Una vez modificado pulsamos el botón “Guardar” para salvar los cambios.

En la pantalla principal, también podemos añadir un nuevo trabajador pulsando el botón “Añade”. Éste, nos lleva a un listado de usuarios -recordemos que están asociados al rol User- del portal. Si pulsamos el enlace “Añade” del usuario deseado, éste nos llevará a una pantalla con un formulario relleno parcialmente con los datos del usuario. Si en esta pantalla pulsamos el botón “Guardar”, el usuario pasará a formar parte de la organización creada, en nuestro caso “Sonicon Systems”, con el rol configurado “MpowcUser” por defecto.

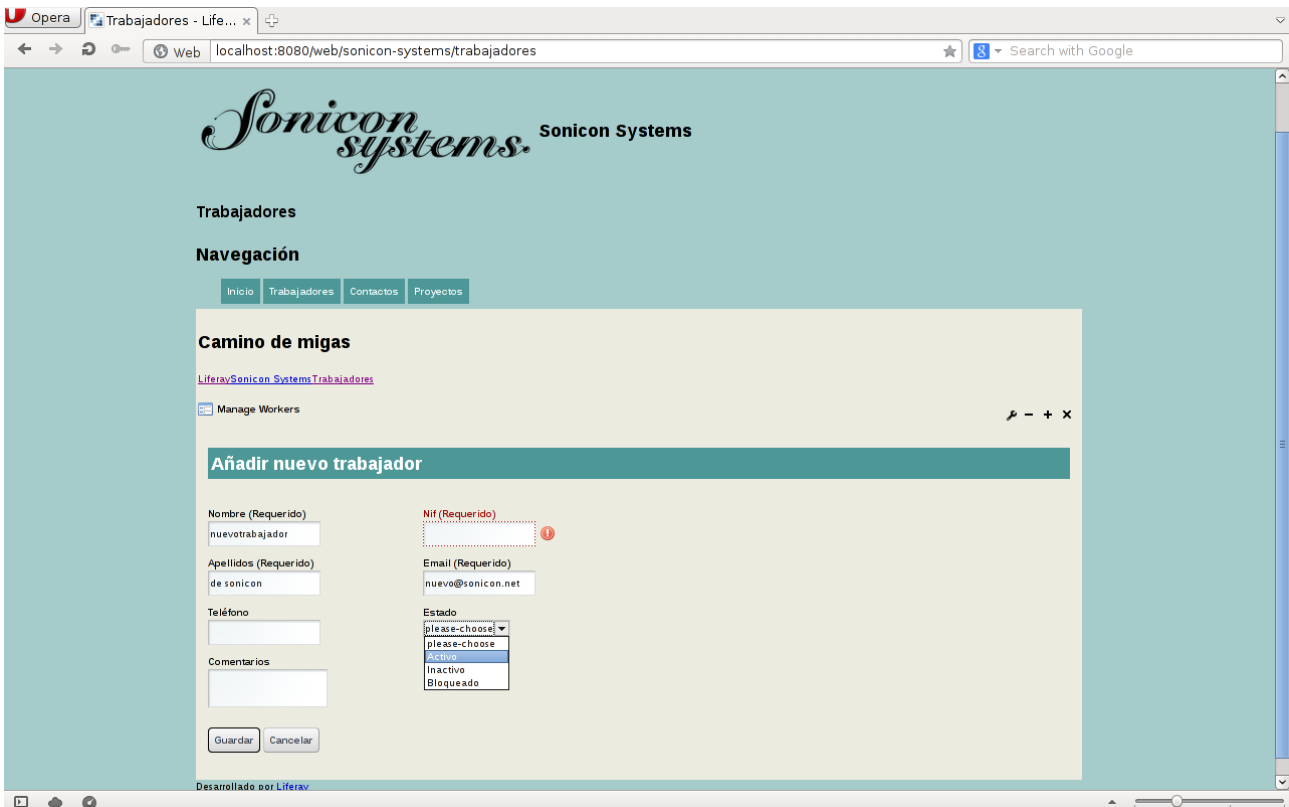


Illustration 26

11.1.7 Uso de Mpwc Project

Del mismo modo que para el portlet anterior, en la pantalla principal del portlet de gestión de proyectos aparece un listado. En la parte superior del listado disponemos de una serie de filtros para buscar proyectos en función de ciertos criterios. Por ejemplo, si queremos encontrar proyectos de tipo “Proyecto”, con estado “Activo”, una fecha de fin máxima el “25 de diciembre de 2013” y con la opción “Si” del desplegable “Imputar tiempo en el proyecto”, debemos seleccionar los respectivos valores y pulsamos el botón “Filtrar”. En el listado aparecerán los proyectos que coincidan con estos criterios. El funcionamiento es el mismo para todos los filtros.

