



Protocolo de seguridad y salud

Memoria del Proyecto Final de Grado

Grado Multimedia

Ingeniería web

Autor: Cristina Calvo Mendoza

Consultor: Ignasi Lorente Puchades

Profesor: Carlos Casado Martínez

13-01-2015

Copyright

La documentación de este Trabajo de Fin de Grado se rige por la licencia **Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0)**, que indica:

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

Adaptar — remezclar, transformar y crear a partir del material

El licenciador no puede revocar estas libertades mientras cumpla con los términos de la licencia.

Bajo las condiciones siguientes:



Reconocimiento — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador, pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra.



NoComercial — No puede utilizar el material para una finalidad comercial.



CompartirIgual — Si remezcla, transforma o crea a partir del material, deberá difundir sus contribuciones bajo la misma licencia que el original.

No hay restricciones adicionales — No puede aplicar términos legales o medidas tecnológicas que legalmente restrinjan realizar aquello que la licencia permite.

No hay restricciones adicionales — No puede aplicar términos legales o medidas tecnológicas que legalmente restrinjan realizar aquello que la licencia permite.

Avisos:

No tiene que cumplir con la licencia para aquellos elementos del material en el dominio público o cuando su utilización esté permitida por la aplicación de una **excepción o un límite**.

No se dan garantías. La licencia puede no ofrecer todos los permisos necesarios para la utilización prevista. Por ejemplo, otros derechos como los de **publicidad, privacidad, o los derechos morales** pueden limitar el uso del material.

Dedicatoria/Cita

A todos mis *maestros* que enriquecieron mi conocimiento y sabiduría.

Resumen

Este proyecto consiste en la elaboración de una aplicación web donde se gestionará la información y documentación de las obras de la Diputación de Barcelona – Gerencia de servicios de infraestructuras Viarias y Movilidad, cuya seguridad se rige en base a los Reales Decretos, Leyes y Normativas Técnicas recogidas en el Instituto Nacional de Seguridad e Higiene en el trabajo.

Esta aplicación se desarrollará como una plataforma web donde todos los intervinientes en la obra podrán acceder, gestionar y consultar las obras. Asimismo, a través de ella, el coordinador de seguridad y salud elaborará los informes periódicos que verifiquen si la obra se realiza bajo las pautas de seguridad establecidas.

De esta manera se pretende unificar la información y el sistema de trabajo, facilitando la tarea de control y seguimiento de la obra, y como consecuencia mejorar la seguridad y acelerar la resolución de riesgos detectados.

Palabras clave: seguridad y salud, coordinador, contratista, obra, promotor, riesgo, protocolo, ingeniería web, Java, J2EE, Jboss, Hibernate, Struts2, Bootstrap.

Abstract

This project involves the development of a web application for the management of the documents and information of public works of the "Diputación de Barcelona – Gerencia de servicios de infraestructuras Viarias y Movilidad" (Management Service for Road Infrastructure and Mobility - City of Barcelona)

The safety of construction works is regulated according to the laws and regulations contained in the "Instituto Nacional de Seguridad e Higiene en el trabajo" (National Technical Institute of Health and Safety at Work).

This application is developed as a web platform. All the people in charge of the works can access, manage and consult the works. Also, the health and safety coordinator can make periodic reports to verify whether the works are being developed according to the established safety guidelines.

This application aims to unify the work and information systems, making easier the tasks to check and monitor the works, improving security and speeding up the response to possible risks.

Keywords: safety and health coordinator, contractor, public works, promoter, risk, protocol, web engineering, Java, J2EE, Jboss, Hibernate, Struts2, Bootstrap.

Notaciones y Convenciones

Título 1: Arial 20 negrita

Título 2: Arial 13 negrita

Título 3: Arial 11 negrita

Cuerpo: Arial 10 regular

Palabras en otros idiomas: cursiva

Código: Courier New 8 regular con sangría de 1,15 cm en los laterales, en fondo gris claro

Notaciones: Arial 10 regular en cursiva con sangría de 1,15 cm en los laterales, en fondo azul claro y alineado al centro

Índice

1. Introducción.....	12
2. Descripción	13
3. Objetivos	14
3.1 Principales	14
3.2 Secundarios.....	14
4. Alcance	15
5. Metodología	16
6. Arquitectura de la aplicación	17
6.1 Infraestructura de la aplicación	17
6.2 Equipo y software de escritorio.....	17
6.3 Servidor o maquina virtual.....	17
7. Planificación.....	19
7.1 Relación de actividades.....	19
7.2 Calendario	22
7.3 Análisis de riesgos.....	24
8. Proceso de trabajo.....	25
9. Descripción de Requisitos.....	26
10. Definición de usuarios.....	29
11. Arquitectura.....	31
12. Entorno de Desarrollo	34
13. Entorno de integración	35
14. Requisitos funcionales	36
15. Casos de Uso.....	37
16. Diseño de clases.....	40
17. Modelo de base de datos.....	43
18. Diseño de pruebas.....	44
19. Prototipos.....	46
18.1 Login.....	46
18.2 Inicio (noticias).....	46
18.3 Nueva Obra	47
18.4 Listado de obras	47
18.5 Modificación Obra.....	48
18.6 Detalle de obra	49
18.7 Visita de obra.....	50
18.8 Cierre de obra.....	52
18.9 Perfil	53
20. Configuración entorno de trabajo.....	54
Configuración de OpenShift.....	54
Configuración de MySQL.....	56
Configuración de Java.....	56
Configuración de Maven.....	58
Configuración de Tomcat.....	58
Configuración de Jboss Developer Studio.....	59
Configuración de Struts 2	61
Configuración de Hibernate	62
21. Perfiles de usuario	63
21.1 Jefe de la Diputación	63
21.2 Técnico de la Diputación.....	63
21.3 Coordinador de Seguridad y Salud.....	63
21.4 Empresa contratista.....	64
22. Bugs.....	65
23. Proyección a futuro	66
Mejoras dentro de la versión actual	66
Ampliaciones futuras	66
24. Conclusiones.....	67
Anexo 1. Entregables del proyecto	68
Proyecto completo en web	68
Memoria del proyecto	68
Código fuente	68
Defensa audiovisual	68
Presentación libre	68
Autoinforme de evaluación	68
Anexo 2. Ingeniería inversa con Hibernate	69
Anexo 3. Código fuente (extractos).....	79

Fichero de dependencias de Maven pom.xml	79
Hibernate.cfg.xml	82
Definición de acciones	83
Clase Action	84
Clases Dao	87
Mapeos de la base de datos	92
Grabación de ficheros	96
Subir imágenes	97
Anexo 4. Capturas de pantalla	98
4.1 Login	98
4.2 Inicio	99
4.3 Listado de obras	100
4.4 contacto	101
4.5 Nueva obra	102
4.6 Editar obra	103
4.7 Ver obra	104
4.8 Crear visita de obra	105
4.9 Visita de obra	106
4.10 Editar visita de obra	107
4.11 Cerrar obra	108
4.11 Perfil de usuario	109
Anexo 5. Guía de usuario	110
Acceder a la plataforma	110
Salir de la plataforma	110
Modificar datos de usuario	110
Actualizaciones de obra	111
Listado de obras	112
Detalle de obra	112
Visita de obra	113
Contacto con la plataforma	113
Crear obra (sólo Técnicos)	113
Editar obra (sólo Coordinador de Seguridad y Salud)	114
Añadir documentación a la obra (sólo Coordinador de Seguridad y Salud)	115
Crear visita de control (sólo Coordinador de Seguridad y Salud)	115
Editar visita de control (sólo Coordinador de Seguridad y Salud)	115
Añadir fotos a la visita de control (sólo Coordinador de Seguridad y Salud)	116
Borrar Visita de control (sólo Coordinador de Seguridad y Salud)	116
Enviar visita de control (sólo Coordinador de Seguridad y Salud)	117
Cerrar Obra (sólo Coordinador de Seguridad y Salud)	117
Anexo 6. Glosario	118
Anexo 7. Bibliografía	120

Figuras y tablas

Índice de figuras

Figura 1: Diagrama de Gantt.....	23
Figura 2: Arquitectura en tres capas	31
Figura 3: Arquitectura Struts	32
Figura 4: Arquitectura Hibernate	33
Figura 5: Diagrama de clases Usuario	40
Figura 6: Diagrama de secuencia para el caso de uso Login	41
Figura 7: Modelo de BBDD	44
Figura 8 : Prototipo de la pantalla de Login.....	46
Figura 9 : Prototipo de la pantalla de Noticias.....	46
Figura 10: Prototipo de la pantalla de Nueva Obra	47
Figura 11: Prototipo de la pantalla de Listado de Obras	47
Figura 12: Prototipo de la pantalla de modificación obra	48
Figura 13: Prototipo de la pantalla de Detalle de obra	49
Figura 14: Prototipo de la pantalla de Visita de obra - Equipamiento de protección individual	50
Figura 15: Prototipo de la pantalla de Visita de obra - Equipamiento de protección colectiva	50
Figura 16: Prototipo de la pantalla de Visita de obra - Señalización de obra	51
Figura 17: Prototipo de la pantalla de Visita de obra - Informe de visita y reportaje fotográfico.....	51
Figura 18: Prototipo de la pantalla de Cierre de obra.....	52
Figura 19: Prototipo de la pantalla de Perfil	53
Figura 20: OpenShift	55
Figura 21: Add Application Open Shift	55
Figura 22: Add MySQL 5.5.....	56
Figura 23: Variables de entorno	57
Figura 24: Insertar variable de entorno JAVA_HOME.....	57
Figura 25: Maven version.....	58
Figura 26: Añadir servidor Tomcat.....	59
Figura 27: Hibernate, ingeniería inversa	69
Figura 28: Alta configuración Hibernate.	70
Figura 29: Configuración Hibernate edición	70
Figura 30: Seleccionar base de datos para Hibernate	71
Figura 31: Driver base de datos de Hibernate.	71
Figura 32: Eliminar driver por defecto de Hibernate.....	72
Figura 33: Ruta fichero properties de Hibernate	74
Figura 34: Fichero configuración Hibernate	75
Figura 35: Crear fichero configuración Hibernate (cfg.xml).....	75
Figura 36: Coger los datos de la base de datos.....	76
Figura 37: Generación de código a través de Hibernate.....	76
Figura 38: Configuración creacion de Clases a traves de Hibernate.....	77
Figura 39: Ejecutar la generación de clases a través de Hibernate	78
Figura 40: Captura PC. Login	98
Figura 41: Captura Móvil. Login	98

Figura 42: Captura PC. Inicio	99
Figura 43: Captura Móvil. Inicio.....	99
Figura 44: Captura PC. Listado.....	100
Figura 45: Captura Móvil. Listado	100
Figura 46: Captura PC. Contacto	101
Figura 47: Captura Móvil. Contacto.....	101
Figura 48: Captura PC. Nueva Obra	102
Figura 49: Captura Móvil. Nueva Obra.....	102
Figura 50: Captura PC. Editar Obra	103
Figura 51: Captura Móvil. Editar Obra.....	103
Figura 52: Captura PC. Ver Obra.....	104
Figura 53: Captura Móvil. Ver Obra	104
Figura 54: Captura PC. Crear Visita de Obra.....	105
Figura 55: Captura Móvil. Crear Visita de Obra	105
Figura 56: Captura PC. Visita de obra	106
Figura 57: Captura Móvil. Visita de obra	106
Figura 58: Captura PC. Editar Visita de Obra	107
Figura 59: Captura Móvil. Editar Visita de Obra	107
Figura 60: Captura PC. Cerrar Obra	108
Figura 61: Captura Móvil. Cierre de Obra	108
Figura 62: Captura PC. Perfil de usuario	109
Figura 63: Captura Móvil. Perfil de usuario	109
Figura 64: Guía de usuario - Acceso a la plataforma.....	110
Figura 65: Guía de usuario - Salir de la plataforma.....	110
Figura 66: Guía de usuario – Acceso a Perfil.....	110
Figura 67: Guía de usuario - Modificar perfil	111
Figura 68: Guía de usuario - Últimas actualizaciones	111
Figura 69: Guía de usuario - Listado de obras	112
Figura 70: Guía de usuario - Detalle de obra	112
Figura 71: Guía de usuario - Ver visita de obra	113
Figura 72: Guía de usuario - Contacto	113
Figura 73: Guía de usuario - Crear nueva obra	114
Figura 74: Guía de usuario - Editar obra.....	114
Figura 75: Guía de usuario - Añadir documentación.....	115
Figura 76: Guía de usuario - Crear visita de control.....	115
Figura 77: Guía de usuario - Editar visita de control	116
Figura 78: Guía de usuario - Añadir fotografías a visita de control	116
Figura 79: Guía de usuario - Borrar visita de control.....	116
Figura 80: Guía de usuario - Enviar visita de control	117
Figura 81: Guía de usuario - Cerrar obra	117

Índice de tablas

Tabla 1: Relación casos de uso y actores.....	39
Tabla 2: Diseño de pruebas	45

1. Introducción

El protocolo de seguridad y salud es una plataforma web que controla, gestiona y fomenta la participación de todos los técnicos y personal involucrado en proyectos y obras de carácter civil o en materia de seguridad y salud, consiste en un sistema que enlaza los diferentes perfiles de un proyecto u obra: coordinador, contratista y promotora.

El objetivo principal es responder de forma adecuada a cuatro pilares fundamentales del sistema de seguridad y salud: Organización, Cumplimiento, Idoneidad y Seguridad.

Organización de la gestión técnica y documental

La gestión técnica y documental antes, durante y en la finalización de los trabajos de Seguridad y Salud queda registrada de forma que los documentos pueden ser consultados de forma continua y con un sistema organizativo definido y único, indiferente de la cantidad de personas que utilicen el programa.

Cumplimiento de la normativa vigente

El sistema está desarrollado en base a los trámites definidos por la normativa vigente, la gestión de la documentación y de las visitas técnicas a pie de obra, queda codificada y guardada para consultas posteriores.

Idoneidad del sistema ante acciones judiciales o situaciones adversas

Las leyes en materia de Seguridad y Salud son varias y diversas. Todas las partes implicadas, tanto contratistas, coordinadores y promotores están obligadas a cumplirlas.

La responsabilidad es subsidiaria, con lo cual es importante tener un soporte objetivo y completo para poder afrontar posibles situaciones que exigen un alto grado de organización y gestión, en este aspecto, el Protocolo de Seguridad y Salud garantiza la correcta gestión del proyecto y obra consiguiendo una política de transparencia y profesionalidad factores clave para el éxito en todos los procesos.

Seguridad y Salud aplicadas en las fases de redacción y ejecución de proyectos y obras

La Seguridad y Salud no se resume en una gestión documental, el primordial objetivo es la reducción y eliminación de los diferentes riesgos presentes en ámbito de las obras y la previsión de medidas correctoras. En este aspecto el protocolo de Seguridad y Salud marca unas pautas de trabajo y un control en tiempo real de todas las medidas a aplicar en la obra, así aseguramos la aplicación correcta de las mismas.

2. Descripción

El protocolo para las coordinaciones de seguridad y salud de las obras de la Diputación de Barcelona - Gerencia de servicios de infraestructuras Viarias y Movilidad establece las directivas básicas para coordinar las tareas de seguridad en la fase de ejecución entre la Diputación, como empresa promotora, la empresa contratista y la coordinadora de seguridad y salud en fase de obra.

Este protocolo está basado y elaborado según los Reales Decretos, Leyes y Normativas Técnicas recogidas en el Instituto Nacional de Seguridad e Higiene en el trabajo y los criterios de los profesionales implicados en las tareas de seguridad y salud.

Las tareas que engloban todo el proceso de coordinación de seguridad y salud hasta el cierre técnico de la obra son:

1. Designación del Coordinador de Seguridad y Salud en fase de obra.
2. Entrega del Plan de Seguridad y Salud.
3. Entrega de una copia del proyecto constructivo.
4. Ratificación y aprobación del Plan de Seguridad y Salud.
5. Notificación Aviso Previo.
6. Solicitud de documentación previa al inicio de las obras.
7. Visita de control antes del inicio de las obras al ámbito del proyecto.
8. Coordinación de las visitas con la Dirección Facultativa, contratista e implicados.
9. Visitas periódicas
10. Cierre de obra.

Este proyecto consiste en una aplicación web donde se gestionará la información y documentación de las obras y desde la que se llevará a cabo las visitas periódicas, elaborando con ellas un informe con un reportaje fotográfico y las medidas adoptadas y pendientes para la próxima visita. Se notificará al contratista del resultado de la visita, el cual tendrá que resolver las incidencias encontradas en el menor tiempo posible. Además, desde la aplicación se llevará a cabo el cierre de la obra con un breve informe resumen de lo más destacado del periodo de la obra.

La aplicación web, gestionará las dos últimas etapas del proceso de coordinación de seguridad y salud, visitas periódicas y cierre de obra, y recogerá la información y documentación que se genera en las etapas anteriores.

3. Objetivos

3.1 Principales

- Crear un sistema único de trabajo donde tener agrupada toda la información de la obra.
- Facilitar la tarea de seguimiento y control de la obra al realizar el proceso online, en lugar de rellenar informes y enviarlos por correo.
- Establecer un formato único de informe de visita que facilite su elaboración e interpretación.
- Permitir la elaboración de los informes a pie de obra mediante un diseño adaptable a dispositivos móviles.
- Hacer partícipe a los intervinientes en el proceso de la mejora de la aplicación.

3.2 Secundarios

- Saber establecer y definir las dimensiones principales de un proyecto web.
- Entender el ciclo de vida y las etapas principales de un proyecto de desarrollo web.
- Progresar en el uso de las herramientas necesarias para la realización de la aplicación web.

4. Alcance

El presente proyecto incluye:

- Toma de requisitos y definición de casos de uso para cubrir las siguiente funcionalidades:
- Acceso a la plataforma
- Listado de Obras.
- Detalle de una obra específica.
- Alta de obra.
- Introducción de datos generales de la obra.
- Introducción de documentación de la obra.
- Elaboración de informes periódicos de visita, donde se recogerán los siguientes datos:
 - Equipamiento de protección individual.
 - Equipamiento de protección colectiva.
 - Señalización de obra.
 - Informe de la visita.
- Investigación y selección de la tecnología más apropiada.
- Diseño de interfaces gráficos
- Diseño de Modelo de datos
- Diseño de la aplicación
- Desarrollo y pruebas de la aplicación
- Integración y despliegue en un servidor de pruebas accesible de forma online
- Documentación final y presentación virtual

No se incluye:

- Puesta en producción real
- Compra de dominio
- Compra de certificados para protocolo https
- Servicio de alojamiento web o data-center.

5. Metodología

La metodología seleccionada para llevar a cabo el proyecto es la de **modelo en cascada**, es un proceso secuencial de las fases genéricas de desarrollo de software, que permite cierta superposición y flujo entre fases.

El tiempo invertido en la especificación y diseño dará más posibilidades al proyecto para alcanzar sus objetivos, así como, evitar posibles desvíos de planificaciones realizadas. Su estrecha rigidez, la hace adecuada en desarrollos estables sin cambios de requerimientos como es el caso del proyecto. Tanto la definición de proyecto, como su alcance, son específicos, puede haber pequeñas desviaciones, que causen alguna rectificación, pero no será la norma.

Las fases principales son: Propuesta, análisis y diseño, implementación, pruebas, memoria y debate virtual. Cada una de las fases es comentada más profundamente en la planificación.

6. Arquitectura de la aplicación

6.1 Infraestructura de la aplicación

La aplicación web, utilizará distintos componentes de software, tanto para su ejecución como para su desarrollo. Se debe disponer de los siguientes elementos:

- **Servidor de aplicaciones.** Se encargará de alojar y ejecutar el código de nuestra aplicación. El lenguaje de programación seleccionado es Java por lo tanto se utilizará *Tomcat* o *JBoss*, La decisión final se tomará en la etapa de análisis y diseño.
- **Sistema gestor de base de datos.** Se encarga de la persistencia de los datos que manejará la aplicación. Se utilizará *MySQL*.
- **Sistema de control de versiones.** En cualquier proyecto es conveniente utilizar un software que guarde las distintas versiones de los ficheros que se modifican. Existen soluciones como CVS, SVN o Git.
- **Lenguaje de programación.** Se utilizará Java en el lado del servidor y Javascript, Html y Css en el lado del cliente.
- **Frameworks y librerías de desarrollo.** Los frameworks de desarrollo y las librerías existentes pueden facilitar el desarrollo de una aplicación. Se utilizará *Struts* para Java, *jQuery* para Javascript y *Bootstrap* para las hojas de estilos CSS.

6.2 Equipo y software de escritorio.

Para realizar todo el trabajo será se utilizará al menos un equipo de escritorio conectado a Internet. Sobre él se instalará el software de la infraestructura indicado en el punto anterior. Adicionalmente será necesario el siguiente software:

- **Software de ofimática:** *MS office* o *LibreOffice*.
- **Software la para la planificación del proyecto:** *MS Project* o *ProjectLibre*.
- **Entornos de desarrollo:** Es la herramienta que se utilizará para escribir el código de nuestra aplicación. Se utilizará eclipse para trabajar con Java y *Toad for MySQL* para la base de datos.
- **Navegadores web:** Conviene tener varios instalados para probar la compatibilidad de nuestro software con todos ellos. Ejemplos: *MS Internet Explorer*, *Chrome*, *Firefox*, *Safari*, *Opera*.

6.3 Servidor o maquina virtual.

Toda la infraestructura de la aplicación puede instalarse en la máquina local del desarrollador, sin embargo, es conveniente disponer de un servidor o máquina virtual adicional donde se instala la aplicación con la finalidad de realizar pruebas sobre una versión más estable o integrar todas las partes

de la aplicación. Además puede servir para mostrar la aplicación a terceros. Se utilizará un plan gratuito basado en Linux, OpenShift de la empresa Red Hat.

No se contempla la puesta en producción real. Por lo tanto no se considera la compra de un dominio, certificados de seguridad ni contratación de hosting.

7. Planificación

7.1 Relación de actividades

A continuación se enumeran las tareas que incluye el proyecto, estas se distribuyen en las diferentes etapas del proyecto que se corresponden a su vez con las pruebas de evaluación continua.

7.1.1 PAC1 – Inicio

7.1.1.1 Propuesta

Del 17-09-2014 al 19-09-2014.

Elaboración de un breve resumen del proyecto explicando en que va consistir, cuál es su ámbito y las herramientas que se van a utilizar para desarrollarlo.

7.1.1.2 Elaboración de plan de trabajo

Del 20-09-2014 al 26-09-2014.

Planificación del proyecto y elaboración del documento “Plan de trabajo”.

7.1.1.3 Revisión

Del 27-09-2014 al 28-09-2014.

Entrega parcial para una valoración previa por parte del consultor y aplicación de las correcciones necesarias.

7.1.1.4 Entrega plan de trabajo

30-09-2014.

Entrega definitiva del documento “Plan de trabajo”.

7.1.2 PAC2 – Análisis y diseño

7.1.2.1 Definición de requisitos

Del 01-10-2014 al 02-10-2014.

Definición de las características que debe cumplir la aplicación web.

7.1.2.2 Diseño de casos de uso

Del 03-10-2014 al 04-10-2014.

Especificación del comportamiento de la aplicación web frente a una interacción con el usuario.

- Descripción de casos de uso.
- Esquema de casos de uso.

7.1.2.3 Investigación y selección de tecnología

Del 05-10-2014 al 06-10-2014.

Estudio y elección de la tecnología necesaria y adecuada para desarrollar el proyecto.

7.1.2.4 Diseño de interfaces

Del 07-10-2014 al 11-10-2014.

Definición de la apariencia de las pantallas y elaboración de wireframes o prototipos.

- Elaboración wireframes.
- Prototipos reales en HTML y CSS.
- Guía de estilo.

7.1.2.5 Diseño de modelo de datos

Del 12-10-2014 al 14-10-2014.

Elaboración el esquema que represente la estructura de la base de datos y las relaciones entre sus objetos, teniendo en cuenta el tipo de DBMS seleccionado.

- Esquema conceptual.
- Diseño lógico.
- Diseño físico.(Prioridad baja)

7.1.2.6 Diseño de aplicación

Del 15-10-2014 al 17-10-2014.

- Diseño de clases
- Análisis de patrones

7.1.2.7 Diseño de pruebas

Del 18-10-2014 al 19-10-2014.