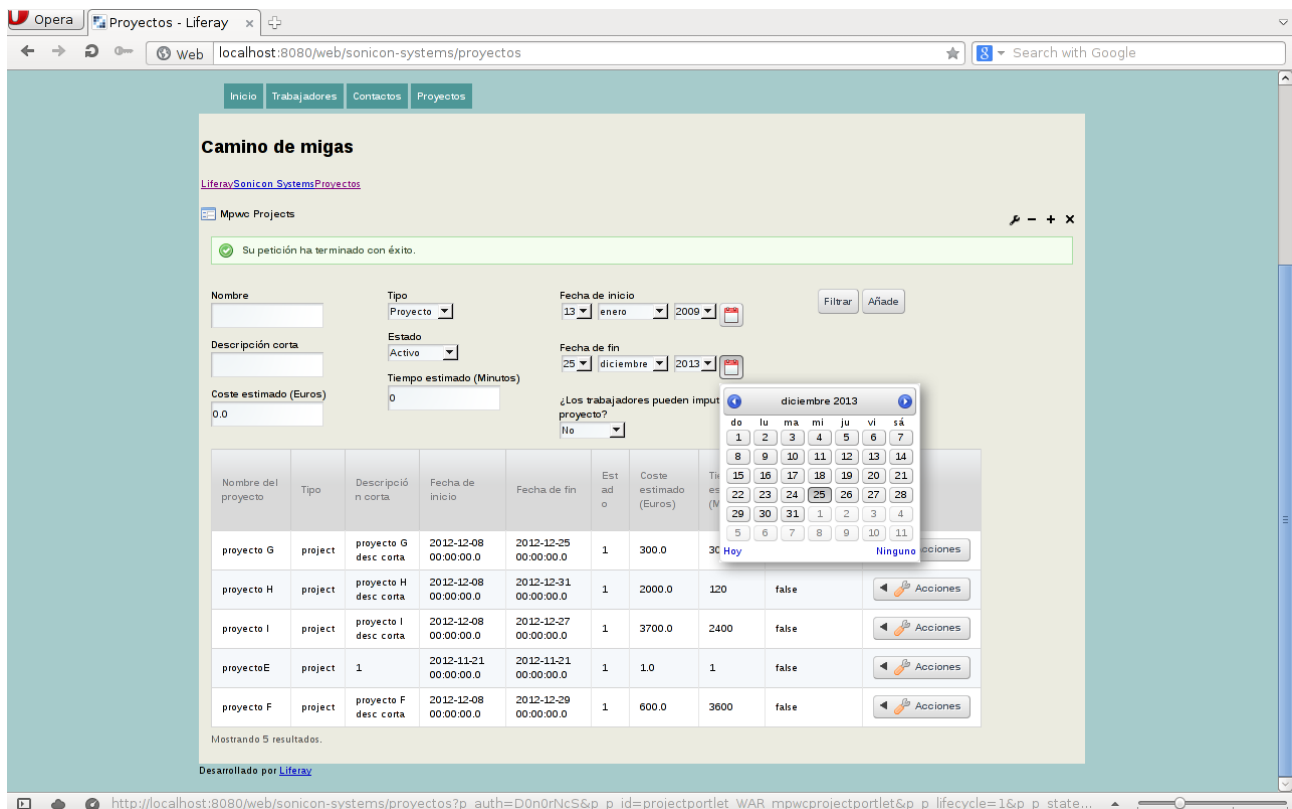


Illustration 27

Ahora, para cada proyecto del listado, si pulsamos el botón “Acciones” veremos que tenemos disponibles una serie de opciones, en función de los permisos del usuario con el que hemos accedido a la aplicación. Con un rol de administrador veremos todas las opciones, que son: ver, editar, eliminar, añadir tiempo y ver tiempo.

La opción “Ver” nos muestra los detalles del proyecto, junto con los trabajadores que trabajan en él. Es interesante remarcar que aparece también el cliente, el total de tiempo imputado en el proyecto y quién es el jefe de proyecto, si lo hay.