Definición de las pruebas que se realizarán para garantizar el correcto funcionamiento de la aplicación, teniendo en cuenta las funcionalidades más críticas y los requerimientos solicitados.

7.1.2.8 Revisión

Del 20-10-2014 al 24-10-2014.

Entrega parcial para una valoración previa por parte del consultor y aplicación de las correcciones necesarias.

7.1.2.9 Entrega análisis de diseño

26-10-2014.

Entrega definitiva de la documentación correspondiente al análisis y diseño.

7.1.3 PAC3 – Implementación

7.1.3.1 Configuración (BBDD, servidor, entorno...)

Del 27-10-2014 al 30-10-2014.

Preparación y configuración del entorno de desarrollo.

7.1.3.2 Desarrollo BBDD

Del 31-10-2014 al 03-11-2014.

Implementación de la base de datos según el modelo de datos desarrollado en la etapa anterior de análisis y diseño.

7.1.3.3 Desarrollo y pruebas de la aplicación

Del 04-11-2014 al 19-11-2014.

Elaboración de la versión beta del proyecto siguiendo los diseños elaborados en la etapa anterior.

- Desarrollo Java.
- Pruebas Junit (prioridad baja).
- Pantallas en HTML,CSS

7.1.3.4 Documentación

Del 20-11-2014 al 23-11-2014.

Elaboración de la documentación correspondiente a todas las tareas realizadas en esta etapa de implementación.

- Documentación PAC3.
- Manual de usuario. (Prioridad baja)

7.1.3.5 Revisión

Del 24-11-2014 al 27-11-2014.

Entrega parcial para una valoración previa por parte del consultor y aplicación de las correcciones necesarias.

7.1.3.6 Entrega versión beta.

30-11-2014.

Entrega de una versión beta del proyecto junto con la documentación correspondiente.

7.1.4 PAC4 – Entrega final

7.1.4.1 Depuración y mejoras

Del 01-12-2014 al 22-12-2014.

Corrección de los errores encontrados en la versión beta y aplicación de mejoras necesarias para ofrecer una versión definitiva.

7.1.4.2 Memoria

Del 01-12-2014 al 04-01-2014.

Finalización del documento de memoria que se ha ido elaborando a lo largo del proyecto.

7.1.4.3 Presentación virtual

Del 31-12-2014 al 04-01-2015.

Elaboración de una presentación virtual que muestre el trabajo realizado de forma sintética y atractiva.

7.1.4.4 Revisión

Del 04-01-2015 al 09-01-2015.

Entrega parcial de la memoria para una valoración previa por parte del consultor y aplicación de las correcciones necesarias.

7.1.4.5 Entrega final

13-01-2015.

Entrega definitiva de la memoria, presentación virtual y los entregables correspondientes.

7.1.5 PAC5 - Debate virtual

7.1.5.1 Debate virtual

Del 24-11-2014 al 27-11-2014.

Responder a las preguntas elaboradas por el Tribunal de evaluación.

7.2 Calendario

La Figura 1 muestra el diagrama de Gantt donde se refleja la distribución de las tareas dentro del calendario.

Los días asignados son días laborables, eliminando fines de semana y días festivos del calendario de trabajo. Cada día contabiliza ocho horas de trabajo realizadas en jornada partida, dejando los fines de semana libres pero disponibles en caso de retrasos.

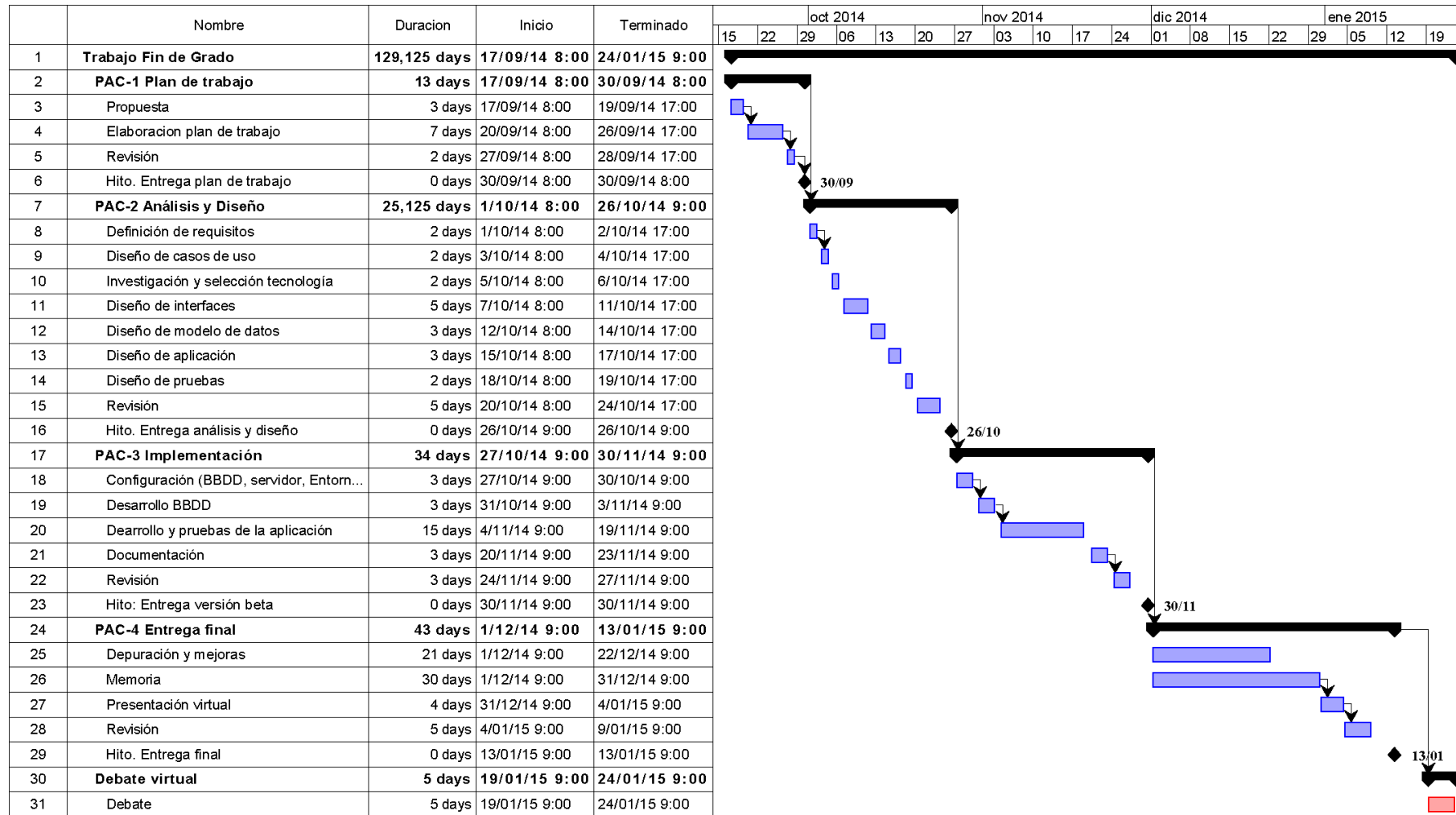


Figura 1: Diagrama de Gantt.

7.3 Análisis de riesgos

A continuación se enumeran los posibles riesgos que pueden afectar a este tipo de proyecto:

- R1-Falta de tiempo para el desarrollo debido a algún imprevisto como puede ser enfermedad, asuntos personales, o problemas en el trabajo
- R2-Falta de tiempo para el desarrollo por planificación incorrecta o demasiado optimista.
- R3-Falta de conocimientos y experiencia suficientes para llevar a cabo el proyecto en el tiempo estimado.
- R4-Averías en el hardware utilizado

Para minimizar los posibles efectos de los riesgos anteriores se tomarán las siguientes medidas

- Se planificará el final de cada fase dos o tres días antes de la fecha de entrega o hito. Además, para cada fase se tendrá en cuenta que la última tarea debe ser la revisión de todo el trabajo de cada fase, donde se corregirán errores o se finalizará alguna tarea que quedase pendiente (R1, R2)
- Se clasificarán las tareas por orden de prioridad, clasificando como de baja prioridad, aquellas que pueden ser prescindibles y no son estrictamente necesarias para completar el proyecto. Pudiendo no realizarse si no se dispone del tiempo suficiente (R1, R2)
- En la segunda fase del proyecto (diseño y análisis) se intentarán escoger aquellas tecnologías de las que se dispone de más conocimientos y experiencia para agilizar el desarrollo (R1, R2, R3)
- Se dispondrá de un segundo equipo donde pueda replicarse el entorno de desarrollo (R4)
- Se utilizará un control de versiones online que nos sirva de Backup del código (R4)

8. Proceso de trabajo

El proceso de trabajo, acorde con la metodología de modelo en cascada, que se seguirá para la realización de la aplicación web es la siguiente:

1. Conocer las primeras necesidades y expectativas de los usuarios.
2. Elegir una arquitectura adecuada para el desarrollo.
3. Definir el entorno de desarrollo.
4. Definir los requisitos funcionales
5. Definir los Actores.
6. Definir los casos de uso.
7. Realizar el diseño de clases
8. Diseño del modelo de BBDD.
9. Diseño de pruebas.
10. Realizar el diseño de pantallas.
11. Inicio del desarrollo en función de los análisis y diseños previos. En esta fase el diseño definitivo puede variar ligeramente.
12. Realización de pruebas y corrección de errores.
13. Elaboración del manual de usuario.
14. Entrega y publicación.

9. Descripción de Requisitos

La Diputación de Barcelona – Gerencia de servicios de infraestructuras Viarias y Movilidad necesita gestionar la información y documentación de las obras viales que tienen lugar dentro de su territorio, así como realizar el seguimiento y control para garantizar las condiciones de seguridad según las leyes y normativas recogidas en el Instituto Nacional de Seguridad e Higiene en el trabajo.

Obra, datos generales

Las obras serán dadas de alta por los técnicos de la Diputación, los cuales únicamente crearán el registro de la obra, código y título, se la adjudicarán a un coordinador de obra que será encargado de introducir el resto de datos y seleccionarán al contratista encargado de la realización de la obra.

La obra se identificará por un código y guardará los datos: título, carretera, punto kilométrico, fecha de inicio de la obra, presupuesto y la duración en meses, se plantea la posibilidad de guardar las coordenadas para localizar la obra en Google maps. También se guardará los datos de los implicados en la obra, siendo los siguientes:

- **Técnico de la Diputación:** creará la obra y de él se guardará el nombre completo, email, teléfono de la diputación junto con la extensión y el teléfono móvil.
- **Coordinador de obra:** se encargará del resto de datos de la obra y de las visitas de control se guardará el Nif, nombre completo, email, teléfono, empresa a la que pertenece, Cif de la empresa, dirección de la empresa y teléfono de la empresa.
- **Empresa contratista:** es la empresa que realiza la obra, los datos a guardar son Cif, nombre, dirección, teléfono, fax, email. Dentro de la empresa contratista existirá una persona que ejerce el cargo de **jefe de la obra**, de esta persona se guardará su nif, nombre completo, teléfono móvil y email

Documentación

De la obra se guardará la **documentación obligatoria** para su ejecución, estos documentos son: Aviso Previo, TC2 de los Trabajadores, Entrega de EPIs (Equipos de protección individual), Aprobación del plan de seguridad, Apertura del centro de trabajo, Libro de incidencias, Libro de subcontrataciones, Plan de Obra, ITV de los Vehículos y Hoja de Hospitales y Emergencias, también se permitirá añadir algún documento adicional que pueda ser relevante.

Estos documentos se guardarán con un nombre que los identifique y que identifique la obra a la que pertenecen.

Visitas de control

Mientras la obra esté en ejecución el coordinador de obra, realizará **visitas** para comprobar el estado de la misma y el seguimiento de las medidas de seguridad correspondientes. De la visita se guardarán los siguientes datos:

- **Equipamiento de protección individual:** se guardará que elementos de protección individual se están utilizando dentro de los siguientes: Casco, Pantalla facial, Arnés, Chaleco reflectante, Gafas, Botas, Orejeras, Guantes, Máscara y Gorra/similar.
- **Equipamiento de protección colectiva:** se guardará que elementos de protección colectiva se están utilizando dentro de los siguientes: Botiquín, Barandilla, Redes de seguridad, Líneas de vida, Extintores CO2, Extintores Agua y Extintores Polvo.
- **Señalización de obra:** se guardará que elementos de señalización se están utilizando dentro de los siguientes: Señalización Fija, Señalización Móvil, Conos, Semáforo provisional, Banda rugosa, Señaladores y Extintores Polvo. Se especificará cual de las siguientes señales se están utilizando: TP3, TP15, TP15a, TP15b, TP17, TP17a, TP17b, TP18, TP19, TP25, TP28, TP30, TP50, TR301, TR302, TR303, TR305. También se guardará si la señalización utilizada es en una dirección de la carretera o en dos.

Por cada uno de estos grupos de medidas de seguridad, se permitirá añadir un par de fotos y escribir un texto de observaciones y se determinará un nivel en el estado de cada uno de ellos, dentro del rango de valores: Muy bueno, Bueno, Básico e Insuficiente.

La visita guardará, además, un texto lo suficientemente extenso donde el coordinador de obra realizará un informe general de la visita. Además, la visita de obra, contendrá un reportaje fotográfico de unas seis fotografías donde se muestre de manera visual el estado de las condiciones de seguridad.

Una vez finalizado el informe, este se guardara y se enviará al técnico de la obra y al contratista, en caso de que el estado de seguridad no sea el adecuado el contratista tendrá que aplicar los cambios oportunos para que el nivel de seguridad sea el adecuado.

Cierre de obra

Cuando las tareas de obra se hayan finalizado por completo, el coordinador de seguridad y salud escribirá un informe final para resumir los puntos más importantes y relevantes durante la ejecución de las obras, se guardará la fecha de finalización y se enviará al técnico de la diputación y al contratista.

Listado de las obras

La aplicación mostrará un listado de las obras con los datos principales de la misma, y dentro del cual se pueda seleccionar una obra en concreto y acceder a los informes de visita de la misma. Por defecto las obras mostradas serán las obras en ejecución, pero se añadirá un filtro para poder visualizar obras ya cerradas y buscar a través de campos clave como son el código y el título.

Los usuarios de la aplicación solo verán las obras en las cuales estén involucrados.

El diseño de las pantallas tendrá que estar adaptado a dispositivos móviles, en especial el de la pantalla de informe de visita, para que el coordinador de obra lo pueda completar en el mismo

momento en el que realice la visita, pudiendo enviarlo directamente o guardarlo para un envío posterior.

10. Definición de usuarios

A continuación se analizan los distintos usuarios que utilizarán la aplicación, sus características y necesidades, en base a las cuales se tomarán las decisiones de diseño.

Jefe de la Diputación

- **Definición:** Son los responsables de coordinar todas las obras.
- **Tareas:** Controlan las tareas llevadas a cabo por los Técnicos de la Diputación.
- **Funciones que necesitan:** Listado de las obras, ver información detallada de la obra, acceder a la documentación de la obra, acceder a las visitas de la obra
- **Nivel académico:** Alto
- **Nivel de experiencia:** Alto, utilizan internet a diario, correo electrónico y aplicaciones específicas de gestión.

Técnico de la Diputación

- **Definición:** Son los responsables de coordinar el trabajo técnico de la obras del servicio de infraestructuras viales y de movilidad, representan a la promotora, Diputación de Barcelona, para las obras a las que han sido asignados.
- **Tareas:** Crean la obra y asignan al coordinador y al contratista.
- **Funciones que necesitan:** Crear obra, listado de obras, ver información detallada de la obra, acceder a la documentación de la obra, accedes a las visitas de la obra, recibir la información de la obra cuando el coordinador realice cambios o visitas.
- **Nivel académico:** Alto.
- **Nivel de experiencia:** Alto, utilizan internet a diario, correo electrónico y aplicaciones específicas de gestión.

Coordinador de seguridad y salud

- **Definición:** Son los responsables de coordinar los aspectos de seguridad y salud para las obras.
- **Tareas:** Coordinar los aspectos de seguridad y salud por cada obra asignada, verificar la información y documentación de la obra y subirla a la plataforma web. Realizar las visitas de control y sus correspondientes informes y elaborar el informe de cierre de obra.
- **Funciones que necesitan:** Editar obra, listado de obras, ver información detallada de la obra, acceder a la documentación de la obra, crear el informe de visita de obra, enviar sus informes y actualizaciones de obra a los involucrados.
- **Nivel académico:** Medio - Alto.
- **Nivel de experiencia:** Alto, utilizan internet a diario, correo electrónico y aplicaciones específicas de gestión.

Contratista

- **Definición:** Es la empresa seleccionada por la diputación para llevar a término la ejecución de las obras.
- **Tareas:** Enviar la información correspondiente al coordinador de obra y cumplir con la normativa de seguridad y salud. Estar en comunicación con el coordinador de seguridad y salud y rectificar los sistemas de seguridad cuando el coordinador se lo indique. El encargado en realizar estas tareas es el Jefe de Obra.
- **Funciones que necesitan:** visualizar la obra, visualizar los informes de obra, recibir la información de los informes.
- **Nivel académico:** Medio - Bajo.
- **Nivel de experiencia:** Bajo, realiza tareas básicas como enviar emails y visualizar información, no utiliza aplicaciones de gestión.

11. Arquitectura

Se trata de una aplicación web con arquitectura clásica en tres capas o niveles: nivel de cliente, nivel de servidor de aplicaciones y nivel de bases de datos

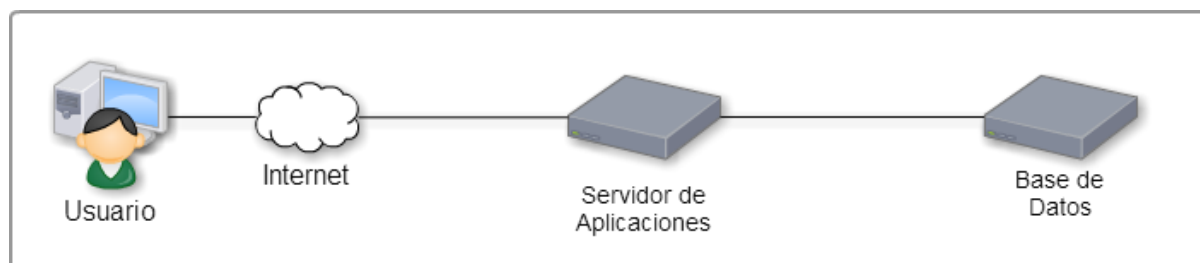


Figura 2: Arquitectura en tres capas¹

Nivel de cliente. El usuario interactúa con la aplicación a través de cualquier navegador web desde donde se envían peticiones al servidor y se muestran la respuesta del mismo. Esta capa se implementa en HTML5², Javascript³, CSS y utiliza las librerías y frameworks de Bootstrap⁴ y JQuery⁵.

Nivel Servidor de aplicaciones. Este es el nivel encargado de ejecutar la aplicación en el servidor. Contiene la lógica de negocio de la aplicación. Existe un servidor de aplicaciones: Tomcat⁶ que ejecuta código escrito en lenguaje java. Se utilizará el *framework* Struts2 junto con Hibernate para el desarrollo de la aplicación. Struts e Hibernate son dos *frameworks* maduros, ampliamente adoptados y de código libre, existiendo una gran comunidad de desarrolladores que pueden proporcionar soporte a través de wikis, foros, páginas de preguntas y respuestas etc.

Struts⁷ es un *framework* desarrollado en Java⁸, perteneciente a la *Apache Software Foundation*. Nos provee de la estructura de la aplicación basada en la arquitectura MVC: modelo-vista-controlador. Nos va a permitir definir unas acciones en un fichero de configuración y cada acción asocia una URL a una clase java con un método que se encarga de procesarla. Además, en cada acción se define las páginas que puede tener como resultado. Existe por lo tanto un 'controlador' que lee el fichero de

¹ Arquitectura en tres capas. Gráfico generado con Cacao: <https://cacao.com>

² Html <http://www.w3.org/html/>

³ JavaScript <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

⁴ Bootstrap <http://getbootstrap.com/>

⁵ JQuery <http://jquery.com/>

⁶ Tomcat: <http://tomcat.apache.org/>

⁷ Struts: <http://struts.apache.org/>

⁸ Java <http://www.java.com>

configuración y para cada acción decide la clase java que se debe ejecutar. Esta clase java representa el 'modelo' y tiene por respuesta y decide la página se va a mostrar. Las páginas se implementan en ficheros JSP y representan la 'vista'. de nuestra arquitectura.

En la siguiente imagen, se puede ver la arquitectura de Struts de forma más pormenorizada.

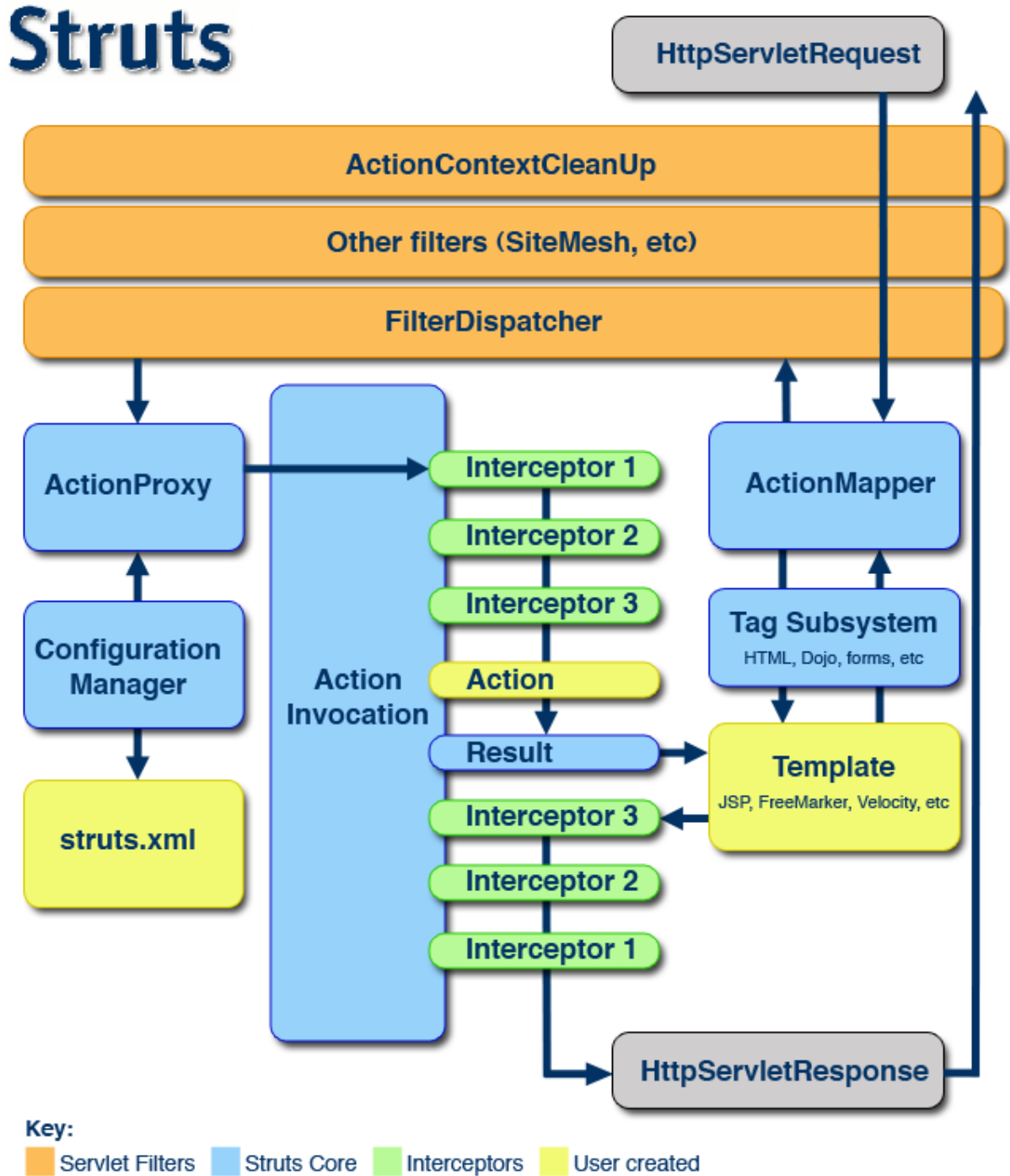


Figura 3: Arquitectura Struts⁹

⁹ Arquitectura Struts <http://struts.apache.org/release/2.0.x/docs/architecture.html>

Hibernate¹⁰ es un *framework* ORM perteneciente a la empresa Red Hat. Nos facilita el acceso a bases de datos, traduciendo el modelo entidad-relación de la base de datos a un modelo orientado a objetos. Básicamente traduce las tablas de nuestro modelo de datos a clases de un modelo orientado a objetos escritas en java. Los accesos a la BBDD los realizaremos a través de estos objetos, sin tener que realizar programación a nivel de BBDD.

La siguiente imagen, nos da una visión de alto nivel de la arquitectura de Hibernate

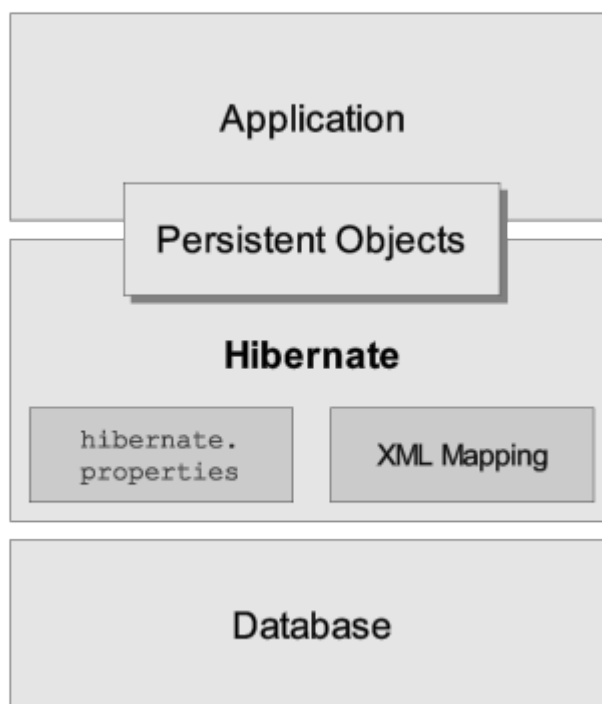


Figura 4: Arquitectura Hibernate¹¹

La utilización de estos dos *frameworks* simultáneamente nos permite simplificar el desarrollo de forma que el programador concentra sus esfuerzos en el desarrollo de la lógica de negocio de la aplicación, sin dedicar tiempo al desarrollo de la arquitectura.

Nivel de bases de datos. Como sistema gestor de base de datos se utiliza MySQL¹². Es un sistema gestor de bases de datos relacional que puede usarse con licencia de software libre. Es uno de los más utilizados a nivel mundial

¹⁰ Hibernate <http://hibernate.org/>

¹¹ Arquitectura Hibernate https://docs.jboss.org/hibernate/orm/3.2/reference/en/html_single

¹² MySQL: <http://www.mysql.com/>

12. Entorno de Desarrollo

Se plantea la utilización de un conjunto de herramientas, todas ellas gratuitas, para el desarrollo del proyecto.

Para el desarrollo y mantenimiento de la aplicación java se utiliza **Jboss Studio Developer**¹³ como entorno de desarrollo. Este IDE, está basado en Eclipse¹⁴, incluye un plugin para Git, herramientas para Hibernate, plugin para Maven y además se integra con la plataforma OpenShift.

Git¹⁵ es un sistema de control de versiones distribuido. Como servidor de Git se ha escogido la plataforma OpenShift. Tanto Git como OpenShift están integrados con el JBossStudioDeveloper

Maven¹⁶ es una herramienta ampliamente usada en proyectos java que nos permite automatizar ciertas tareas como la compilación, resolución de dependencias, generación de documentación, ejecución de pruebas etc. En este proyecto, se va a usar principalmente para compilar y resolver las dependencias de todas las librerías usadas.

Para el desarrollo de la base de datos MySQL disponemos de numerosas herramientas gratuitas. Se pueden usar indistintamente: **MySQL WorkBench**¹⁷, **Toad for MySql**¹⁸ o **phpMyAdmin**¹⁹

Notepad++²⁰: es un editor de textos muy potente, de código libre y diseñado para programadores

¹³ Jboss Studio Developer: <http://www.jboss.org/products/devstudio/overview/>

¹⁴ Eclipse: <https://www.eclipse.org/>

¹⁵ Git: <http://git-scm.com/>

¹⁶ Maven: <http://maven.apache.org/>

¹⁷ WorkBench <http://www.mysql.com/products/workbench/>

¹⁸ Toad for MySQL <http://www.quest.com/toad-for-mysql/>

¹⁹ PHPMyAdmin http://www.phpmyadmin.net/home_page/index.php

²⁰ Notepad++: <http://notepad-plus-plus.org/>

13. Entorno de integración

OpenShift²¹ es una plataforma de desarrollo en la nube desarrollada por la empresa Red Hat. Permite disponer de una cuenta en un servidor online, donde desplegar y ejecutar aplicaciones basadas en software libre. En nuestro caso vamos a utilizar el servidor Tomcat, la base de datos MySQL y el control de versiones GIT, todas estas herramientas, al ser de código libre, están incorporadas en OpenShift.

La plataforma se integra con el entorno de desarrollo Jboss Developer Studio, de forma que al incorporar cambios a Git desde nuestro entorno de desarrollo, estos se despliegan de forma automática en el servidor de aplicaciones y quedan accesibles de forma online para cualquiera con conexión a internet. Disponemos así de un control de versiones en la nube que nos sirve de respaldo de nuestra aplicación y de un servidor de integración accesible desde internet. La plataforma ofrece cuentas gratuitas como distintos planes de precios para adaptarse a las necesidades del cliente.

Git es un sistema de control de versiones distribuido.

Control de versiones Git (El servidor lo suministra OpenShift y el cliente está integrado como plugin en el entorno JBoss Studio Developer)

Gestión de BBDD MySql (phpMyAdmin + mySqlWorkBench + Toad for MySQL)

²¹ OpenShift <https://www.openshift.com/>

14. Requisitos funcionales

RQ-1 Sólo se permitirá el acceso mediante usuario/contraseña.

RQ-2 El sistema distinguirá cuatro tipos de usuarios: Jefe de la Diputación, Técnico de la diputación, Coordinador de Seguridad y Salud y Contratista

RQ-3 El sistema permitirá a un técnico de la Diputación crear una obra.

RQ-4 El sistema generará un código para una nueva obra con formato OBaa-999 (aa->año, 999 ->número del 001 al 999)

RQ-5 El sistema solicitará a un técnico de la Diputación asociar la obra a un coordinador de seguridad y salud.

RQ-6 El sistema solicitará a un técnico de la Diputación asociar la obra a un contratista

RQ-7 El sistema permitirá ver a los usuarios un listado de las obras en las que estén trabajando

RQ-8 El sistema permitirá filtrar las obras del listado

RQ-9 El sistema permitirá ver un detalle de la obra

RQ-10 El sistema permitirá al coordinador de seguridad y salud editar los datos de la obra.

RQ-11 El sistema permitirá a los usuarios editar sus datos personales.

RQ-12 El sistema permitirá a un coordinador de seguridad y salud crear el informe de visita de control.

RQ-13 El sistema permitirá a un coordinador de obra modificar la visita de control.

RQ-14 El sistema permitirá a un coordinador de obra enviar el informe a los involucrados.

RQ-15 El sistema no permitirá modificar una visita de control enviada

RQ-16 El sistema permitirá a un coordinador de obra cerrar la obra.

RQ-17 El sistema enviará un aviso a los implicados en la obra cuando esta haya sido creada.

RQ-18 El sistema enviará un aviso a los implicados en la obra cuando esta haya sido modificada.

RQ-19 El sistema enviará un aviso a los implicados en la obra cuando se haya enviado un informe de visita de control.

RQ-20 El sistema enviará un aviso a los implicados en la obra cuando se haya cerrado.

RQ-21 El sistema permitirá a los usuarios ver la información de las obras cerradas.

RQ-22 El sistema mostrará las acciones que hayan ocurrido sobre la obra

15. Casos de Uso

CU-1 Login usuario. Este caso de uso representa la operación de hacer login en la aplicación por parte de un usuario. Se corresponde con el requisito RQ-1

CU-2 Alta de obra. Este caso de uso representa la operación de dar de alta una obra. La realizará el técnico de la Diputación encargado de la obra y tiene que asociar a la misma un coordinador de seguridad y salud y una empresa contratista. Se corresponde con los requisitos RQ-3, RQ-4, RQ-5, RQ-6, RQ-2 y RQ-17.

CU-3 Modificación de obra. Este caso representa la operación de modificar una obra, al modificar la obra se incluyen:

los datos generales de la obra: Carretera, P.K, Localización, Presupuesto, Inicio, Duración aprox (meses).

la documentación relacionada: Aviso Previo, TC2 Trabajadores, Entrega de EPIs(Equipos de protección individual), Aprobación del plan de seguridad, Apertura del centro de trabajo, Libro incidencias, Libro de subcontrataciones, Plan de Obra, ITV Vehículos, Hoja de Hospitales y Emergencias más dos ficheros adicionales que denominamos Otros.

Esta acción será realizada únicamente por el coordinador de seguridad y salud. Se corresponde con el requisito RQ-10, RQ-2 y RQ-18.

CU-4 Modificación datos personales. Este caso de uso representa la operación de modificación de datos personales del usuario, incluye la modificación de datos de la empresa a la que pertenecen. Se corresponde con el requisito RQ-11

CU-5 Mostrar listado de obras. Este caso de uso representa la operación de visualizar un listado de obras. El usuario solo visualizará las obras con las que esté relacionado. Se corresponde con el requisito RQ-7, RQ-8 y RQ-21

CU-6 Detalle de obra. Este caso de uso representa la operación de visualizar el detalle de la obra. Únicamente permite la visualización y si el usuario es el coordinador de Seguridad y Salud se le habilitará un botón "Modificar" que le llevará a la modificación de la obra. Se podrán visualizar obras cerradas. Se corresponde con el requisito RQ-9 y RQ-21.

CU-7 Alta Visita de control. Este caso de uso representa la operación de dar de alta una visita de control. Esta operación solo la realizará el coordinador de Seguridad y Salud. Incluye los datos:
Equipamiento de protección individual: El coordinador de obra seleccionará qué elementos de protección individual está utilizando el contratista entre los siguientes: Casco, Pantalla facial, Arnés, Chaleco reflectante, Gafas, Botas, Orejeras, Guantes, Máscara, Gorra/similar. También indicará el estado general del equipamiento, los estados posibles son: Muy bueno, bueno, básico e insuficiente. Permitirá subir un máximo de dos imágenes que muestren el estado del equipamiento. El coordinador podrá incluir un texto de observaciones de un máximo de 1000 caracteres.

- Equipamiento de protección colectiva: El coordinador de obra seleccionará qué elementos de protección colectiva está utilizando el contratista entre los siguientes: Botiquín, Barandilla,

Redes de seguridad, Líneas de vida, Extintores CO2, Extintores Agua, Extintores Polvo.

También indicará el estado general del equipamiento colectivo, los estados posibles son: Muy bueno, bueno, básico e insuficiente. Permitirá subir un máximo de dos imágenes que muestren el estado del equipamiento. El coordinador podrá incluir un texto de observaciones de un máximo de 1000 caracteres.

- Señalización de obra: El coordinador de seguridad indicará si la señalización de obra se está utilizando en un sentido de la carretera o en dos. Seleccionará qué elementos de señalización de obra está utilizando el contratista entre los siguientes: Fija, Móvil, Conos, Semáforo provisional, Banda rugosa, Señaladores. También seleccionará qué señales se están utilizando entre las siguientes, las señales irán acompañadas por un icono: TP3, TP15, TP15a, TP15b, TP17, TP17a, TP17b, TP18, TP19, TP25, TP28, TP30, TP50, TR301, TR302, TR303, TR305. También indicará el estado general de la señalización, los estados posibles son: Muy bueno, bueno, básico e insuficiente. Permitirá subir un máximo de dos imágenes que muestren el estado de la señalización. El coordinador podrá incluir un texto de observaciones de un máximo de 1000 caracteres.

La visita contendrá un reportaje fotográfico, que consiste en un máximo de 6 fotografías

La visita tendrá una caja de texto donde el coordinador de seguridad y salud elaborará un informe de un máximo de 2500 caracteres. Se corresponde con el requisito RQ-12 y RQ-2.

CU-8 Modificación visita de control. Este caso de uso representa la operación de modificar la visita de control, únicamente será realizado por el coordinador de Seguridad y salud. No será posible la modificación de una visita de control que haya sido enviada. Únicamente realizará la modificación el coordinador de seguridad y salud. Se corresponde con el requisito RQ-13, RQ-15 y RQ-2

CU-9 Envío visita de control. Este caso de uso representa la operación de enviar la visita a todos los intervinientes en la misma. Envía un mail con información de la visita, un enlace para verla completa y añade la información en Noticias. Únicamente será realizado por el coordinador de seguridad y salud. Se corresponde con el requisito RQ-14, RQ-2 y RQ-19

CU-10 Cierre de obra. Este caso de uso representa la operación de cierre de obra. Únicamente será realizada por el coordinador de obra. El coordinador de obra introducirá una fecha y un resumen de la obra. Se corresponde con el requisito RQ-16 y RQ-20.

CU-11 Visualización de noticias. Este caso de uso representa la operación de visualizar las acciones cometidas en las obras, como son: Alta, Modificación, Visita de control y cierre. Cada usuario podrá visualizar las noticias de las obras con las que esté relacionado. Se corresponde con el requisito RQ-22.

Caso de uso	Jefe Diputación	Técnico Diputación	Coordinador Seguridad	Contratista
CU-1 Login usuario	X	X	X	X
CU-2 Alta de obra.		X		
CU-3 Modificación de obra.			X	
CU-4 Modificación datos personales.	X	X	X	X
CU-5 Mostrar listado de obras.	X	X	X	X
CU-6 Mostrar detalle de obra.	X	X	X	X
CU-7 Alta Visita de control.			X	
CU-8 Modificación visita de control.			X	
CU-9 Envío visita de control.			X	
CU-10 Cierre de obra.			X	
CU-11 Visualización de noticias.	X	X	X	X

Tabla 1: Relación casos de uso y actores

16. Diseño de clases

El hecho de utilizar dos *frameworks* como Struts2 e Hibernate nos va a simplificar el proceso de diseño y desarrollo de las clases.

Las clases encargadas de realizar las acciones heredan de la clase `ActionSupport`, suministrada por Struts. A estas clases les asignamos un nombre `xxxxxAction`.

Por otra parte, el código encargado de interactuar con la BBDD se incluye en clases con nombre `xxxxDAO`. Estas clases acceden a la BBDD mediante el uso de Hibernate y utilizarán las clases que mapean las tablas de la BBDD.

Por ejemplo, para la comprobación de un usuario y contraseña, definimos una clase `LoginAction` donde implementamos nuestra acción `comprobarLogin()`. Esta acción debe acceder a BBDD por lo que llama a la clase `UsuarioDAO` donde tenemos un método `obtenerClave()`, encargado de buscar si el usuario está en BBDD y obtener su clave. Esta clase `usuarioDAO` utiliza la clase `Usuario` que mapea nuestra tabla `Usuario`. El código de la clase `Usuario` se genera automáticamente con las herramientas incluidas en nuestro entorno de desarrollo.

Mediante UML podemos dibujar siguiente diagrama de clases:

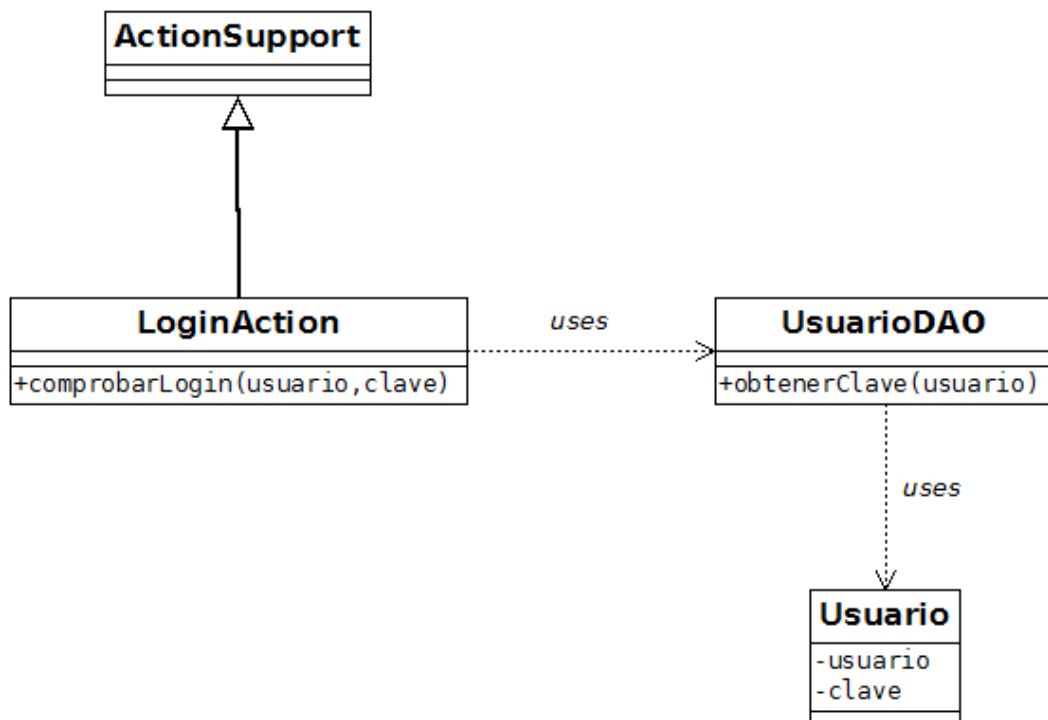


Figura 5: Diagrama de clases Usuario

El diagrama de secuencia queda de la siguiente manera:

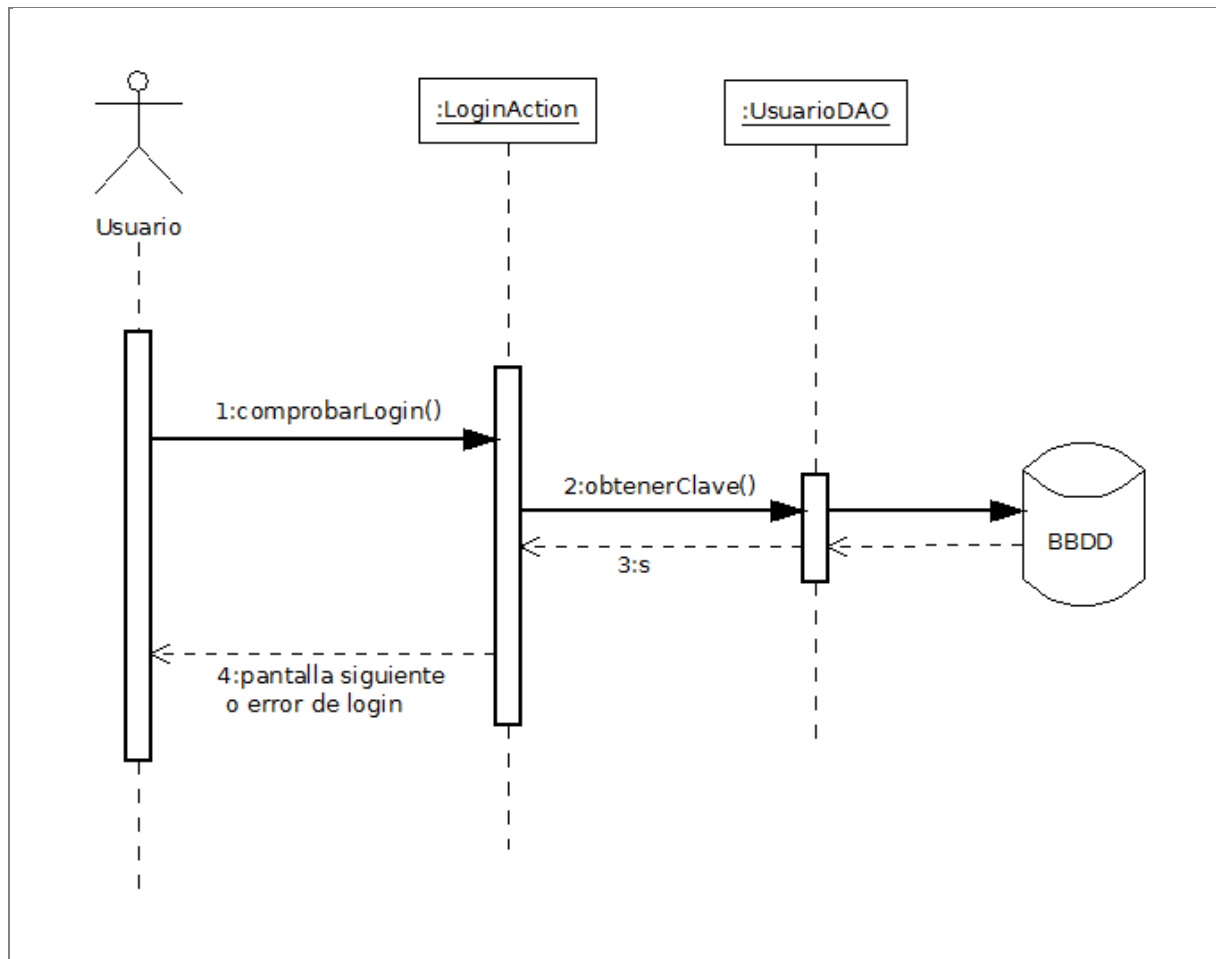


Figura 6: Diagrama de secuencia para el caso de uso Login

Siguiendo el mismo diseño podemos definir las siguientes clases encargadas de las acciones:

LoginAction: clase encargada de comprobar el acceso mediante usuario y contraseña y cargar el perfil de usuario en sesión.

ObraAction: clase encargada de realizar las acciones relacionadas con al obra: alta, modificación, consulta, cierre...

VisitaControlAction: clase encargada de realizar las acciones relacionadas con las visitas de control: alta, modificación, consulta, envío

DatosPersonalesAction: clase encargada de realizar las acciones relacionadas con los datos personales: alta, modificación, baja, consulta...

NoticiasAction: clase encargada de mostrar un listado de noticias y acceder al detalle de cada una de ellas

De la misma manera tendremos las siguientes clases encargadas de acceder a BBDD

- **UserDAO:** clase encargada de acceder a los datos de usuario en BBDD

- **ProjectDAO:** clase encargada de acceder a los datos de la obra en BBDD
- (Etc)

Además, por cada tabla, existe una clase que mapea dicha tabla y que tiene el mismo nombre que la tabla.