En segundo lugar, la opción “Editar” nos permite editar la información del proyecto: su nombre, estado, fechas de inicio y fin, etc. También podemos asignar o quitar trabajadores o definir un jefe de proyecto. Para esto último, basta con clicar el botón “Acciones” de alguno de los trabajadores del proyecto y marcar la opción “Jefe de proyecto”. Al finalizar los cambios pulsamos “Guardar”.

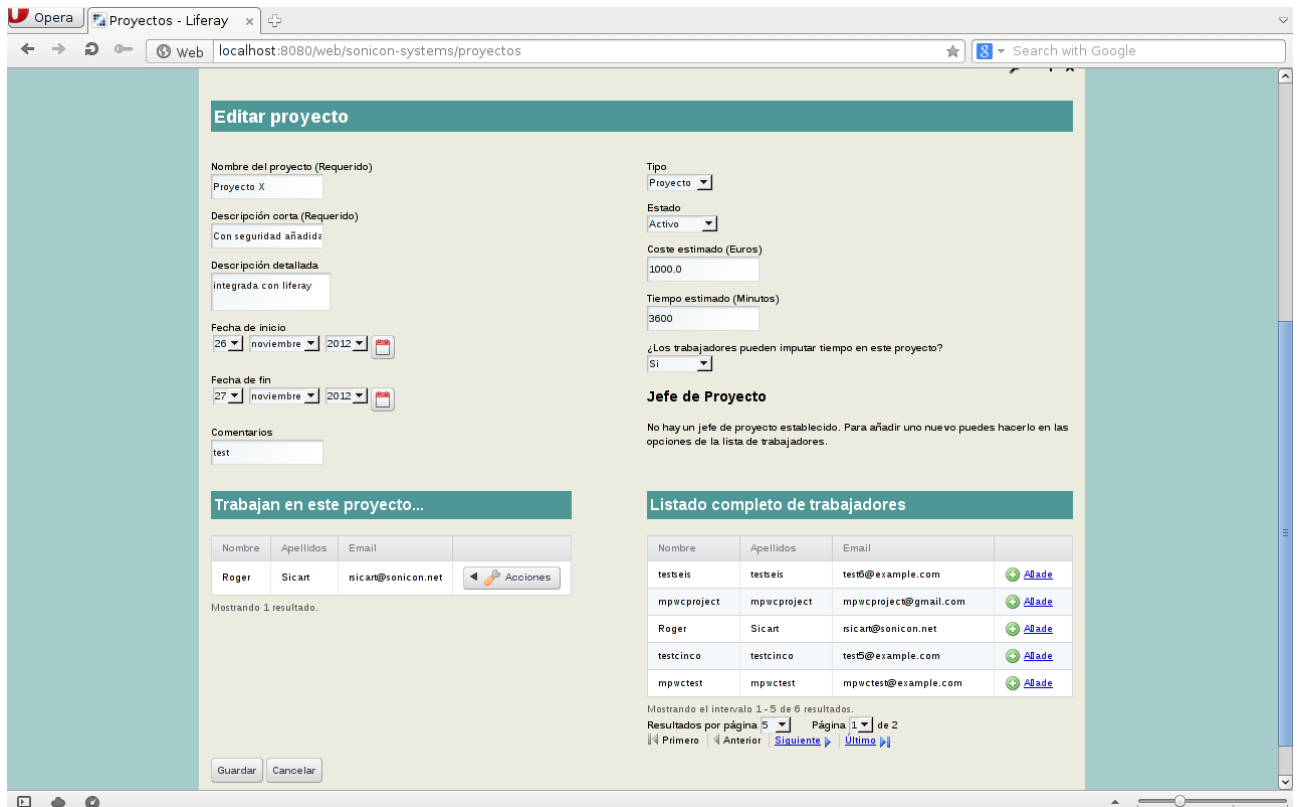


Illustration 28

La opción “Eliminar”, es trivial, se encarga de borrar el proyecto de la base de datos.

La cuarta opción, “Añadir tiempo”, nos presenta un formulario donde podemos introducir la cantidad de minutos que queremos imputar al proyecto, los comentarios y la fecha en la que hemos dedicado esos minutos al proyecto.

Como última opción tenemos “Ver tiempo”, donde podemos ver una página con un listado de “Timebox” o tiempos propios introducidos para el proyecto en cuestión, un formulario para añadir un nuevo tiempo (igual que en la opción “Añadir tiempo”), y un listado de trabajadores que trabajan en el proyecto con el total de tiempo que han dedicado a ese proyecto. En el listado de tiempos propios tenemos las opciones de editar un tiempo introducido o eliminarlo.

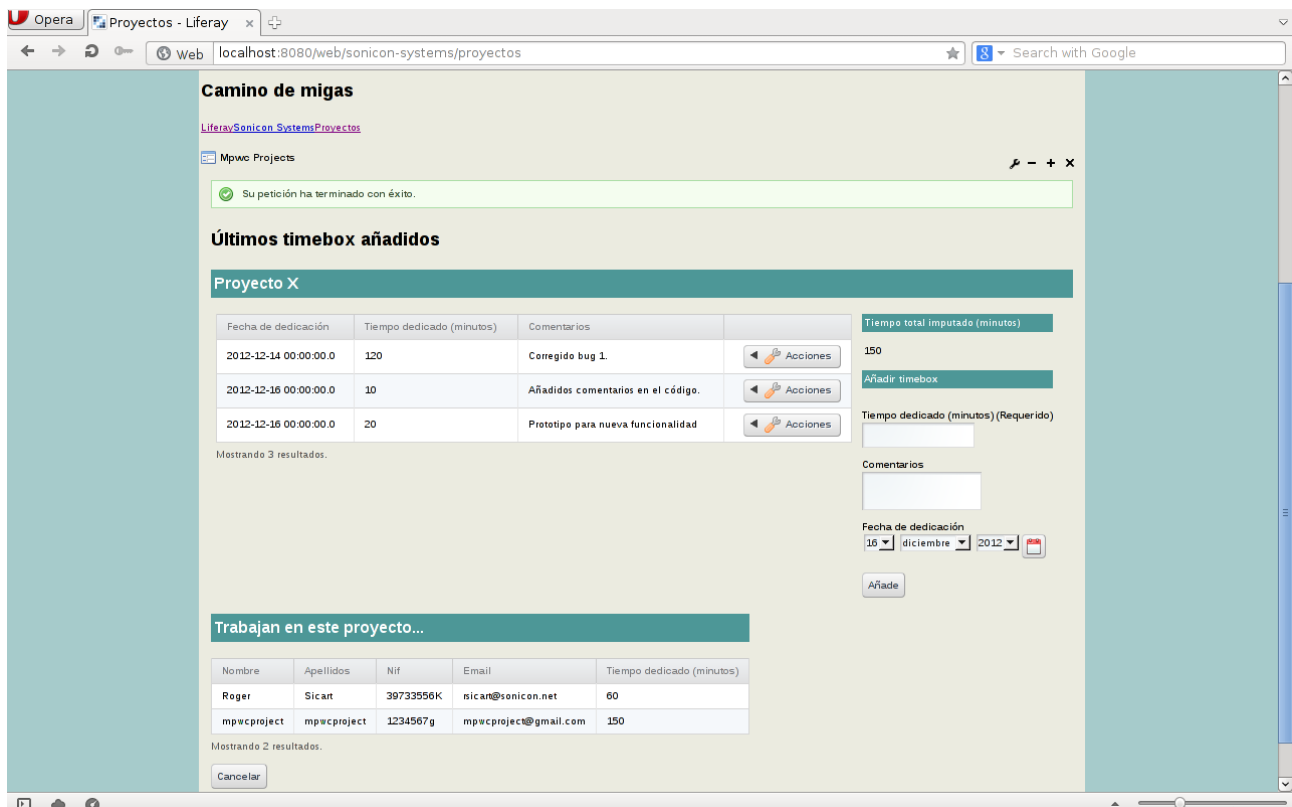


Illustration 29

Para añadir un nuevo proyecto, solamente tenemos que pulsar el botón “Añade” en la página inicial del portlet, situado junto al botón “Filtrar”. Se abrirá una página con un formulario que debemos rellenar, y pulsar el botón “Guardar”. Es importante cumplimentar correctamente todos los campos, de esta forma nuestra base de datos será más completa, y las páginas mostrarán información más detallada.