- User
- User_Data
- Company
- Project
- (Etc)

Nota. El diseño definitivo podrá variar durante la fase de desarrollo

17. Modelo de base de datos

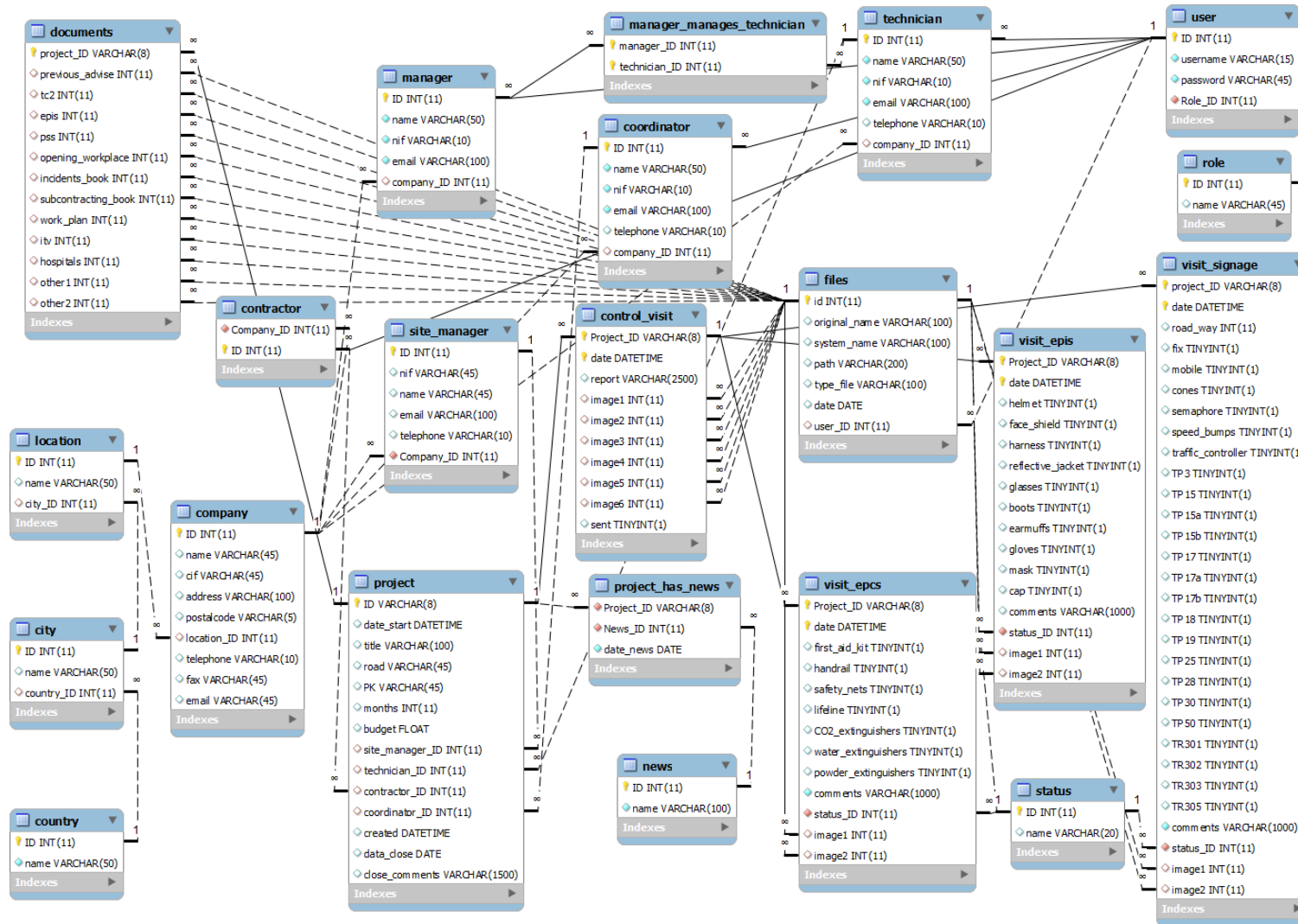


Figura 7: Modelo de BBDD

18. Diseño de pruebas

Código	Requisito a probar	Objetivo de la prueba	Precondiciones	Datos de entrada	Resultado esperado	Cumple Sí/No
PR-1	RQ-1	Hacer login	El usuario debe estar registrado	usuario clave	Accede a la aplicación sólo si el usuario y clave son correctos	
PR-2	RQ-2, RQ-3, RQ-4, RQ-5, RQ-6	Crear obra	El usuario debe estar registrado como Técnico de la Diputación y validado por el sistema	Título de la obra, coordinador de seguridad y salud y contratista.	La nueva obra queda registrada en el sistema. La creación de la obra queda registrada en Noticias.	
PR-3	RQ-7, RQ-8	Listar Obras	El usuario debe estar registrado y validado por el sistema	Filtros de obras (Código, Abierta/Cerrada, Título)	Se muestran los datos de las obras.	
PR-4	RQ-10	Modificar obra	El usuario debe estar registrado como Coordinador de Seguridad y Salud y validado por el sistema	Datos asociados a la obra	Se modifican los datos de la obra en el sistema. La modificación de la obra queda registrada en Noticias.	
PR-5	RQ-12	Dar de alta una visita de control	El usuario debe estar registrado como Coordinador de Seguridad y Salud y validado por el sistema	Datos asociados a la visita	La visita queda registrada en el sistema. El alta de la visita queda registrada en Noticias.	
PR-6	RQ-13, RQ-15	Modificar una visita de	El usuario debe estar registrado como Coordinador de Seguridad	Datos asociados a la visita	Se modifican los datos de la visita. La modificación queda registrada en	

Código	Requisito a probar	Objetivo de la prueba	Precondiciones	Datos de entrada	Resultado esperado	Cumple Sí/No
		control	y Salud y validado por el sistema. La visita tiene que haber sido dada de alta y no haberse enviado.		Noticias.	
PR-7	RQ-19	Enviar una visita de control	El usuario debe estar registrado como Coordinador de Seguridad y Salud y validado por el sistema	Código de la visita	El resto de involucrados en la obra, reciben un mail de la visita. El envío queda registrado en Noticias.	
PR-8	RQ-16	Cerrar obra	El usuario debe estar registrado como Coordinador de Seguridad y Salud y validado por el sistema	Código de la obra	La obra queda registrada como cerrada.	
PR-9	RQ-11	Modificar perfil	El usuario debe estar registrado y validado.	Datos asociados al perfil	Se modifican los datos del perfil.	
PR-10	RQ-10	Ver detalle obra	El usuario debe estar registrado como Coordinador de Seguridad y Salud y validado por el sistema. La obra tiene que estar dada de alta.	Código de obra	Se accede a los datos de la obra.	

Tabla 2: Diseño de pruebas

19. Prototipos

18.1 Login



Figura 8 : Prototipo de la pantalla de Login

18.2 Inicio (noticias)



Figura 9 : Prototipo de la pantalla de Noticias

18.3 Nueva Obra



Figura 10: Prototipo de la pantalla de Nueva Obra

18.4 Listado de obras



Figura 11: Prototipo de la pantalla de Listado de Obras

18.5 Modificación Obra



Protocolo de Seguridad y Salud

Inicio
Nueva Obra
Modificar Obra
Licéado
Contacto

OB14-001 - Conservación de elementos de estructura de puentes y obras de fabrica

Datos Generales

Carretera

P.K.

Localización

Presupuesto

Inicio

Duración aprox (meses)

Técnico Diputación

eMail

Teléfono

Extensión

Móvil

Contratista

Contratista

Dirección

Teléfono

Fax

eMail

Jefe de Obra

Teléfono

Documentación

Aviso Previo Ningún arch...leccionado

TC2 Trabajadores Ningún arch...leccionado

Entrega de EPIS(Equipos de protección individual) Ningún arch...leccionado

Aprobación del plan de seguridad Ningún arch...leccionado

Apertura del centro de trabajo Ningún arch...leccionado

Libro Inidennitas Ningún arch...leccionado

Libro de subcontrataciones Ningún arch...leccionado

Plan de Obra Ningún arch...leccionado

ITV Vehículos Ningún arch...leccionado

Hoja de Hospitales y Emergenas Ningún arch...leccionado

Otros Ningún arch...leccionado

Otros Ningún arch...leccionado

Figura 12: Prototipo de la pantalla de modificación obra

18.6 Detalle de obra



Protocolo de Seguridad y Salud

Inicio
Nueva Obra
Modificar Obra
Listado
Contacto

OB14-001 Conservación de elementos de estructura de puentes y obras de fabrica

Técnico

Maite Martínez Gil - Diputación de Barcelona
 ☎ 934022222 ✉ mmartinez@gencat.cat

Coordinador de Seguridad y Salud

Joan Ripoll Oliva - ASSY S.L
 ☎ 650556677 ✉ jripoll@assy.cat

Contratista

IFA Infraestructuras S.L
 ☎ 650112233 ✉ ifa@ifa.cat
 JEFE DE OBRA: Josep López
 ☎ 650987654

Datos Generales

Carretera: CV530 Presupuesto: 53.000	P.K: 23 Inicio: 25-10-2014	Localización: Avda Diagonal sn Duración aprox (meses): 5
---	-------------------------------	---

Documentación

Aviso Previo: OB14-001_avisoprevio.pdf	Libro de subcontrataciones: OB14-001_LibroSubcontr.pdf
TC2 Trabajadores: OB14-001_aprovisionPSS.pdf	Plan de Obra: OB14-001_HospitalesEmergencias.pdf
Entrega de EPIs: OB14-001_LibroSubcontr.pdf	ITV Vehiculos: OB14-001_avisoprevio.pdf
Aprobación del plan de seguridad: OB14-001_HospitalesEmergencias.pdf	Hoja de Hospitales y Emergencias: OB14-001_aprovisionPSS.pdf
Apertura del centro de trabajo: OB14-001_avisoprevio.pdf	Otros: OB14-001_LibroSubcontr.pdf
Libro Inordenolas: OB14-001_aprovisionPSS.pdf	Otros: OB14-001_HospitalesEmergencias.pdf

Visitas de control

Fecha	Estado EPI	Estado EPC	Señalización	Visita
10-10-2014	Insuficiente	Básica	Buena	ver
17-10-2014	Muy buena	Insuficiente	Básica	ver
24-10-2014	Muy buena	Básica	Básica	ver

Figura 13: Prototipo de la pantalla de Detalle de obra

18.7 Visita de obra



Protocolo de Seguridad y Salud

Inicio Nueva Obra Modificar Obra Listado Contacto

OB14-001 - Conservación de elementos de estructura de puentes y obras de fabrica

Equipamiento de protección individual EPI

<input type="checkbox"/> Casco	<input type="checkbox"/> Botas
<input type="checkbox"/> Pantalla facial	<input type="checkbox"/> Orejeras
<input type="checkbox"/> Arnés	<input type="checkbox"/> Guantes
<input type="checkbox"/> chaleco reflectante	<input type="checkbox"/> Máscara
<input type="checkbox"/> Gafas	<input type="checkbox"/> Gorra/similar



Estado general del equipamiento de protección individual



Muy Bueno Bueno Básico Insuficiente

Observaciones

Figura 14: Prototipo de la pantalla de Visita de obra - Equipamiento de protección individual

Equipamiento protección colectiva

<input type="checkbox"/> Botiquín	<input type="checkbox"/> Extintores CO2
<input type="checkbox"/> Barandilla	<input type="checkbox"/> Extintores Agua
<input type="checkbox"/> Redes de seguridad	<input type="checkbox"/> Extintores Polvo
<input type="checkbox"/> Líneas de vida	



Estado general del equipamiento de protección colectiva

Muy Bueno Bueno Básico Insuficiente

Observaciones

Figura 15: Prototipo de la pantalla de Visita de obra - Equipamiento de protección colectiva

Señalización de obra

1 Sentido de la carretera 2 Sentidos de la carretera

Fija Banda rugosa
 Móvil Señaladores
 Conos Extintores Polvo
 Semáforo provisional



TP3 TP15 TP15a TP15b TP17 TP17a TP17b TP18 TP19 TP25 TP28 TP30

TP50 TR301 TR302 TR303 TR305

Estado general de la señalización de obra


Muy Bueno Bueno Básico Insuficiente


Observaciones

Figura 16: Prototipo de la pantalla de Visita de obra - Señalización de obra

Informe de la visita

Reportaje fotográfico








Figura 17: Prototipo de la pantalla de Visita de obra - Informe de visita y reportaje fotográfico

18.8 Cierre de obra



Prototipo de la pantalla de Cierre de obra. La interfaz muestra un encabezado con un ícono de frutos y el título "Protocolo de Seguridad y Salud". Debajo hay una barra de navegación con botones para "Inicio", "Nueva Obra", "Modificar Obra", "Listado" (destacado) y "Contacto". El contenido principal muestra el ID de obra "OB14-001" y su descripción "Conservación de elementos de estructura de puentes y obras de fabrica". Sección "Cierre de Obra" con un campo "Fecha de cierre" (formato dd/mm/aaaa) y un área de "Observaciones" con un botón "Aceptar" en la esquina inferior derecha.



Protocolo de Seguridad y Salud

Inicio Nueva Obra Modificar Obra **Listado** Contacto

OB14-001 - Conservación de elementos de estructura de puentes y obras de fabrica

Cierre de Obra

Fecha de cierre

Observaciones

Aceptar

Figura 18: Prototipo de la pantalla de Cierre de obra

18.9 Perfil

The image shows a user profile page for 'Protocolo de Seguridad y Salud'. At the top left is a profile picture of a person wearing a blue surgical cap and mask. To the right of the picture is the title 'Protocolo de Seguridad y Salud' and a user name 'Cristina Calvo' with a dropdown arrow. Below this is a navigation bar with five buttons: 'Inicio', 'Nueva Obra', 'Modificar Obra', 'Listado', and 'Contacto'. The main content area is titled 'Perfil' and is divided into three sections: 'Datos personales', 'Modificar contraseña', and 'Datos Empresa'. Each section contains several input fields for text and email addresses. At the bottom right, there are two buttons: 'Cancelar' and 'Guardar'.

Protocolo de Seguridad y Salud Cristina Calvo

Inicio Nueva Obra Modificar Obra **Listado** Contacto

Perfil

Datos personales

Nif Nombre

Teléfono email

Modificar contraseña

Contraseña Actual Nueva contraseña Repetir Nueva contraseña

Datos Empresa

Cif Nombre

Teléfono email

Figura 19: Prototipo de la pantalla de Perfil

20. Configuración entorno de trabajo

MySQL Es un sistema gestor de bases de datos gratuito y ampliamente utilizado en proyectos web. También se puede utilizar con OpenShift.

Java SE Development Kit 7. Java es un lenguaje que se utiliza especialmente en el desarrollo de aplicaciones web. En nuestro caso estamos usando la versión 7. Se puede usar gratuitamente y además es compatible con OpenShift.

Maven. Es una herramienta que automatiza la compilación de proyectos, se utiliza especialmente con java. Además de compilar el proyecto, nos permite resolver las dependencias de las bibliotecas que necesitamos, conectándose a un repositorio y descargándose las bibliotecas indicadas. También es utilizada por OpenShift para los proyectos allí desplegados.

Tomcat Se trata de un contenedor de servlets de java que vamos a utilizar como servidor de nuestra aplicación java. Además de tener en OpenShift un servidor Tomcat, instalaremos otro en nuestra máquina para el desarrollo en local.

Jboss Developer Studio 7 G.A. Es un entorno de desarrollo basado en eclipse y desarrollado por Jboss que a su vez pertenece a Red Hat. Está especialmente diseñado para el desarrollo de aplicaciones web en java y se integra con OpenShift

Struts 2 Es un framework de desarrollo de aplicaciones web en java. Nos va a proporcionar la estructura sobre la que crearemos el proyecto

Hibernate. Es un framework para aplicaciones java que nos permite traducir el modelo entidad relación de las BBDD a un modelo orientado a objetos. Nos simplifica el acceso y consultas a BBDD

Configuración de OpenShift

En primer lugar abrimos una cuenta gratuita en OpenShift, para ello accedemos a:

<https://www.openshift.com/>

Y haciendo click en la sección *Online-Sign up for free*, accedemos a la siguiente página donde podemos obtener una cuenta gratuita:

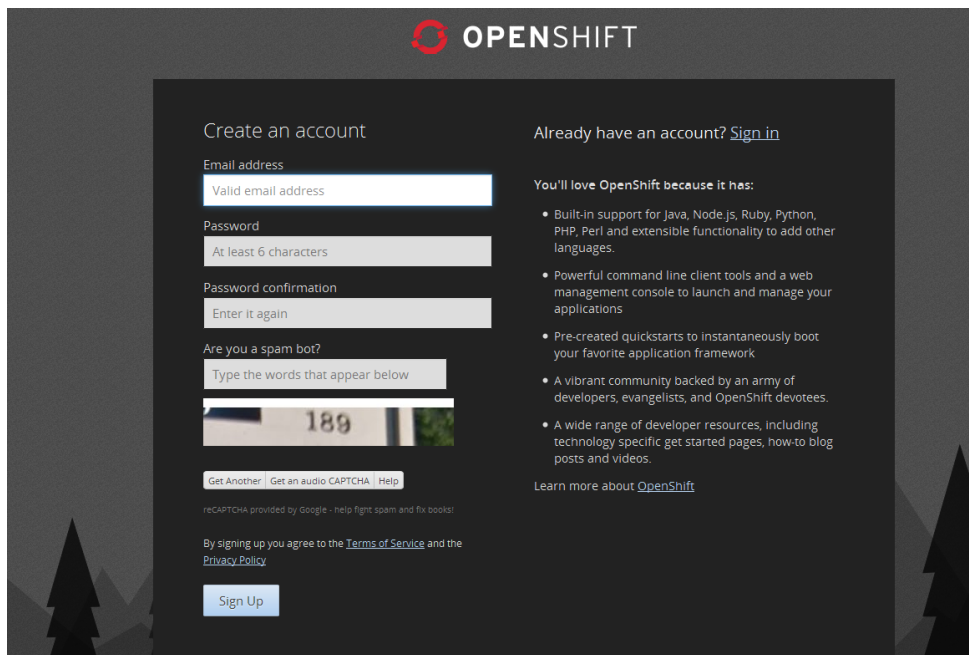


Figura 20: OpenShift

Una vez que nos hemos dado de alta, accedemos a nuestra cuenta y haciendo click en “Add Application”, nos muestra un listado de las herramientas que tenemos disponibles, en primer lugar hay que seleccionar el tipo de aplicación: “Tomcat 7”

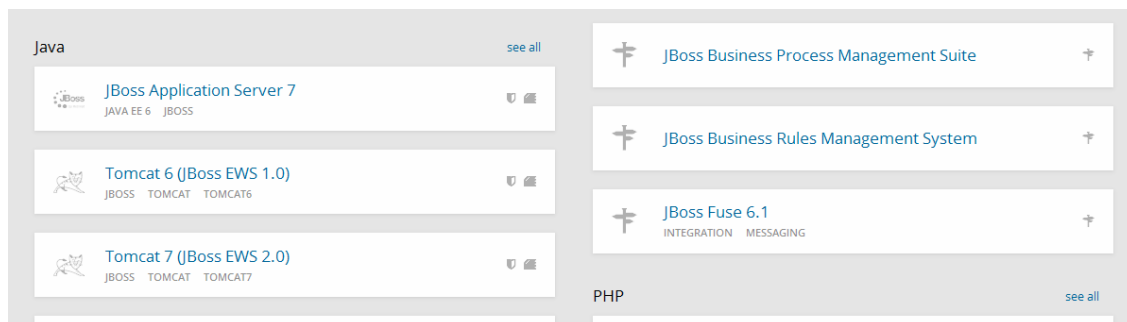


Figura 21: Add Application Open Shift

Introducimos el nombre del dominio que vamos a usar para acceder a través de internet y hacemos click en “Create Application”. Hay que esperar unos minutos y se nos crea automáticamente una aplicación y nos muestra las instrucciones para acceder a ella mediante el sistema de control de versiones *git*. En la siguiente pantalla nos aparecerán opciones para añadir bases de datos, en nuestro caso se selecciona *MySQL 5.5*.

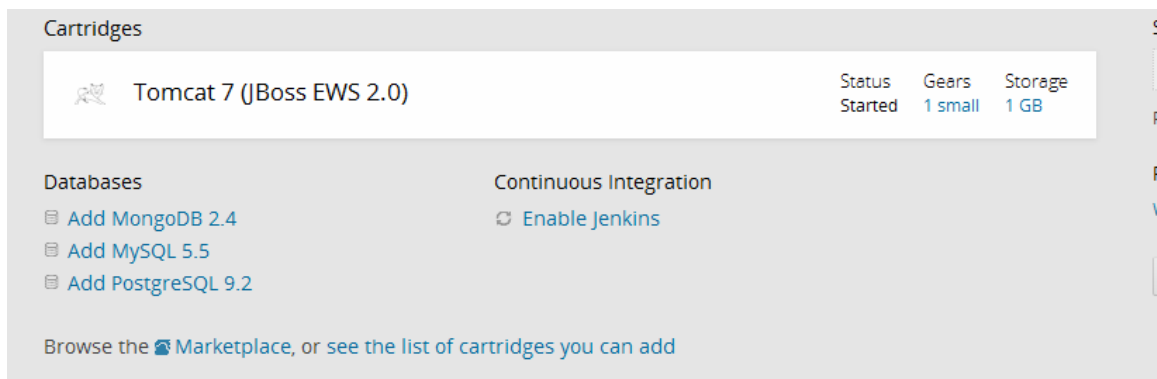


Figura 22: Add MySQL 5.5

Configuración de MySQL

Además de usar la BBDD de OpenShift como entorno de integración también instalamos MySQL en nuestra máquina local. Accedemos a <http://dev.mysql.com/downloads/mysql/> y descargamos la versión 5.5 para nuestra arquitectura. En Windows, ejecutamos el fichero descargado con extensión .msi. Durante la instalación nos pedirá que introduzcamos la contraseña para el usuario root de administración.

También es recomendable utilizar un entorno gráfico para gestionar MySQL. Hay varios disponibles de forma gratuita, por ejemplo MySQLWorkbench: <http://www.mysql.com/products/workbench/>
En la BBDD local creamos el mismo usuario que se creó en *OpenShift*, mediante:

```
CREATE USER 'usuarioDeAdministracion'@'localhost' IDENTIFIED BY 'claveDelUsuario';
GRANT ALL PRIVILEGES ON jbossews.* TO 'usuarioDeAdministracion'@'localhost';
FLUSH PRIVILEGES;
```

Configuración de Java

Para el desarrollo de aplicaciones con Java es necesario disponer de *Java Development Kit*.

Descargamos la versión 7 de la página de Oracle:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

Hay que seleccionar la versión para nuestra plataforma. Para sistemas *Windows* descarga un ejecutable que se encarga de instalarlo. Además, como vamos a usar *Maven* desde consola, configuramos la variable de entorno `JAVA_HOME`. Para ello, en el Panel de Control de *Windows*, buscamos 'Variables de Entorno'.

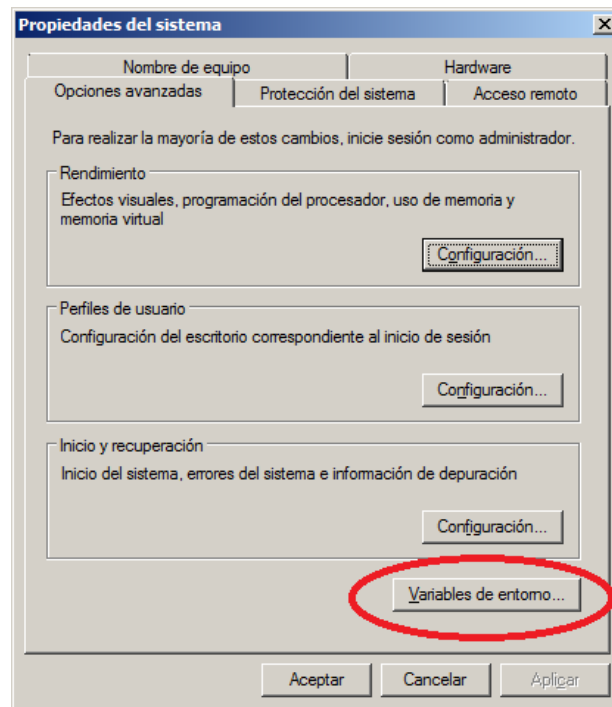


Figura 23: Variables de entorno

A continuación añadimos la nueva variable `JAVA_HOME`, con la ruta de nuestra instalación de java:

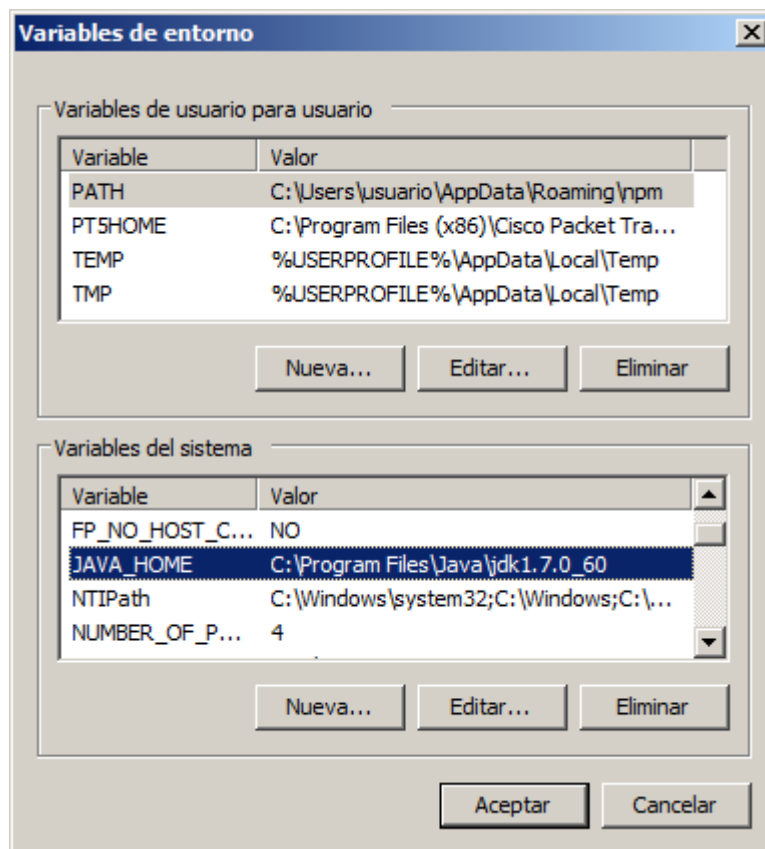


Figura 24: Insertar variable de entorno JAVA_HOME

También, en esa misma ventana, hay que añadir a la variable `Path`, el valor `%JAVA_HOME%\bin;`

Es necesario reiniciar **Windows** para que actualice los cambios

Configuración de Maven

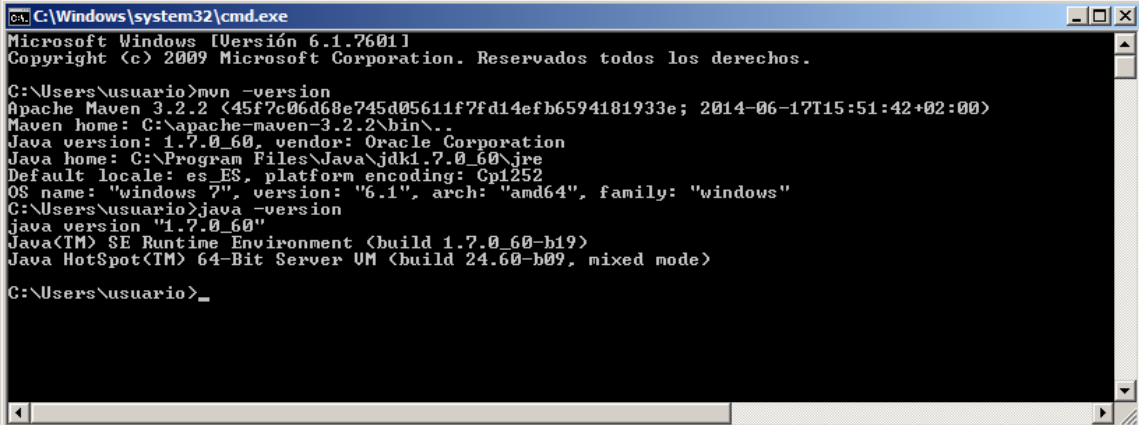
Para instalar *Maven* primero hay que descargarlo desde: <http://maven.apache.org/download.cgi>. En nuestro caso estamos usando la versión 3.2.2. Nos descargamos un fichero en formato .zip que descomprimos en cualquier carpeta de nuestra elección, por ejemplo directamente en la unidad C: de Windows. A continuación añadimos a la variable de entorno Path, la ruta de la carpeta con los binarios de *Maven*. En nuestro caso hay que añadir, al igual que se indicó en el punto anterior, la ruta:

```
C:\apache-maven-3.2.2\bin
```

Para actualizar los cambios, reiniciamos el sistema. Se puede comprobar que tanto el JDK de java y Maven están configurados correctamente abriendo un terminal y ejecutando:

```
mvn -version
```

Debemos obtener un resultado como el siguiente:



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\usuario>mvn -version
Apache Maven 3.2.2 (45f7c06d68e745d05611f7fd14efb6594181933e; 2014-06-17T15:51:42+02:00)
Maven home: C:\apache-maven-3.2.2\bin\..
Java version: 1.7.0_60, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.7.0_60\jre
Default locale: es_ES, platform encoding: Cp1252
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
C:\Users\usuario>java -version
java version "1.7.0_60"
Java(TM) SE Runtime Environment (build 1.7.0_60-b19)
Java HotSpot(TM) 64-Bit Server VM (build 24.60-b09, mixed mode)

C:\Users\usuario>_

```

Figura 25: Maven version

Mediante *Maven* añadiremos todas las bibliotecas necesarias para nuestra aplicación, como puede ser el driver para conectar con *MySQL*, la biblioteca de *Struts 2*, la biblioteca de *Hibernate* y cualquier otra que consideremos necesaria.

Configuración de Tomcat

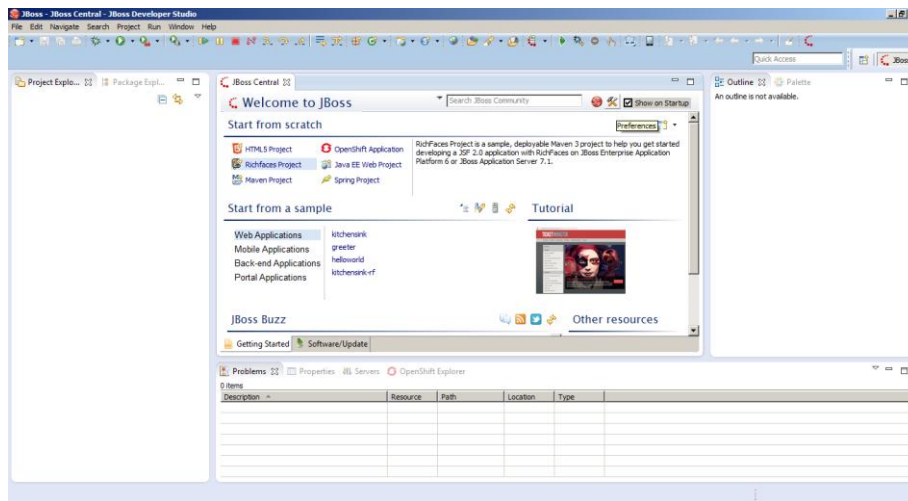
Para instalar el servidor Tomcat en nuestra maquina local, basta con descargarlo desde: <http://tomcat.apache.org/download-70.cgi> Nos descargamos el .Zip y los descomprimos en alguna carpeta de nuestra elección. Por ejemplo, directamente en la unidad C: en Windows. Posteriormente lo integramos con el entorno de desarrollo *Jboss Developer Studio*

Configuración de Jboss Developer Studio

Podemos descargar *Jboss Developer Studio* desde:

<http://www.jboss.org/products/devstudio/download/>

En nuestro caso descargamos la versión 7.1.1.GA . Haciendo doble click en el jar que nos descargamos comienza la instalación. Debemos aceptar los términos de la licencia y tras la instalación podemos arrancar el *Jboss Developer Studio* obteniendo la siguiente pantalla:



Para añadir el servidor *Tomcat* a nuestro entorno de desarrollo, hacemos click en la pestaña *Servers*, y con el botón derecho seleccionamos *New/Server*. En la ventana siguiente seleccionamos el tipo de servidor:

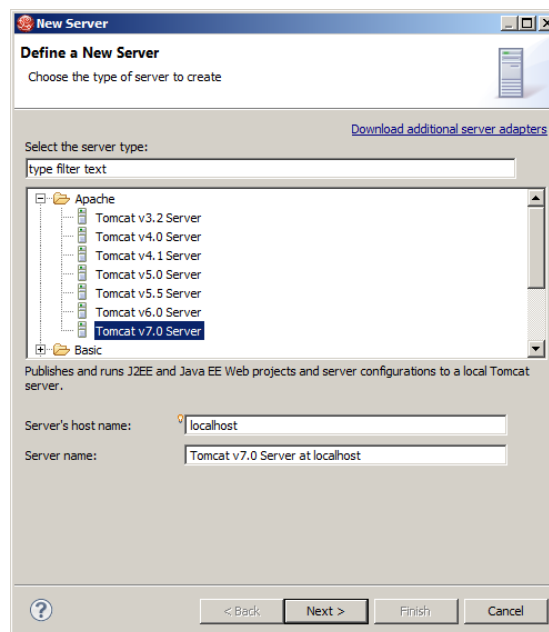
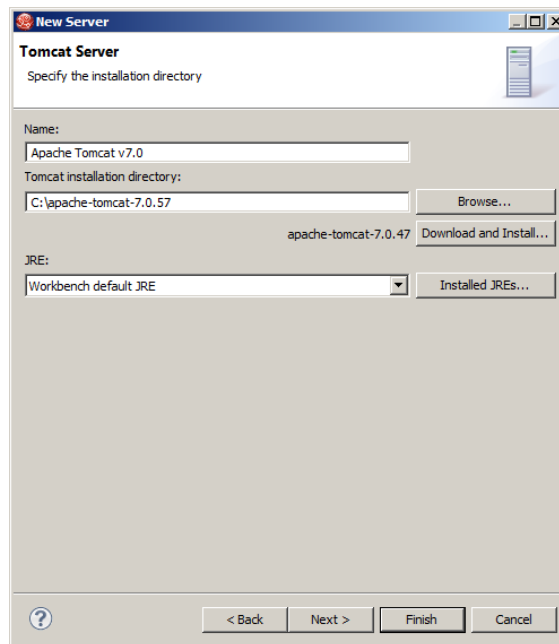


Figura 26: Añadir servidor Tomcat

A continuación introducimos el directorio donde previamente hemos instalado *Tomcat*:



Para configurar la conexión entre Tomcat y la BBDD MySQL, modificamos el fichero context.xml del servidor que nos acabamos de incluir y añadimos lo siguiente al nodo `<Context>`:

```
<Resource name="jdbc/MySQLDS" auth="Container" type="javax.sql.DataSource"
    maxActive="100" maxIdle="30" maxWait="10000"
    username="xxxxxxx" password="xxxxxxx"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/jbossews"/>
```

Tenemos que modificar el usuario y password con los de nuestra base de datos.

A continuación vamos a descargarnos la aplicación creada en *OpenShift* en el primer paso. Para transferir los ficheros de la aplicación entre nuestro entorno local y *OpenShift*, se utiliza el sistema de control de versiones Git. La comunicación se realiza utilizando el protocolo *ssh* mediante el uso de clave pública/privada. Debemos crearnos una clave pública y privada en nuestro local para el usuario de *Openshift*. Para ello seguimos las instrucciones en:

https://access.redhat.com/documentation/en-US/Red_Hat_JBoss_Developer_Studio/7.0/html/User_Guide/ch08s02s05.html

Haciendo click en la pestaña '*OpenShift Explorer*' debemos crear una nueva conexión a *Openshift* con el botón derecho y '*New connection*' donde introducimos nuestro usuario y contraseña de *Openshift*. Posteriormente, en la conexión que acabamos de crear, con el botón derecho del ratón, seleccionamos '*manage SSH keys*' y creamos las claves publica/privada.

Una vez que tenemos la conexión, podemos descargar la aplicación desde el '*Project Explorer*' con el botón derecho seleccionamos *import/OpenShift/Existing OpenShift Application*.

Url de la aplicación dentro del servidor:

<https://jbosssews-ccalvouoc.rhcloud.com/jbosssews>

Configuración de Struts 2

Para configurar Struts es necesario descargarse la biblioteca de Apache Struts 2. Como este proyecto está usando maven, la forma más sencilla de descargarse la biblioteca es añadir la referencia al fichero pom.xml de nuestra aplicación. Basta con añadir al nodo 'dependencies' lo siguiente:

```
<dependency>
  <groupId>org.apache.struts</groupId>
  <artifactId>struts2-core</artifactId>
  <version>2.3.16.3</version>
</dependency>
```

Luego se instala con Maven. Se puede hacer directamente desde consola, nos situamos en la carpeta donde se encuentra el fichero pom.xml y ejecutamos:

```
mvn clean
mvn install
mvn eclipse:eclipse -Dwtpversion=2.0
```

Al abrir el proyecto con *Jboss Developer Studio* ya nos aparece el el jar correspondiente. A continuación hay que añadir el filtro de Struts al fichero src\main\webapp\WEB-INF\web.xml. Para ello, en este fichero añadimos las siguientes líneas dentro del nodo webapp:

```
...
<filter>
  <filter-name>struts2</filter-name>

  <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>struts2</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
...
```

Con eso ya podemos empezar a desarrollar usando la especificación de Struts. El desarrollo de Struts está basado en clases java que ejecutan 'acciones' y despachan páginas .jsp. La definición de las 'acciones', las clases java asociadas y los jsp despachados se definen en el fichero de configuración src\resources\struts.xml (ver código fuente)

Configuración de Hibernate

La configuración de hibernate se realiza mediante maven de forma similar a como se ha instalado Struts 2. Para ello añadimos las siguientes dependencias en el fichero pom.xml:

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>3.5.6-Final</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-annotations</artifactId>
  <version>3.5.6-Final</version>
</dependency>
```

Y se procede de forma similar a como se hizo con Struts2.

Para generar los ficheros de configuración de hibernate, utilizamos las herramientas de jboss para hibernate, incluidas en el entorno *Jboss Developer Studio* según se indican en el documento 'Ingeniería inversa con hibernate'

21. Perfiles de usuario

Los usuarios que utilizarán la aplicación se dividen en cuatro perfiles:

21.1 Jefe de la Diputación

Es el responsable de coordinar todas las obras y el trabajo de los técnicos que tenga a su cargo.

Funciones y permisos dentro de la aplicación

- Ver actualizaciones de las obras
- Ver listado de obras
- Ver detalle de las obras
- Ver visitas de control de las obras
- Contactar con la plataforma para informar de errores, sugerencias o quejas.

21.2 Técnico de la Diputación

Responsable de coordinar el trabajo técnico de las obras.

Funciones y permisos dentro de la aplicación

- Crear las obras y asignarlas a un coordinador y un contratista.
- Ver listado de obras
- Ver detalle de las obras
- Ver visitas de control de las obras
- Contactar con la plataforma para informar de errores, sugerencias o quejas.

21.3 Coordinador de Seguridad y Salud

Responsable de coordinar los aspectos de seguridad y salud en las obras que tenga asignadas, incorporar la información y documentación de estas a la plataforma, realizar las visitas de control oportunas y documentarlas dentro de la aplicación.

Funciones y permisos dentro de la aplicación

- Editar obra
- Crear, editar y enviar visita de control
- Ver listado de obras
- Ver detalle de las obras
- Ver visitas de control de las obras
- Contactar con la plataforma para informar de errores, sugerencias o quejas.

21.4 Empresa contratista

Encargada de la ejecución de las obras. Tiene como obligación enviar la documentación concerniente al coordinador de obra, cumplir con la normativa de seguridad y salud y rectificar los sistemas de seguridad y salud bajo las indicaciones del coordinador cuando este se lo indique.

Funciones y permisos dentro de la aplicación

- Ver actualizaciones de las obras
- Ver listado de obras
- Ver detalle de las obras
- Ver visitas de control de las obras
- Contactar con la plataforma para informar de errores, sugerencias o quejas.

22. Bugs

Se han detectado algunos errores leves que por falta de tiempo no se han podido arreglar, son los siguientes:

- Tamaño y tipo de los campos de input en los formularios. Hay que validarlos antes de intentar grabarlos.
- En edición de obra si alguno de los datos introducido es erróneo salta la pantalla “grabando” y no el evento de validación de datos aunque si está hecho. Primero tiene que hacer la validación de los datos y sólo si puede grabar sacar el mensaje de “Grabando”
- Cuando la obra está cerrada no se puede modificar las visitas. Ahora lo permite.
- Campo presupuesto en la pantalla de edición de obra, tiene errores de formato con el punto decimal y al grabar confunde los decimales con valores enteros.

23. Proyección a futuro

El resultado de este proyecto es una aproximación a las necesidades reales de la Diputación para gestionar las obras y el cumplimiento de la normativa en seguridad y salud, aun así, establece una base muy sólida para seguir evolucionando con el desarrollo al mismo tiempo que constituye un marco de trabajo en el cual los participantes puedan ir familiarizándose y contribuyendo en su mejora y adaptación.

A continuación, expongo las mejoras que considero deberían realizarse.

Mejoras dentro de la versión actual

Filtro para el listado de obras. Crear filtros con los campos principales para el listado de las obras y mostrar en un primer momento solo las obras activas.

Paginación en los listados. Actualmente hay muy pocas obras dadas de alta, pero a medida que se van añadiendo obras, es necesario que estas estén paginadas y no ocupen más de dos pantallas.

Encriptación de claves de usuario. Las claves de usuario deben guardarse cifradas en base de datos.

Ayuda directa en web: Añadir una ayuda interactiva dentro de la web.

Ampliaciones futuras

Creación de coordinador y contratista. Crear las pantallas y funcionalidad para crear los nuevos contratistas y coordinadores de obra a los que se adjudiquen obras y no existan en la base de datos. El encargado de generar estos datos sería el técnico de la diputación.

Envío de documentación por parte del contratista. Hasta ahora el contratista envía la información al coordinador vía mail y este le reclama la documentación que falta, hay que proporcionar a través de la aplicación el entorno para realizar este proceso.

Registro de todos los trabajadores. El contratista podrá y deberá registrar todos los empleados que trabajan en la obra.

Registro de empresas subcontratadas. La empresa contratista podrá subcontratar servicios a otras empresas para el desarrollo de la obra, desarrollar una ventana que permita la inserción y edición de esos datos.

Avisos. Desarrollar una ventana para que el coordinador de obra pueda enviar avisos al contratista.

Visita de control en pdf. Cuando la visita de control sea enviada generar un pdf con el resultado.

Firma electrónica de la visita de control. Para posibles acciones legales, sería necesario que el documento pdf de visita de control llevara la firma electrónica del coordinador de seguridad y salud.

24. Conclusiones

A través del desarrollo de este trabajo de fin de grado he tenido la posibilidad de poner en práctica los conocimientos adquiridos a lo largo del grado, principalmente los provenientes de las asignaturas: Lenguajes y Estándares Web, Programación, Programación Web, Programación Orientada a objetos, Ingeniería del Software, Diseño y Uso de Bases de Datos, etc. Asimismo, he tenido la oportunidad de utilizar tecnologías de mayor complejidad, como son Java, Struts2, Hibernate, lo cual ha requerido un mayor esfuerzo y al mismo tiempo un mayor aprendizaje.

Una buena planificación inicial y un seguimiento de esta en las fases siguientes me ha permitido desarrollar con éxito el trabajo y conseguir los objetivos establecidos, de esta manera, tanto en la fase de Análisis y Diseño como en la fase de Implementación el desarrollo ha seguido las pautas establecidas y no se han tenido que aplicar cambios considerables que no estuvieran previstos en los riesgos.

Finalmente, este trabajo supone la culminación de cuatro años de estudio, la satisfacción de haber aprovechado y adquirido todos los conocimientos esperados y el orgullo de haber logrado desarrollar una herramienta original, lógica y sencilla, adaptable a todo tipo de dispositivos cuyo uso repercute en la seguridad de los trabajadores.

Anexo 1. Entregables del proyecto

Proyecto completo en web

La aplicación está accesible a través de la dirección: <https://jbossews-ccalvouoc.rhcloud.com/> para acceder a ella se pueden utilizar los siguientes datos de acceso, según el rol del usuario:

Jefe de la Diputación

- **Usuario:** user1
- **Password:**tfg

Técnico de la Diputación

- **Usuario:** user2
- **Password:**tfg

Jefe de la Diputación

- **Usuario:** user3
- **Password:**tfg

Técnico de la Diputación

- **Usuario:** user4
- **Password:**tfg

Memoria del proyecto

El presente documento.

Código fuente

Código fuente de la aplicación, con el fichero de dependencias en el fichero *jbossews.rar*
Script de creación de base de datos en el fichero *jbossews.sql*

Defensa audiovisual

Video de no más de 15 minutos donde se expone de forma rigurosa en qué consiste el trabajo, como se ha llevado a término y los resultados. Incluye demostración de la plataforma.

Presentación libre

Presentación breve visual para dar a conocer al público general la aplicación y sus usos.

Autoinforme de evaluación

Informe de reflexión sobre las seis competencias transversales utilizadas durante el proyecto.

Anexo 2. Ingeniería inversa con Hibernate

Antes de empezar, conviene tener a mano los datos de la Base de Datos (nombre, usuario, password)
También hay que tener localizado el driver de MySQL. Es buena idea colocar una copia de dicho driver en la carpeta de nuestro Tomcat local: [Home de Tomcat]/lib.

Para generar las clases de mapeo de tablas, lo primero que hay que hacer es abrir la vista 'Hibernate' en el entorno de desarrollo JBoss Developer Studio

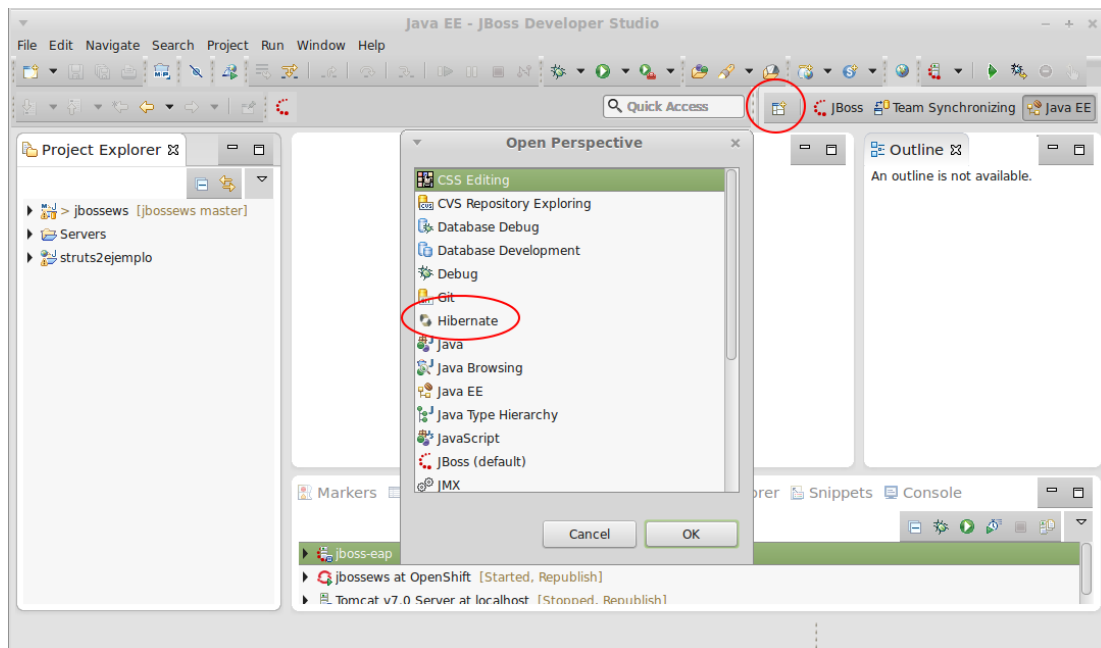


Figura 27: Hibernate, ingeniería inversa

A continuación hay que dar de alta una configuración de Hibernate con la BBDD:

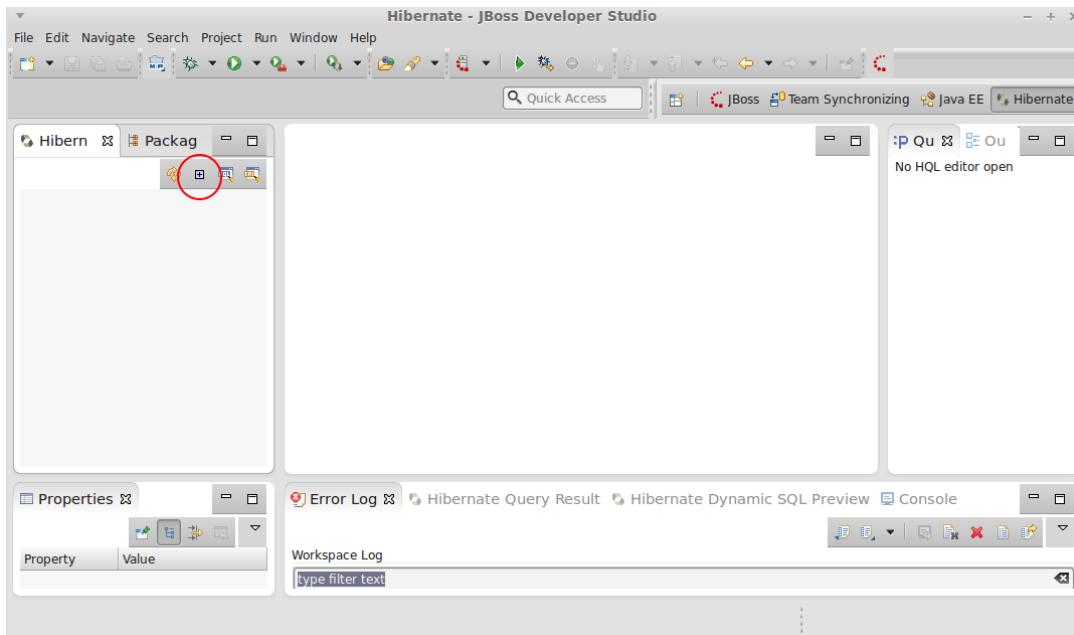


Figura 28: Alta configuración Hibernate.