11.1.8 Uso de Mpwc Contact

En la página principal del portlet aparece un listado de contactos de la empresa. Éstos, pueden ser clientes o proveedores. En la parte superior tenemos múltiples filtros, por los que podemos buscar contactos que cumplan ciertos criterios. Por ejemplo, si queremos encontrar todos los clientes de Tarragona, seleccionamos como tipo de contacto “Cliente” e introducimos como filtro de ciudad “Tarragona”. Pulsamos el botón “Filtrar”.

Para cada contacto de la lista, tenemos un botón de “Acciones” con las opciones ver, editar o eliminar. La primera opción nos muestra información detallada del contacto, que incluye el listado de proyectos en los que trabajamos para él. En la opción “Editar”, se

abre una página con un formulario con los datos del contacto, los cuales podemos modificar. Además, podemos asignar o quitar proyectos, en función de si trabajamos o no para él en dicho proyecto. Al acabar, pulsamos “Guardar” para salvar los cambios.

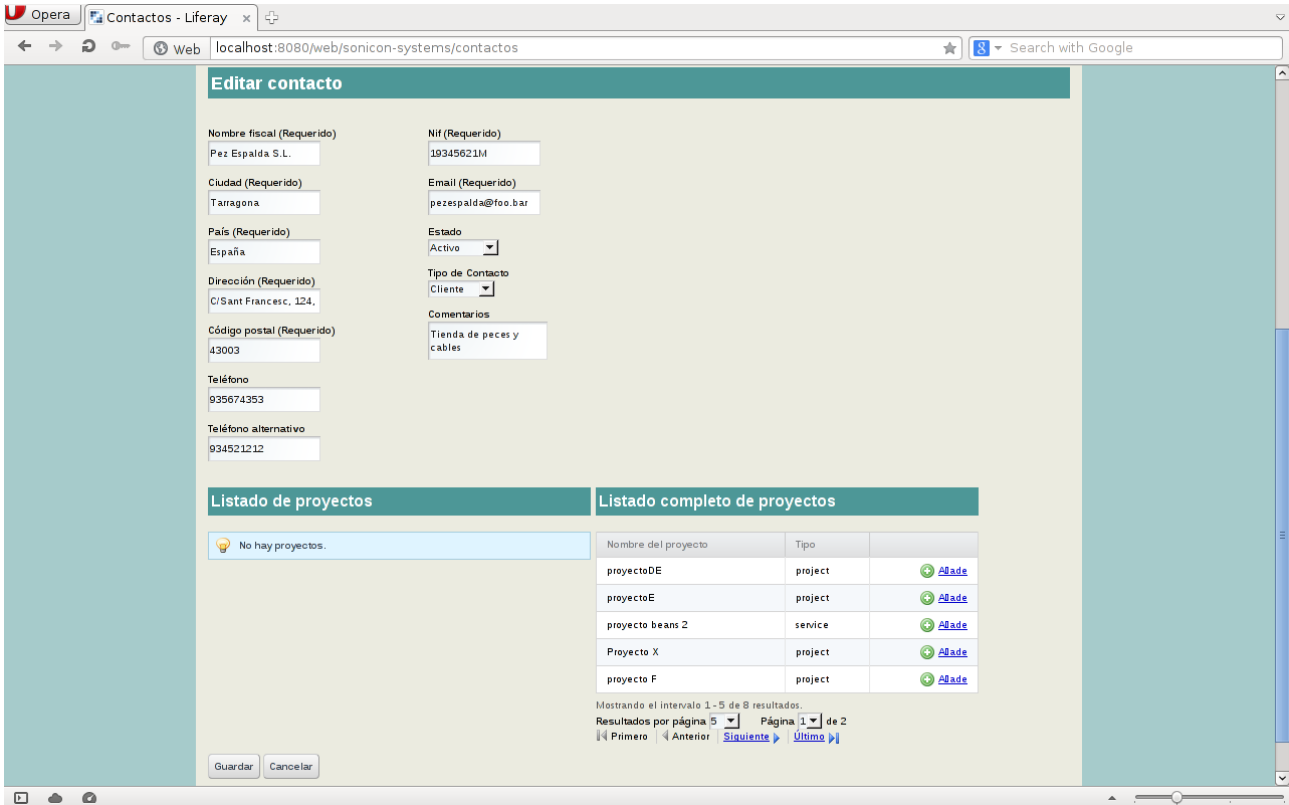


Illustration 30

Como tercera y última opción tenemos “Eliminar”, acción con la cual borraremos el contacto de la base de datos.

11.2 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.