En la siguiente pantalla se indica el tipo anotaciones, la versión que vamos a utilizar, y el proyecto sobre el que vamos a trabajar.

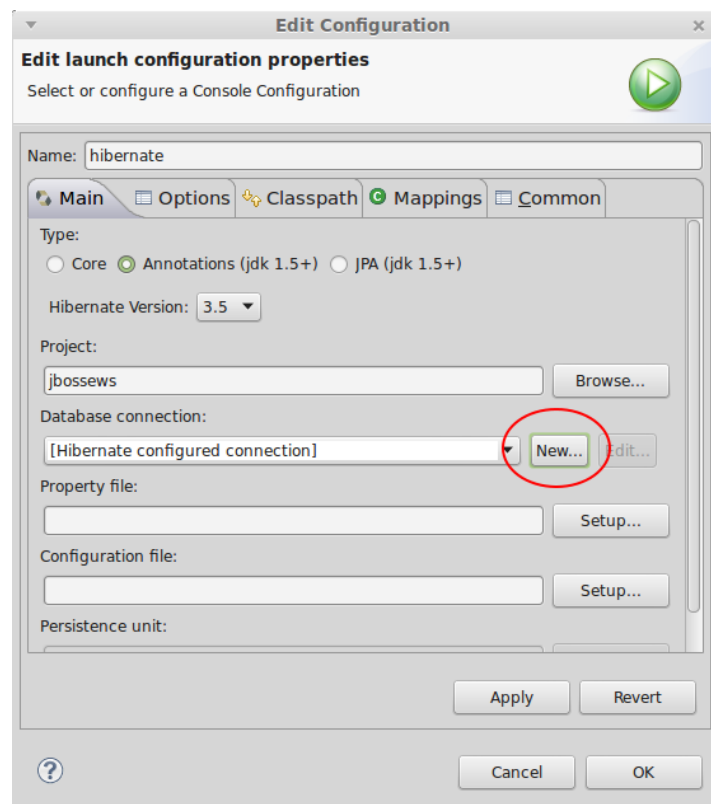


Figura 29: Configuración Hibernate edición

En la siguiente pantalla se selecciona el tipo de BBDD a utilizar:

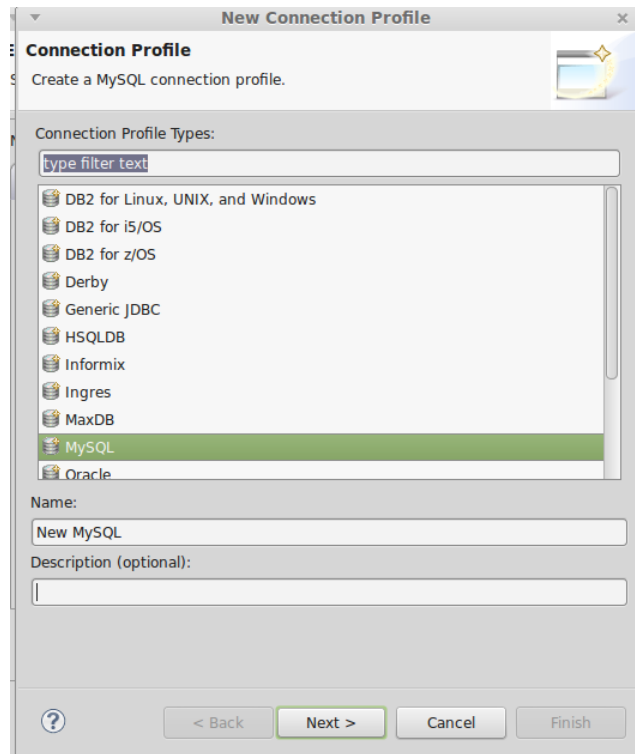


Figura 30: Seleccionar base de datos para Hibernate

Es necesario indicar el driver. Por defecto viene el nombre de uno que hay que eliminar y usar el nuestro.

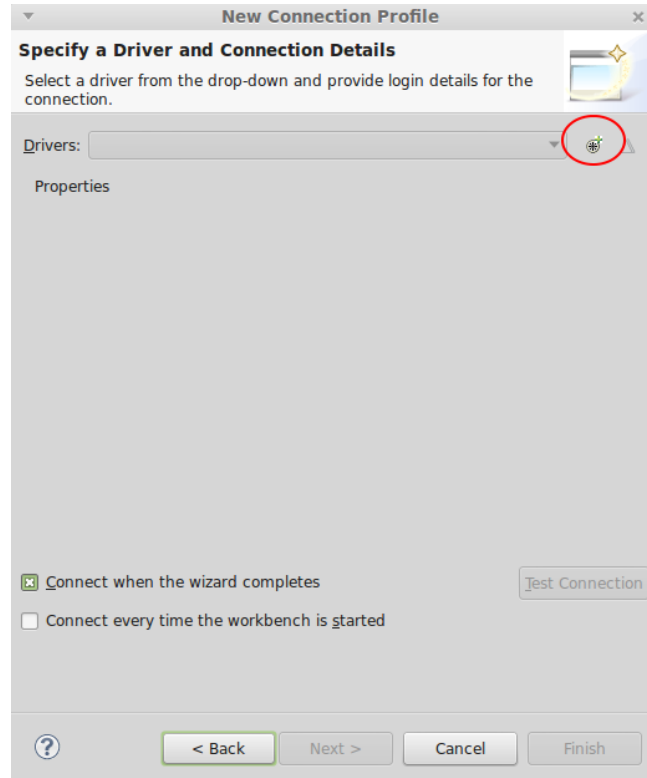


Figura 31: Driver base de datos de Hibernate.

Hacemos click en el nombre del driver que se muestra en la siguiente pantalla y a continuación 'Remove JAR/ZIP'

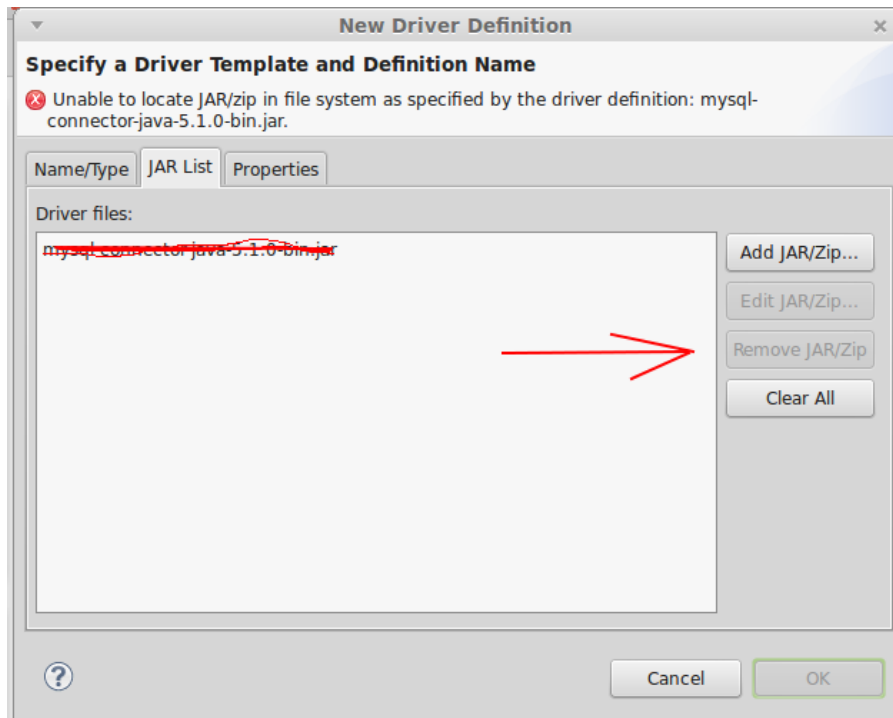
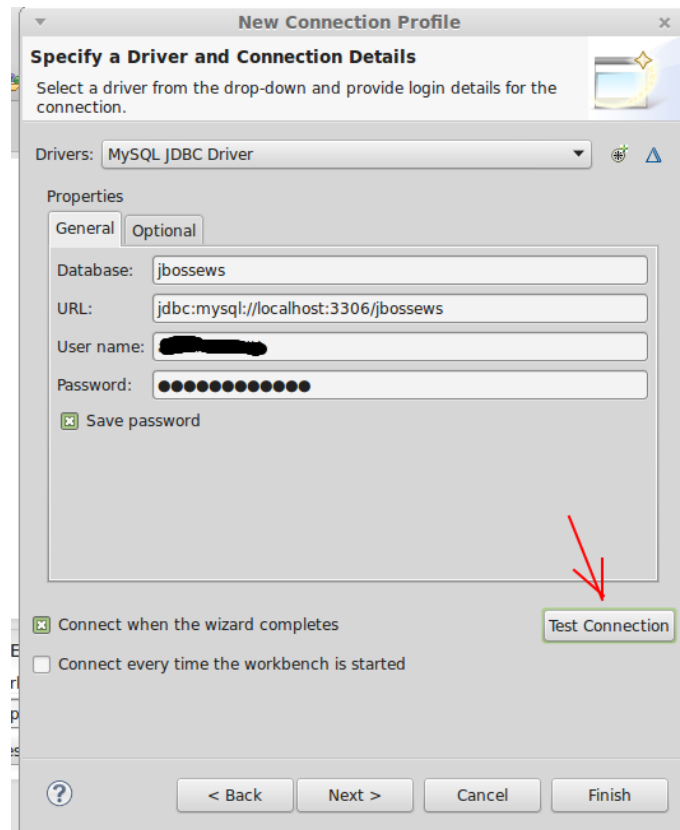
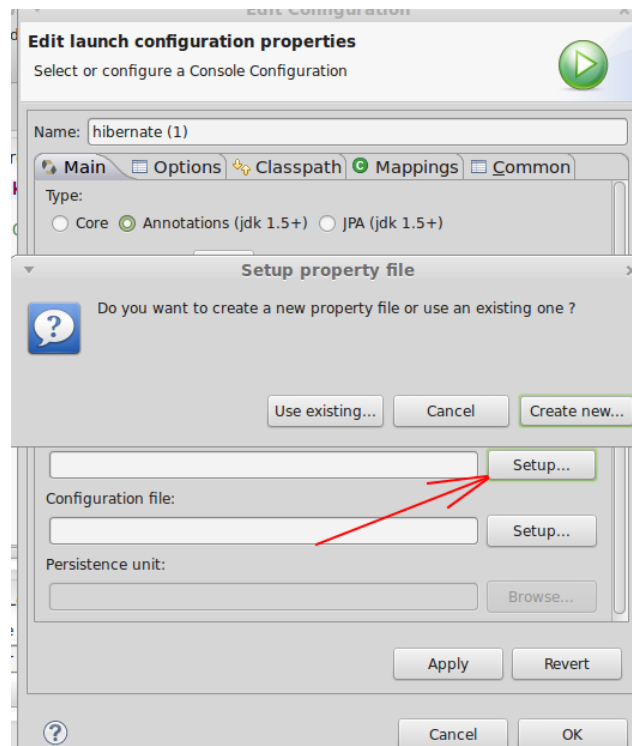


Figura 32: Eliminar driver por defecto de Hibernate

Después indicamos el driver que realmente tenemos. Rellenamos los datos de la conexión a BBDD y pulsamos el botón 'Test connection' para probar que funciona OK:



Volvemos a la pantalla de configuración de Hibernate donde tenemos que indicar el fichero de propiedades hibernate.properties. Hacemos click en 'Setup' y luego en 'Create new'



Y seleccionamos la siguiente ruta para guardar el fichero:

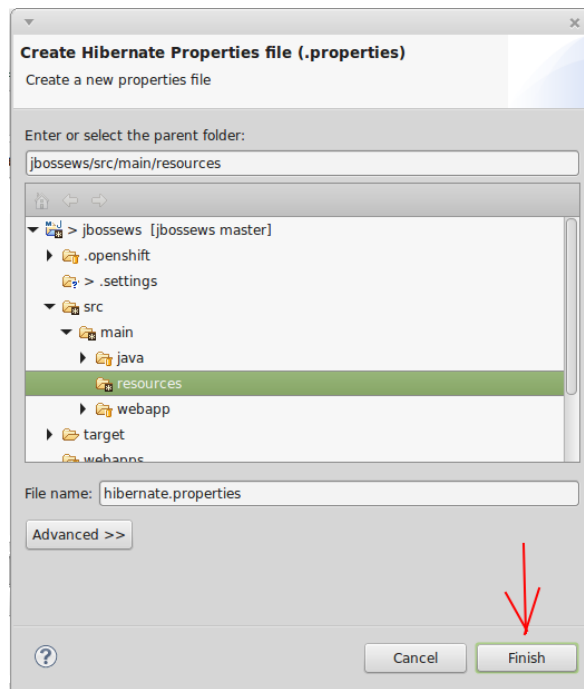


Figura 33: Ruta fichero properties de Hibernate

A continuación hacemos lo mismo para crear el fichero de configuración:

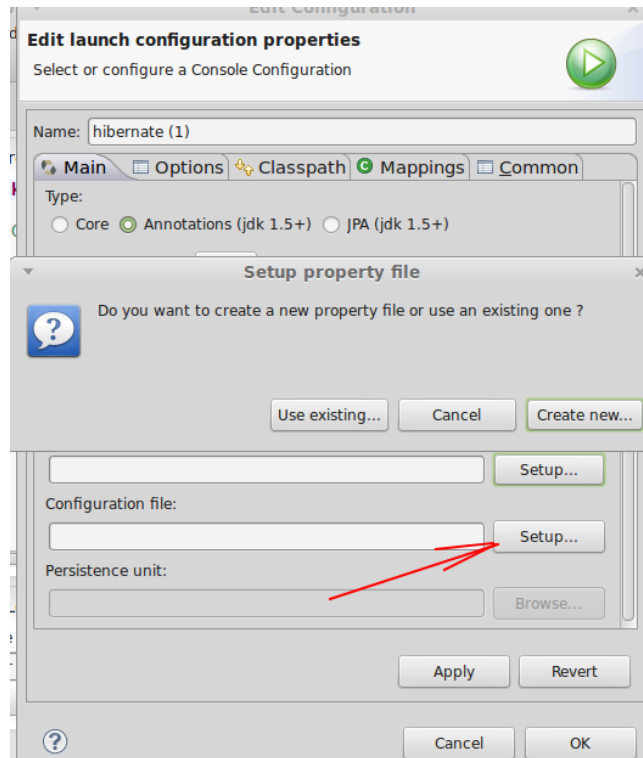


Figura 34: Fichero configuración Hibernate

Seleccionamos la misma carpeta resources y dejamos el nombre: hibernate.cfg.xml

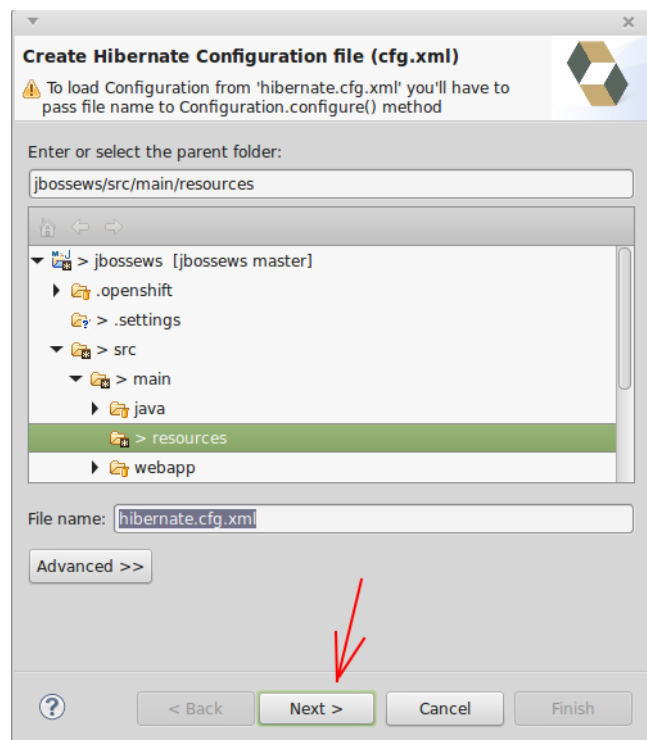


Figura 35: Crear fichero configuración Hibernate (cfg.xml)

No hace falta volver a escribir los datos de la BBDD se puede seleccionar en el link y los rellena automáticamente:

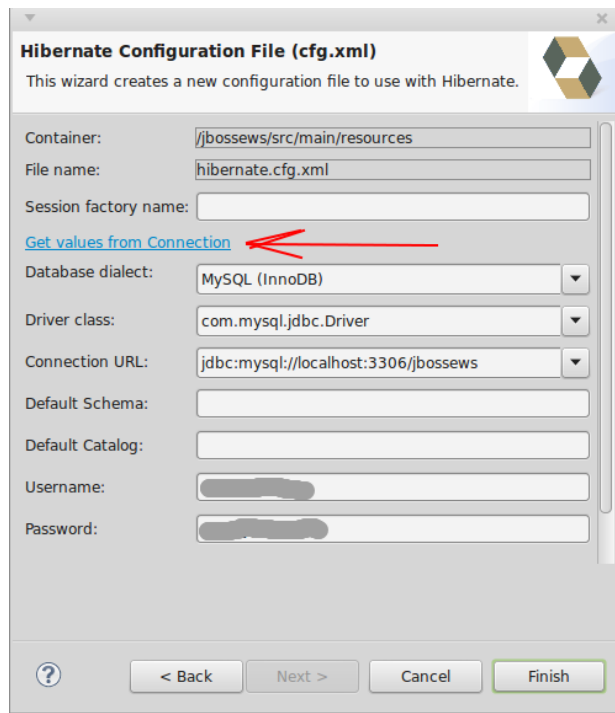


Figura 36: Coger los datos de la base de datos.

Con eso ya estaría hecha la configuración de hibernate. Ahora hay que realizar la 'Ingeniería inversa'

Le damos al botón de Hibernate y a la opción de 'Hibernate code Generation Configurations...'

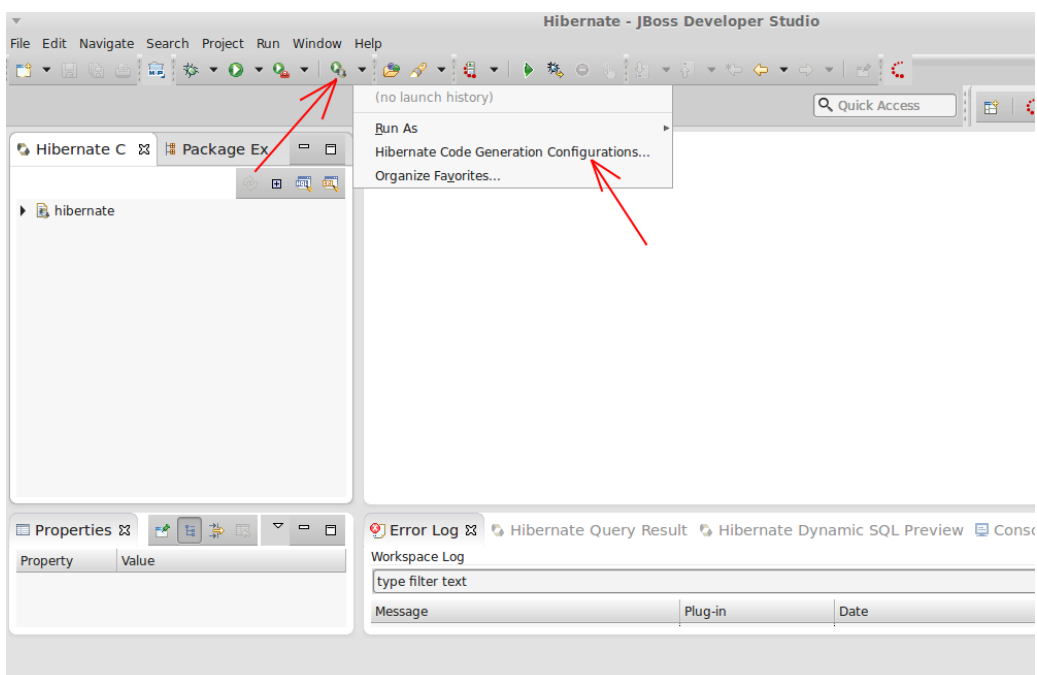


Figura 37: Generación de código a través de Hibernate.

Seleccionamos la configuración que hemos creado anteriormente. Hay que indicar la ruta donde se generarán las clases y el nombre del paquete java.

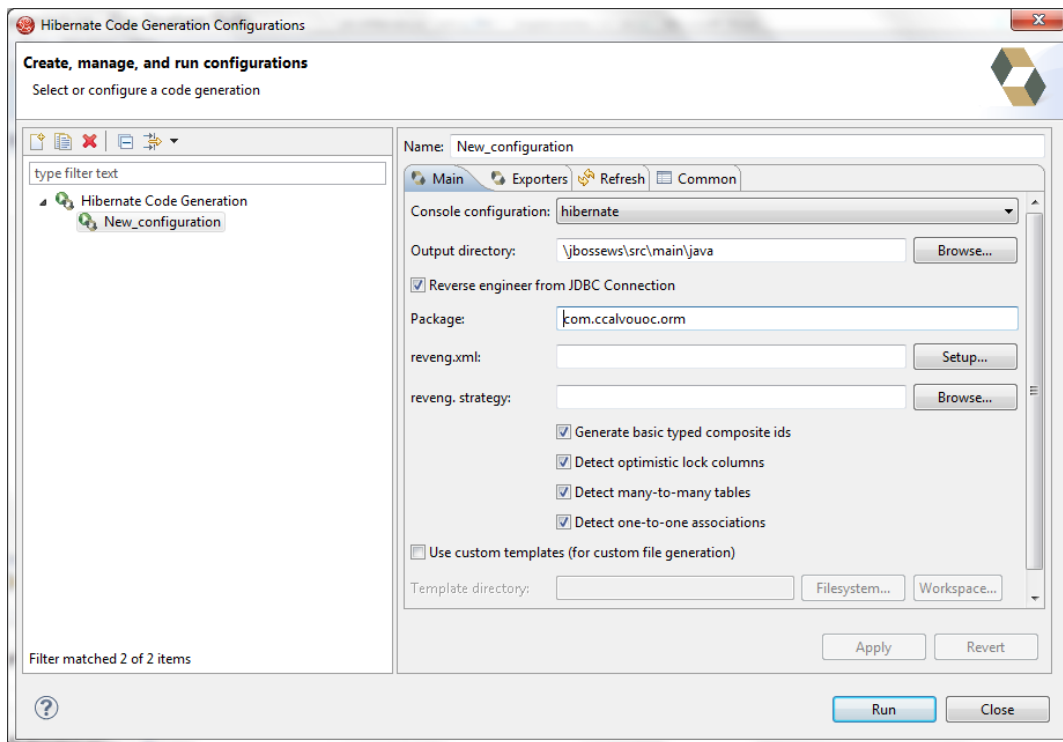


Figura 38: Configuración creacion de Clases a traves de Hibernate.

A continuación, en la pestaña 'Exporters', indicamos lo que queremos generar. Se deja la configuración como en la imagen y hacemos click en el botón 'run' para que genere las clases y los ficheros de configuración.

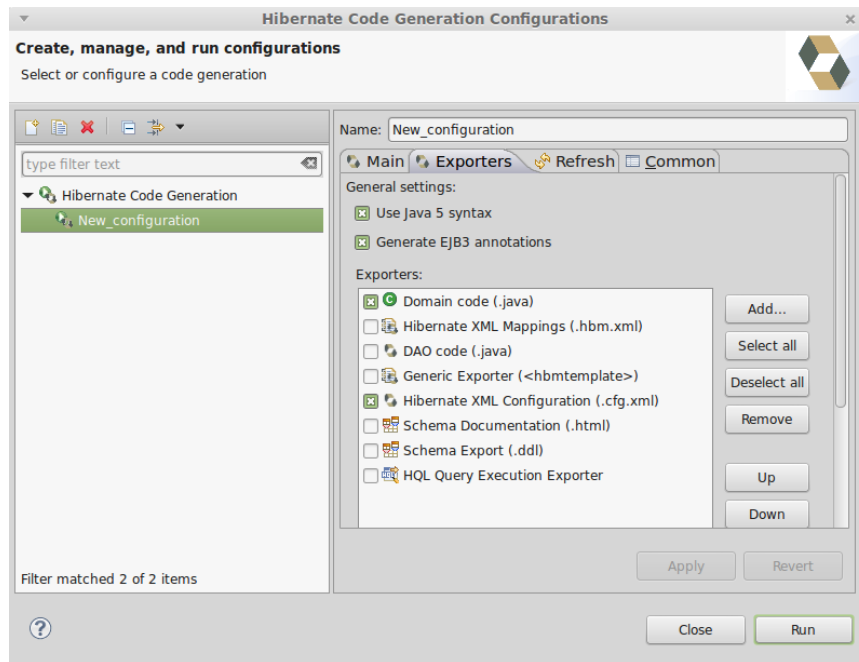


Figura 39: Ejecutar la generación de clases a través de Hibernate

Con eso ya podemos volver a la vista habitual de trabajo: j2ee y veremos que se han generado los ficheros de configuración en la carpeta resources así como un paquete con las clases que mapean las tablas.

Anexo 3. Código fuente (extractos)

Fichero de dependencias de Maven pom.xml

Maven es una herramienta de gestión de proyectos. Se basa en un fichero central, pom.xml, donde se define todo lo que necesita el proyecto. Maven maneja las dependencias del proyecto, compila, empaqueta y ejecuta los test. Una de las cosas más útiles, es el manejo de las dependencias, estas dependencias se definen en el fichero pom.xml y maven las descarga y las añade al classpath.

A continuación se muestra el contenido fichero pom.xml, con las dependencias necesarias para el proyecto.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>jbossews</groupId>
  <artifactId>jbossews</artifactId>
  <packaging>war</packaging>
  <version>1.0</version>
  <name>jbossews</name>
  <repositories>
    <repository>
      <id>eap</id>
      <url>http://maven.repository.redhat.com/techpreview/all</url>
      <releases>
        <enabled>>true</enabled>
      </releases>
      <snapshots>
        <enabled>>true</enabled>
      </snapshots>
    </repository>
  </repositories>
  <pluginRepositories>
    <pluginRepository>
      <id>eap</id>
      <url>http://maven.repository.redhat.com/techpreview/all</url>
      <releases>
        <enabled>>true</enabled>
      </releases>
      <snapshots>
        <enabled>>true</enabled>
      </snapshots>
    </pluginRepository>
  </pluginRepositories>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
```

```

    <maven.compiler.source>1.6</maven.compiler.source>
    <maven.compiler.target>1.6</maven.compiler.target>
</properties>
<dependencies>

    <dependency>
        <groupId>org.postgresql</groupId>
        <artifactId>postgresql</artifactId>
        <version>9.2-1003-jdbc4</version>
    </dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.25</version>
</dependency>

<dependency>
    <groupId>org.apache.struts</groupId>
    <artifactId>struts2-core</artifactId>
    <version>2.3.16.3</version>
</dependency>

<!-- <dependency> <groupId>javax.servlet</groupId>
<artifactId>javax.servlet-api</artifactId>
    <version>3.0.1</version> <scope>provided</scope> </dependency> -->

<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>2.5</version>
</dependency>

<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.1</version>
</dependency>

<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>3.5.6-Final</version>
</dependency>

<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-annotations</artifactId>
    <version>3.5.6-Final</version>
</dependency>

```



```

<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.7.7</version>
</dependency>

<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>

<dependency>
  <groupId>javax.mail</groupId>
  <artifactId>mail</artifactId>
  <version>1.4.7</version>
</dependency>

<dependency>
  <groupId>javax.activation</groupId>
  <artifactId>activation</artifactId>
  <version>1.1.1</version>
</dependency>

<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>3.8.1</version>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>com.jgeppert.struts2.bootstrap</groupId>
  <artifactId>struts2-bootstrap-plugin</artifactId>
  <version>2.0.0</version>
</dependency>
</dependencies>
<profiles>
  <profile>
    <!-- When built in OpenShift the 'openshift' profile will be used
when
        invoking mvn. -->
    <!-- Use this profile for any OpenShift specific customization your
app
        will need. -->
    <!-- By default that is to put the resulting archive into the
'webapps'

```

```

        folder. -->
        <!--
http://maven.apache.org/guides/mini/guide-building-for-different-environments.ht
ml -->
        <id>openshift</id>
        <build>
            <finalName>jbossews</finalName>
            <plugins>
                <plugin>
                    <artifactId>maven-war-plugin</artifactId>
                    <version>2.1.1</version>
                    <configuration>

<outputDirectory>webapps</outputDirectory>
                                <warName>ROOT</warName>
                                </configuration>
                </plugin>
            </plugins>
        </build>
    </profile>
</profiles>
</project>

```

Hibernate.cfg.xml

La forma de configurar hibernate es usando el fichero XML de configuración llamado hibernate.cfg.xml.

La información que contiene es la siguiente:

- Propiedades de configuración.
- Las clases que se quieren mapear.

El fichero, en nuestro proyecto, tiene la siguiente estructura:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <!--
        <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="hibernate.connection.password">b25CVnQ2VQhp</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/jbossews</property>
        <property name="hibernate.connection.username">adminN9Zkfeg</property>
        <property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>-->

```

```

    <property
name="hibernate.connection.datasource">java:comp/env/jdbc/MySQLDS</property>

    <mapping class="com.ccalvouoc.orm.Country" />
    <mapping class="com.ccalvouoc.orm.Location" />
    <mapping class="com.ccalvouoc.orm.ControlVisit" />
    <mapping class="com.ccalvouoc.orm.Coordinator" />
    <mapping class="com.ccalvouoc.orm.ProjectHasNews" />
    <mapping class="com.ccalvouoc.orm.VisitEpis" />
    <mapping class="com.ccalvouoc.orm.Manager" />
    <mapping class="com.ccalvouoc.orm.Project" />
    <mapping class="com.ccalvouoc.orm.Technician" />
    <mapping class="com.ccalvouoc.orm.User" />
    <mapping class="com.ccalvouoc.orm.VisitSignage" />
    <mapping class="com.ccalvouoc.orm.VwNews" />
    <mapping class="com.ccalvouoc.orm.Rol" />
    <mapping class="com.ccalvouoc.orm.VisitEpcs" />
    <mapping class="com.ccalvouoc.orm.Company" />
    <mapping class="com.ccalvouoc.orm.News" />
    <mapping class="com.ccalvouoc.orm.Status" />
    <mapping class="com.ccalvouoc.orm.City" />
    <mapping class="com.ccalvouoc.orm.Documents" />
    <mapping class="com.ccalvouoc.orm.SiteManager" />
    <mapping class="com.ccalvouoc.orm.Contractor" />
  </session-factory>
</hibernate-configuration>

```

Definición de acciones

Las acciones se definen en el fichero struts.xml, cada acción asocia una URL a una clase java con un método que se encarga de procesarla y en cada acción se define las páginas o acciones que puede tener como resultado.

```

...

    <!-- Muestra la pantalla para crear el nuevo proyecto,
         carga los datos de los posibles coordinadores y contratistas
         en las listas desplegables -->
    <action name="showNewProject"
class="com.ccalvouoc.actions.DoNewProjectAction" method="display">
    <result name="none">/newproject.jsp</result>
    </action>

```

```

        <!-- Acción que se ejecuta para crear el nuevo proyecto, recoge los datos
del formulario
        y realiza la inserción en la base de datos.
        SUCCESS: Si el resultado es correcto, muestra las lista de todos los
proyectos
        ERROR: Si ha habido errores al crear muestra la misma pantalla -->
        <action name="doNewProject" class="com.ccalvouoc.actions.DoNewProjectAction"
        method="execute">

        <result name="error" type="redirectAction">
            <param name="actionName">showNewProject</param>

        </result>
        <result name="success" type="redirectAction">
            <param name="actionName">listProjectaction</param>

        </result>

        </action>
        ...

```

Clase Action

Actions son clases encargadas de realizar la lógica para servir una petición. Cada URL es mapeada a una acción específica, la cual proporciona la lógica necesaria para servir a cada petición hecha por el usuario.

Estrictamente hablando, las acciones no necesitan implementar una interface o extender de alguna clase base. El único requisito para que una clase sea considerada un Action es que debe tener un método que no reciba argumentos que regrese ya sea un String o un objeto de tipo Result. Por default el nombre de este método debe ser "execute" aunque podemos ponerle el nombre que queramos y posteriormente indicarlo en el archivo de configuración de Struts.

Cuando el resultado es un String (lo cual es lo más común), el Result correspondiente se obtiene de la configuración del Action. Esto se usa para generar una respuesta para el usuario, lo cual veremos un poco más adelante.

Los Actions pueden ser objetos java simples (POJOs) que cumplan con el requisito anterior, aunque también pueden implementar la interface "com.opensymphony.xwork2.Action" o extender una clase base que proporciona Struts 2: "com.opensymphony.xwork2.ActionSupport" (lo cual nos hace más sencilla su creación y manejo).

DoNewProjectAction es la clase que maneja la petición de crear una nueva obra, en esta clase podemos ver que se han definido las propiedades para rellenar las listas desplegables al mostrar la

pantalla (listCoordinators, listContractors) y las propiedades que recogen los datos del formulario (title, coordinatorId, contractorId). Estas propiedades para que puedan ser utilizadas necesitan sus correspondientes métodos getter y setter. La lógica de negocio en este caso consiste en crear una nueva obra en la base de datos.

```

package com.ccalvouoc.actions;

import java.util.List;
import java.util.Map;

import com.ccalvouoc.dao.ContractorDao;
import com.ccalvouoc.dao.CoordinatorDao;
import com.ccalvouoc.dao.ProjectDao;
import com.ccalvouoc.orm.Contractor;
import com.ccalvouoc.orm.Coordinator;
import com.opensymphony.xwork2.ActionSupport;

public class DoNewProjectAction extends ActionSupport{
    //Variables para cargar las listas desplegables en pantalla
    private List<Coordinator> listCoordinators;
    private List<Contractor> listContractors;

    //Variables que recoge del formulario
    private String title;
    private String coordinatorId;
    private String contractorId;

    public DoNewProjectAction() {
        //rellena las listas de coordinadores y contratistas
        CoordinatorDao coordinatorDao = new CoordinatorDao();
        listCoordinators =coordinatorDao.listCoordinators();

        ContractorDao contractorDao = new ContractorDao();
        listContractors =contractorDao.listContractors();
    }

    @Override
    public String execute() throws Exception {
        //Crea el nuevo proyecto
        System.out.println(title +coordinatorId+contractorId);
        ProjectDao projectDao = new ProjectDao();
        String projectCode = projectDao.createNewProject(title,1,
            Integer.parseInt(coordinatorId) ,
            Integer.parseInt(contractorId));
        System.out.println("New Code"+projectCode);

        if (projectCode!=null){

```

```
        return ActionSupport.SUCCESS;

    }else{

        return ActionSupport.ERROR;

    }

}

public String display() {

    return NONE;

}

public List<Coordinator> getListCoordinators() {

    return listCoordinators;

}

public void setListCoordinators(List<Coordinator> listCoordinators) {

    this.listCoordinators = listCoordinators;

}

public List<Contractor> getListContractors() {

    return listContractors;

}

public void setListContractors(List<Contractor> listContractors) {

    this.listContractors = listContractors;

}

public String getTitle() {

    return title;

}

public void setTitle(String title) {

    this.title = title;

}

public String getCoordinatorId() {

    return coordinatorId;

}

public void setCoordinatorId(String coordinatorId) {

    this.coordinatorId = coordinatorId;

}

public String getContractorId() {

    return contractorId;

}

}
```

```

public void setContractorId(String contractorId) {
    this.contractorId = contractorId;
}

}

```

Clases Dao

La capa de persistencia está formada por las clases Dao, las cuales nos ofrecen un conjunto de métodos para consultar, agregar, modificar, eliminar o listar los objetos imágenes de los datos de la base de datos.

En este caso tenemos la clase ProjectDao, donde tenemos los siguientes métodos:

- listProjects : devuelve una lista de los proyectos
- getProject: recupera el proyecto que corresponda con el id que se pasa como parámetro.
- createNewProject: crea el nuevo proyecto.
- getNewProjectCode: es un método privado que utiliza el método createNewProject, nos devuelve el código (id) para el nuevo proyecto.
- getYearFormattedTwoDigits: método privado, devuelve el año actual en formato yy, para utilizarlo en el código de proyecto

```

package com.ccalvouoc.dao;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

import org.apache.log4j.Logger;
import org.hibernate.Criteria;
import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.AnnotationConfiguration;
import org.hibernate.criterion.Order;
import org.hibernate.criterion.Restrictions;

import com.ccalvouoc.orm.Contractor;
import com.ccalvouoc.orm.Coordinator;
import com.ccalvouoc.orm.News;
import com.ccalvouoc.orm.Project;
import com.ccalvouoc.orm.ProjectHasNews;

```

```

import com.ccalvouoc.orm.ProjectHasNewsId;
import com.ccalvouoc.orm.Technician;

public class ProjectDao {
    private static final Logger logger = Logger.getLogger(ProjectDao.class
        .getName());
    Session session;

    @SuppressWarnings("unchecked")
    public List<Project> listProjects() {
        List<Project> listProjects = null;
        try {
            SessionFactory sessionFactory = createSessionFactory();
            Session session = sessionFactory.openSession();

            Criteria criteria = session.createCriteria(Project.class);
            listProjects = criteria.list();

        } catch (HibernateException e) {
            logger.error(e.getMessage(), e);
            session.getTransaction().rollback();
        } catch (Exception e) {
            logger.error(e.getMessage(), e);
            session.getTransaction().rollback();
        } finally {
            if (session != null && session.isOpen())
                session.close();
        }
        return listProjects;
    }

    @SuppressWarnings("unchecked")
    public Project getProject(String code) {
        Project project = null;
        try {
            SessionFactory sessionFactory = createSessionFactory();
            Session session = sessionFactory.openSession();

            Criteria criteria = session.createCriteria(Project.class);
            System.out.println("criteria.project.." + criteria);
            criteria.add(Restrictions.eq("id", code));
            project = (Project) criteria.list().get(0);

            /*
             * String hql = "FROM Project"; Query query =
             * session.createQuery(hql); listaNames = query.list();
             * session.getTransaction().commit();
             */
        } catch (HibernateException e) {
            logger.error(e.getMessage(), e);
            session.getTransaction().rollback();
        }
    }
}

```



```

    } catch (Exception e) {
        logger.error(e.getMessage(), e);
        session.getTransaction().rollback();
    } finally {
        if (session != null && session.isOpen())
            session.close();
    }
    return project;
}

public String createNewProject(String title, int technicianId,
    int coordinatorId, int contractorId) {

    // logger.warn("*** EJECUTANDO addActividad() ***");
    String projectCode = getNewProjectCode();
    try {

        SessionFactory sessionFactory = createSessionFactory();
        Session session = sessionFactory.openSession();
        session.beginTransaction();

        Project project = new Project();
        project.setId(projectCode);
        project.setTitle(title);

        //recupero el tecnico, el contratista y el coordinador de la obra
        TechnicianDao technicianDao = new TechnicianDao();
        ContractorDao contractorDao = new ContractorDao();
        CoordinatorDao coordinatorDao = new CoordinatorDao();

        Technician technician = technicianDao.getTechnician(technicianId);
        Contractor contractor = contractorDao.getContractor(contractorId);
        Coordinator coordinator = coordinatorDao
            .getCoordinator(coordinatorId);

        // asigno el tecnico, el contratista y el coordinador a la obra
        project.setTechnician(technician);
        project.setContractor(contractor);
        project.setCoordinator(coordinator);

        // añado a la obra la noticia " Nueva obra"
        NewsDao newsDao = new NewsDao();
        News newsNewProject = newsDao.getNews(1);

        ProjectHasNews projectHasNews = new ProjectHasNews();
        ProjectHasNewsId projectHasNewsId= new
ProjectHasNewsId(project.getId(), newsNewProject.getId());
        projectHasNews.setId(projectHasNewsId);

```

```

        projectHasNews.setNews(newsNewProject);
        projectHasNews.setDateNews(new Date());

        Set<ProjectHasNews> listProjectHasNews = new
HashSet<ProjectHasNews>();
        listProjectHasNews.add(projectHasNews);

        project.setProjectHasNewses(listProjectHasNews);

System.out.println(">>>getProjectHasNewses>>>" + project.getProjectHasNewses().s
ize());

        session.save(project);
        session.save(projectHasNews);

        session.getTransaction().commit();

    } catch (HibernateException e) {
        logger.error(e.getMessage(), e);
        session.getTransaction().rollback();
    } catch (Exception e) {
        logger.error(e.getMessage(), e);
        session.getTransaction().rollback();
        } finally {
            if (session != null && session.isOpen())
                session.close();
        }

    return projectCode;
}

/*Generamos el nuevo codigo con formato OByy-999
 * yy: año
 * 999: numero secuencial
 * */
private String getNewProjectCode() {

    int numProject = 1;
    String newCodeProject = null;
    String newCodeProjectPrefix = "OB" + getYearFormatedTwoDigits() + "-";

    try {
        SessionFactory sessionFactory = createSessionFactory();
        Session session = sessionFactory.openSession();

        Criteria criteria = session.createCriteria(Project.class);
        System.out.println("criteria.project.." + criteria);
        criteria.add(Restrictions.like("id", newCodeProjectPrefix + "%"));
        criteria.addOrder(Order.desc("id"));
        if (criteria.list().size() != 0) {

```

```

        Project lastProject = (Project) criteria.list().get(0);
        System.out.println("numsss ultimo proyecto"
            + lastProject.getId());
        numProject =
Integer.parseInt(lastProject.getId().substring(5,
            8)) + 1;
        System.out.println("numero ultimo proyecto" + numProject);

    }

    newCodeProject = newCodeProjectPrefix
        + String.format("%03d", numProject);

} catch (HibernateException e) {
    logger.error(e.getMessage(), e);
    session.getTransaction().rollback();
} catch (Exception e) {
    logger.error(e.getMessage(), e);
    session.getTransaction().rollback();
} finally {
    if (session != null && session.isOpen())
        session.close();
}

return newCodeProject;
}

private String getYearFormattedTwoDigits() {
    //recupera el año actual en formato yy
    SimpleDateFormat sdf = new SimpleDateFormat("yy"); // Just the year,

        // with 2 digits
    String formattedDate = sdf.format(Calendar.getInstance().getTime());
    return formattedDate;
}

public static SessionFactory createSessionFactory() {
    SessionFactory sessionFactory = new AnnotationConfiguration()
        .configure().buildSessionFactory();
    return sessionFactory;
}
}

```

Mapeos de la base de datos

Las clases Dao utilizan las clases que mapean las tablas de la base de datos, estos mapeos son creados a través por Hibernate a través de ingeniería inversa.

A continuación se muestra el mapeo correspondiente a la tabla **project** de la base de datos, mediante anotaciones se identifica a que tabla pertenece, las columnas, la clave primaria y las relaciones.

```

package com.ccalvouoc.orm;

// Generated 27-nov-2014 20:26:52 by Hibernate Tools 3.4.0.CR1

import java.util.Date;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 * Project generated by hbm2java
 */
@Entity
@Table(name = "project", catalog = "jbosssews")
public class Project implements java.io.Serializable {

    private String id;
    private Contractor contractor;
    private Manager manager;
    private SiteManager siteManager;
    private Coordinator coordinator;
    private Technician technician;
    private Date dateStart;
    private String title;
    private String road;
    private String pk;
    private Integer months;
    private Float budget;
    private Documents documents;
    private Set<ControlVisit> controlVisits = new HashSet<ControlVisit>(0);
    private Set<ProjectHasNews> projectHasNewses = new HashSet<ProjectHasNews>(
        0);

```

```

public Project() {
}

public Project(String id) {
    this.id = id;
}

public Project(String id, Contractor contractor, Manager manager,
                SiteManager siteManager, Coordinator coordinator,
                Technician technician, Date dateStart, String title, String road,
                String pk, Integer months, Float budget, Documents documents,
                Set<ControlVisit> controlVisits,
                Set<ProjectHasNews> projectHasNewses) {
    this.id = id;
    this.contractor = contractor;
    this.manager = manager;
    this.siteManager = siteManager;
    this.coordinator = coordinator;
    this.technician = technician;
    this.dateStart = dateStart;
    this.title = title;
    this.road = road;
    this.pk = pk;
    this.months = months;
    this.budget = budget;
    this.documents = documents;
    this.controlVisits = controlVisits;
    this.projectHasNewses = projectHasNewses;
}

@Id
@Column(name = "ID", unique = true, nullable = false, length = 8)
public String getId() {
    return this.id;
}

public void setId(String id) {
    this.id = id;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "contractor_ID")
public Contractor getContractor() {
    return this.contractor;
}

public void setContractor(Contractor contractor) {
    this.contractor = contractor;
}

@ManyToOne(fetch = FetchType.LAZY)

```

```

@JoinColumn(name = "manager_ID")
public Manager getManager() {
    return this.manager;
}

public void setManager(Manager manager) {
    this.manager = manager;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "site_manager_ID")
public SiteManager getSiteManager() {
    return this.siteManager;
}

public void setSiteManager(SiteManager siteManager) {
    this.siteManager = siteManager;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "coordinator_ID")
public Coordinator getCoordinator() {
    return this.coordinator;
}

public void setCoordinator(Coordinator coordinator) {
    this.coordinator = coordinator;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "technician_ID")
public Technician getTechnician() {
    return this.technician;
}

public void setTechnician(Technician technician) {
    this.technician = technician;
}

@Temporal(TemporalType.TIMESTAMP)
@Column(name = "date_start", length = 19)
public Date getDateStart() {
    return this.dateStart;
}

public void setDateStart(Date dateStart) {
    this.dateStart = dateStart;
}

@Column(name = "title", length = 100)
public String getTitle() {

```

```
        return this.title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    @Column(name = "road", length = 45)
    public String getRoad() {
        return this.road;
    }

    public void setRoad(String road) {
        this.road = road;
    }

    @Column(name = "PK", length = 45)
    public String getPk() {
        return this.pk;
    }

    public void setPk(String pk) {
        this.pk = pk;
    }

    @Column(name = "months")
    public Integer getMonths() {
        return this.months;
    }

    public void setMonths(Integer months) {
        this.months = months;
    }

    @Column(name = "budget", precision = 12, scale = 0)
    public Float getBudget() {
        return this.budget;
    }

    public void setBudget(Float budget) {
        this.budget = budget;
    }

    @OneToOne(fetch = FetchType.LAZY, mappedBy = "project")
    public Documents getDocuments() {
        return this.documents;
    }

    public void setDocuments(Documents documents) {
        this.documents = documents;
    }
}
```

```

@OneToMany(fetch = FetchType.LAZY, mappedBy = "project")
public Set<ControlVisit> getControlVisits() {
    return this.controlVisits;
}

public void setControlVisits(Set<ControlVisit> controlVisits) {
    this.controlVisits = controlVisits;
}

@OneToMany(fetch = FetchType.LAZY, mappedBy = "project")
public Set<ProjectHasNews> getProjectHasNews() {
    return this.projectHasNews;
}

public void setProjectHasNews(Set<ProjectHasNews> projectHasNews) {
    this.projectHasNews = projectHasNews;
}
}

```

Grabación de ficheros

Para subir un fichero al servidor, no se carga el fichero directamente en la BBDD, sino que se graba en el sistema de archivos. También se graba un registro en la tabla Files, con el nombre, ruta, fecha de grabación, etc.

El código que graba el fichero en el disco, se encuentra en la clase `com.ccalvouoc.utils.FileUtils`. Concretamente en el método `saveFileOnDisk(...)`

Básicamente se le pasa un objeto java de tipo `File` que representa al fichero, el nombre del mismo y la ruta donde lo vamos a grabar. Usamos objetos del tipo `FileInputStream` y `FileOutputStream` para recorrer con un bucle todos los bytes de fichero y escribirlos en el disco.

```

    public static void saveFileOnDisk(File file, String fileName, String
fileDirectory) throws IOException{
        FileInputStream in = null;
        FileOutputStream out = null;

        File dir = new File (fileDirectory);
        if ( !dir.exists() )
            dir.mkdirs();

        String targetPath = dir.getPath() + File.separator + fileName;
        File destinationFile = new File ( targetPath);
        try {
            in = new FileInputStream( file );
            out = new FileOutputStream( destinationFile );
            int c;

```



```

        while ((c = in.read()) != -1) {
            out.write(c);
        }

    }finally {
        if (in != null) {
            in.close();
        }
        if (out != null) {
            out.close();
        }
    }
}

```

Subir imágenes

Subir imágenes al servidor es igual que subir cualquier fichero. La única peculiaridad, es que al grabar la imagen original en el disco, también grabamos un thumbnail de la imagen. Un Thumbnail es una representación de la imagen pero con una resolución mucho menor. Con eso se crean las imágenes más pequeñas, pero más ligeras que se muestran en los álbumes de fotos.

El usuario puede hacer click en un thumbnail, y entonces se descarga la imagen original.

Para crear un thumbnail con java se puede ver el siguiente método de la clase `com.ccalvouoc.util.FileUtils`

```

public static void saveThumbnail(File imagefile, String fileName, String
fileDirectory) throws Exception{

    try {
        BufferedImage originalImage = ImageIO.read(imagefile);
        BufferedImage resizeImageJpg = resizeImage(originalImage);
        File dir = new File (fileDirectory);
        if ( !dir.exists() )
            dir.mkdirs();
        String targetPath = dir.getPath() + File.separator + fileName;
        ImageIO.write(resizeImageJpg, "jpg", new File(targetPath));
    }catch(Exception ex) {
        logger.error(ex.getMessage(), ex);
        throw ex;
    }

}

```

Anexo 4. Capturas de pantalla

4.1 Login



Protocolo de Seguridad y Salud

 Acceso

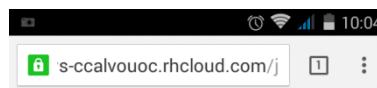
Usuario

Password


Aceptar



Figura 40: Captura PC. Login



Protocolo de Seguridad y Salud

 Acceso

Usuario

Password

Aceptar

Figura 41: Captura Móvil. Login

4.2 Inicio

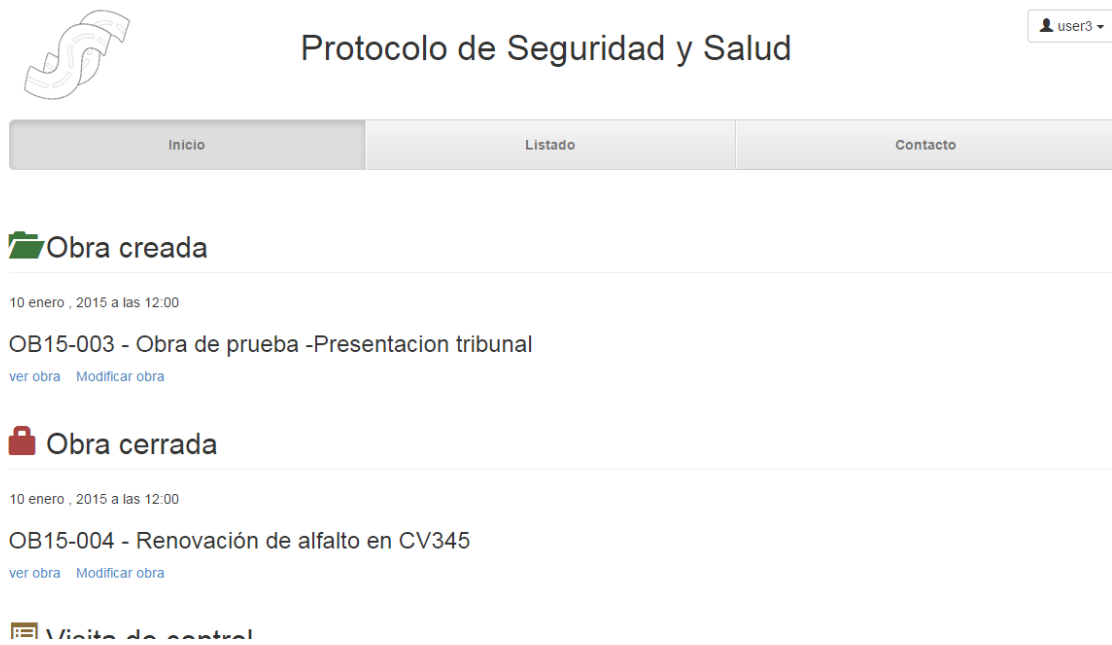


Figura 42: Captura PC.Inicio



Figura 43: Captura Móvil. Inicio

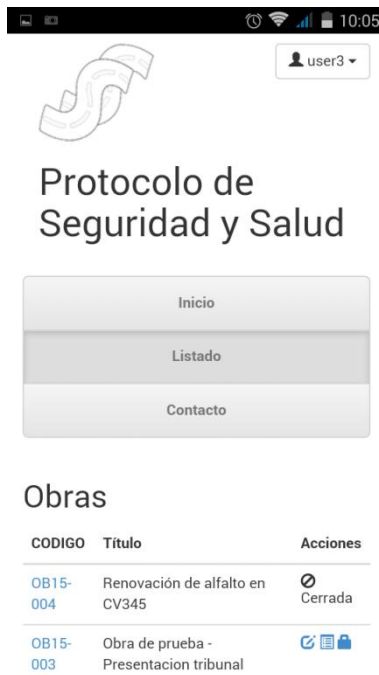
4.3 Listado de obras



The screenshot shows a web application interface for 'Protocolo de Seguridad y Salud'. At the top left is a logo consisting of three interlocking loops. To the right of the logo is the title 'Protocolo de Seguridad y Salud' and a user profile dropdown menu labeled 'user3'. Below the title is a navigation bar with three buttons: 'Inicio', 'Listado', and 'Contacto'. The 'Listado' button is highlighted. Underneath the navigation bar is the heading 'Obras' followed by a table listing various works.

CODIGO	Título	Carretera	PK	Duración	Coordinador de Seguridad y Salud	Jefe de Obra	Acciones
OB15-004	Renovación de alfalto en CV345	CV34	12	11	Anna Moral Feijoo	Joan P	🔒 Cerrada
OB15-003	Obra de prueba -Presentacion tribunal	CV23	23	12	Anna Moral Feijoo		🔗 📄 🔒
OB15-002	Obra presentación				Anna Moral Feijoo		🔗 📄 🔒
OB15-001	Adecuación señalización adaptativa en carreteras convencionales				Anna Moral Feijoo		🔗 📄 🔒
OB14-004	Limpeza de cunetas				Anna Moral Feijoo		🔗 📄 🔒
OB14-003	Reposición del firme	CV1234	23	6	Anna Moral Feijoo		🔗 📄 🔒
OB14-001	Mejora de una carretera	BV2249	25	10	Anna Moral Feijoo	Pau Vidal Albar	🔗 📄 🔒

Figura 44: Captura PC. Listado



The screenshot shows the mobile version of the 'Protocolo de Seguridad y Salud' application. At the top, there is a status bar with icons for signal, Wi-Fi, and battery, and the time 10:05. Below the status bar is the same logo and title 'Protocolo de Seguridad y Salud' as seen in the PC version, along with the 'user3' user profile. A navigation bar with 'Inicio', 'Listado', and 'Contacto' buttons is present, with 'Listado' being the active tab. Below the navigation bar is the heading 'Obras' and a table showing a subset of the work list.

CODIGO	Título	Acciones
OB15-004	Renovación de alfalto en CV345	🔒 Cerrada
OB15-003	Obra de prueba - Presentacion tribunal	🔗 📄 🔒

Figura 45: Captura Móvil. Listado

4.4 contacto



The screenshot shows the PC interface of the 'Protocolo de Seguridad y Salud' application. At the top left is a logo consisting of three interlocking loops. To its right is the title 'Protocolo de Seguridad y Salud' and a user profile dropdown menu labeled 'user3'. Below this is a horizontal navigation bar with three tabs: 'Inicio', 'Listado', and 'Contacto', with 'Contacto' being the active tab. The main content area is titled 'Formulario de Contacto' and contains two input fields: 'Asunto' (a single-line text box) and 'Comentario*' (a larger multi-line text area). A blue 'Aceptar' button is located at the bottom left of the form.

Figura 46: Captura PC. Contacto



The screenshot shows the mobile version of the application. At the top is a black status bar with icons for mail, camera, and other functions, along with the time '10:05'. Below the status bar is the same logo and title 'Protocolo de Seguridad y Salud'. The navigation bar has three tabs: 'Inicio', 'Listado', and 'Contacto', with 'Contacto' selected. The 'Formulario de Contacto' section is vertically oriented and contains the 'Asunto' and 'Comentario*' input fields, with the 'Aceptar' button positioned at the bottom.

Figura 47: Captura Móvil. Contacto

4.5 Nueva obra

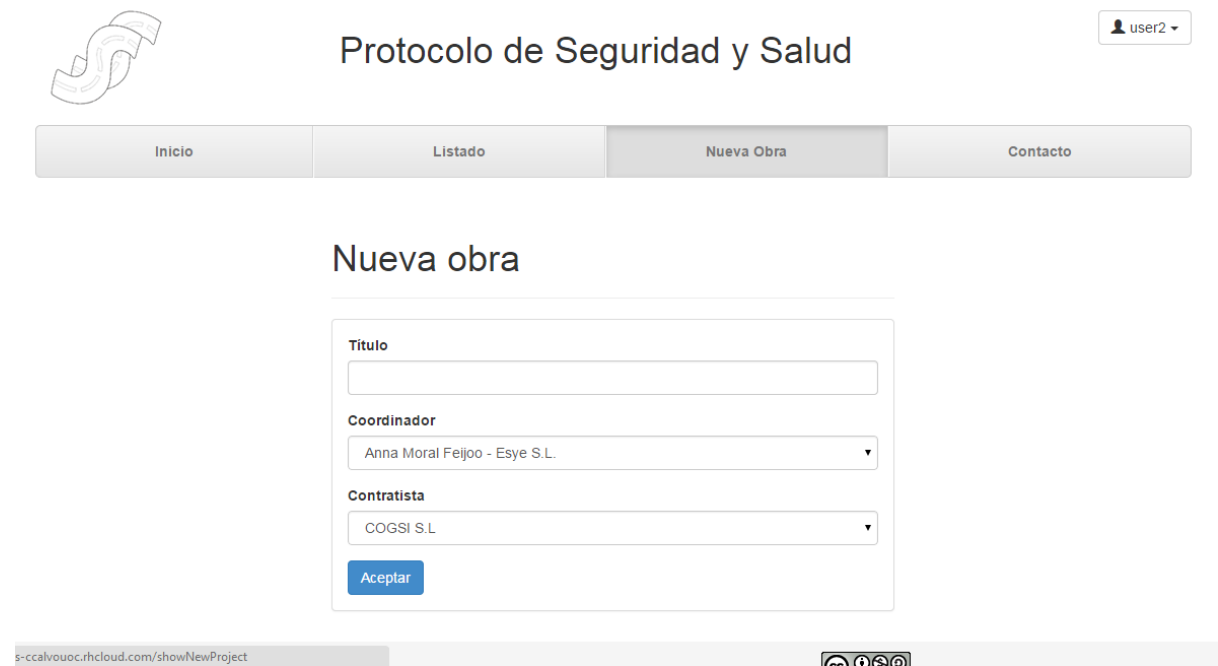


Figura 48: Captura PC. Nueva Obra.



Figura 49: Captura Móvil. Nueva Obra

4.6 Editar obra



Protocolo de Seguridad y Salud user3

Inicio Listado Contacto

OB14-001 - Mejora de una carretera

Técnico	Coordinador de Seguridad y Salud	Contratista
Joan Rueda Solé - Diputación de Barcelona ☎ 637223344 ✉ crisavis@uoc.edu	Anna Moral Feijoo - Esye S.L. ☎ 648334455 ✉ crisavis@uoc.edu	COGSI S.L. ☎ 935555555 ✉ crisavis@uoc.edu JEFE DE OBRA: Pau Vidal Albar ☎ 939898989

Datos Generales

Carretera	P.K.	Presupuesto	Fecha de inicio	Duración aprox (meses)
BV2249	25	10000.0	25/12/2013	10

Jefe de Obra

Nif	Nombre	eMail	Teléfono
44444444A	Pau Vidal Albar	crisavis@uoc.edu	939898989

Figura 50: Captura PC. Editar Obra



☎ 648334455 ✉ crisavis@uoc.edu

Contratista

COGSI S.L.
☎ 935555555 ✉ crisavis@uoc.edu
JEFE DE OBRA:
☎

Datos Generales

Carretera

P.K.

Presupuesto

Fecha de inicio

Figura 51: Captura Móvil. Editar Obra

4.7 Ver obra



user3 ▾

Inicio
Listado
Contacto

OB15-004 Renovación de alfalto en CV345

Técnico	Coordinador de Seguridad y Salud	Contratista
Joan Rueda Solé - Diputación de Barcelona ☎ 637223344 ✉ crisavis@uoc.edu	Anna Moral Feijoo - Eseye S.L. ☎ 648334455 ✉ crisavis@uoc.edu	COGSI S.L. ☎ 935555555 ✉ crisavis@uoc.edu JEFE DE OBRA: Joan P ☎ 679565656

Datos Generales

Carretera: CV34 Inicio: 10/01/15 0:00:00.000	P.K: 12 Duración aprox (meses): 11	Presupuesto: 120000.0
---	---	------------------------------

Documentación

Aviso Previo: TC2 Trabajadores: Entrega de EPIs: Aprobación del plan de seguridad: Apertura del centro de trabajo:	Libro de subcontrataciones: Plan de Obra: Plan de obra.txt ITV Vehiculos: ITV.txt Hoja de Hospitales y Emergencias: Otros:
---	---

Figura 52: Captura PC. Ver Obra



Contacto

OB15-004 Renovación de alfalto en CV345

Técnico Joan Rueda Solé - Diputación de Barcelona ☎ 637223344 ✉ crisavis@uoc.edu
Coordinador de Seguridad y Salud Anna Moral Feijoo - Eseye S.L. ☎ 648334455 ✉ crisavis@uoc.edu
Contratista COGSI S.L. ☎ 935555555 ✉ crisavis@uoc.edu JEFE DE OBRA: Joan P ☎ 679565656

Figura 53: Captura Móvil. Ver Obra

4.8 Crear visita de obra



Figura 54: Captura PC. Crear Visita de Obra



Figura 55: Captura Móvil. Crear Visita de Obra

4.9 Visita de obra



Figura 56: Captura PC. Visita de obra



Figura 57: Captura Móvil. Visita de obra

4.10 Editar visita de obra



Figura 58: Captura PC. Editar Visita de Obra

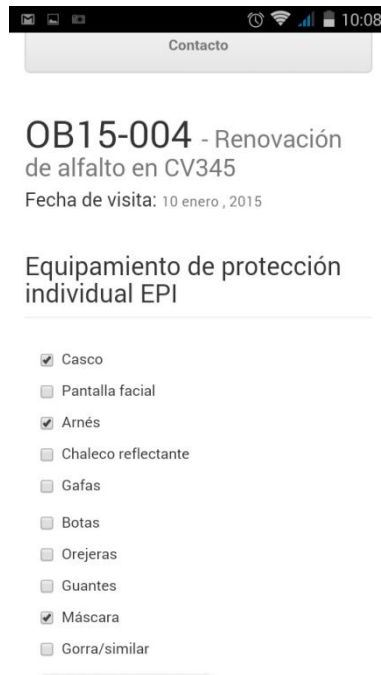


Figura 59: Captura Móvil. Editar Visita de Obra

4.11 Cerrar obra



The screenshot shows a web application interface. At the top left is a logo consisting of three interlocking 'S' shapes. To its right is the title 'Protocolo de Seguridad y Salud'. In the top right corner, there is a user profile icon labeled 'user3'. Below the title is a horizontal navigation bar with three buttons: 'Inicio', 'Listado', and 'Contacto'. The main content area displays 'OB14-003 - Reposición del firme'. Underneath is the section 'Cierre de Obra'. It contains a 'Fecha de cierre' field with the value '12/01/2015' and an 'Observaciones' text area. A blue 'Aceptar' button is located at the bottom right of the form.

Figura 60: Captura PC. Cerrar Obra



The screenshot shows the mobile version of the same web application. The top status bar displays 'Listado' and 'Contacto' as navigation options. The main content area shows 'OB14-003 - Reposición del firme' and 'Cierre de Obra'. The 'Fecha de cierre' field is a dropdown menu showing '12/01/2015'. The 'Observaciones' text area is present. A blue 'Aceptar' button is at the bottom right. At the very bottom, there is a Creative Commons license logo (CC BY-NC-SA).

Figura 61: Captura Móvil. Cierre de Obra

4.11 Perfil de usuario

Protocolo de Seguridad y Salud user3

Inicio Listado Contacto

Perfil

Datos Personales

NIF: 33333333P nombre: Anna Moral Feijoo Teléfono: 648334455 email: crisavis@uoc.edu

Datos Empresa

Cif: B11111111 Nombre: Esey S.L. Teléfono: 933333333 email: crisavis@uoc.edu

[Guardar](#)

Modificar contraseña

Contraseña Actual Nueva contraseña Repetir Nueva contraseña [Modificar contraseña](#)

Figura 62: Captura PC. Perfil de usuario

Contacto

Perfil

Datos Personales

NIF: 33333333P nombre: Anna Moral Feijoo Teléfono: 648334455 email: crisavis@uoc.edu

Datos Empresa

Figura 63: Captura Móvil. Perfil de usuario

Anexo 5. Guía de usuario

Acceder a la plataforma

Introduciendo la dirección <https://jbossews-ccalvouoc.rhcloud.com> en su navegador habitual, puede acceder a la página de acceso de la plataforma, introduzca sus datos de acceso y pulse Aceptar.



Protocolo de Seguridad y Salud

Figura 64: Guía de usuario - Acceso a la plataforma

Salir de la plataforma

Acceda al menú de usuario y pulse la opción *Salir*.

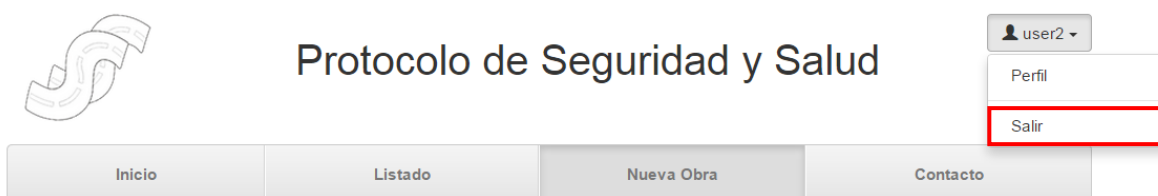


Figura 65: Guía de usuario - Salir de la plataforma

Modificar datos de usuario

Acceda al menú de usuario y pulse la opción perfil.

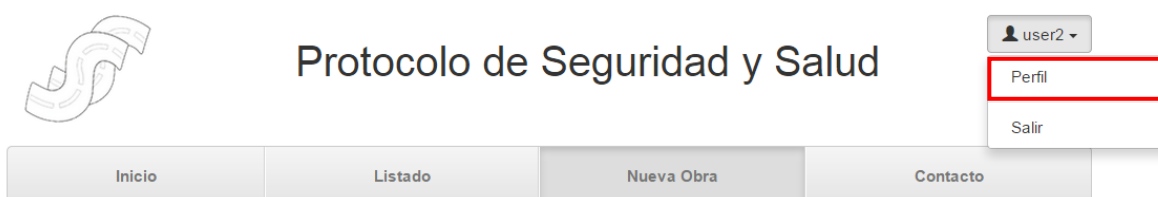


Figura 66: Guía de usuario – Acceso a Perfil

Accederá al perfil donde podrá modificar sus datos personales y contraseña.

Figura 67: Guía de usuario - Modificar perfil

Actualizaciones de obra

A través de *Menu>Inicio* puede ver las últimas actualizaciones de las obras, esta ventana es la primera que se ve al entrar en la plataforma.

Figura 68: Guía de usuario - Últimas actualizaciones

Listado de obras

A través de *Menu>Listado* puede visualizar el listado de obras, en las cuales esta incluido. Pulse en el código de obra para ir al detalle de la misma.

CODIGO	Título	Carretera	PK	Duración	Coordinador de Seguridad y Salud	Jefe de Obra
OB15-001	Adecuación señalización adaptativa en carreteras convencionales				Anna Moral Feijoo	
OB14-004	Limpieza de cunetas				Anna Moral Feijoo	
OB14-003	Reposición del firme	CV1234	23	8	Anna Moral Feijoo	
OB14-001	Mejora de una carretera	BV2249	25	10	Anna Moral Feijoo	Pau Vidal Albar
OB14-002	Rehabilitación del Firme con Mezclas Bituminosas en Caliente.				Anna Moral Feijoo	

Figura 69: Guía de usuario - Listado de obras

Detalle de obra

A través de *Menú>Inicio* pulsando el enlace ver obra correspondiente.

A través de *Menú>listado* pulsando en el código de obra.

OB14-001 Mejora de una carretera

Técnico	Coordinador de Seguridad y Salud	Contratista
Joan Rueda Solé - Diputación de Barcelona ☎ 637223344 ✉ crisavis@uoc.edu	Anna Moral Feijoo - Esye S.L. ☎ 648334455 ✉ crisavis@uoc.edu	COGSI S.L. ☎ 935555555 ✉ crisavis@uoc.edu JEFE DE OBRA: Pau Vidal Albar ☎ 939999999

Datos Generales

Carretera: BV2249 Inicio: 25/12/13 0:00:00.000	P.K: 25 Duración aprox (meses): 10	Presupuesto: 100000.0
---	---------------------------------------	-----------------------

Documentación

Aviso Previo: DocumentoPruebaAttach.txt	Libro de subcontrataciones:
TC2 Trabajadores:	Plan de Obra:
Entrega de EPIS:	ITV Vehiculos:
Aprobación del plan de seguridad:	Hoja de Hospitales y Emergencias:
Actas del comité de trabajo:	...

Figura 70: Guía de usuario - Detalle de obra

Visita de obra

Desde el *Detalle de obra* puede ver el listado de visitas de control, pulse en el enlace *ver* para ver el detalle de la visita seleccionada.

Visitas de control

Fecha	Estado EPI	Estado EPC	Señalización	Ver	Modificar
14/10/2014	Básico	Básico	Básico	ver	modificar

Figura 71: Guía de usuario - Ver visita de obra

Contacto con la plataforma

A través de Menú>Contacto accede al formulario para ponerse en contacto con la plataforma, aquí puede enviar sus sugerencias, dudas, quejas e informarnos acerca de incidencias, nos pondremos en contacto con usted lo antes posible.

Figura 72: Guía de usuario - Contacto

Crear obra (sólo Técnicos)

A través de Menú>Nueva Obra puede acceder al formulario de creación de obra, Introduzca un Título para la obra y seleccione un coordinador de obra, una empresa contratista y pulse Aceptar, la obra se generará y se le asignará un código. El formulario le redirigirá a la pantalla de listado de obras donde la obra se verá en primera posición. El coordinador de obra recibirá un mail para que acceda a la obra e introduzca el resto de datos.

Figura 73: Guía de usuario - Crear nueva obra

Editar obra (sólo Coordinador de Seguridad y Salud)

Existen dos formas de acceder a la modificación de la obra


1. A través del Listado de obras pulsando el icono  de la obra que desea modificar
2. A través de Menú>Inicio pulsando el enlace Modificar obra

Figura 74: Guía de usuario - Editar obra

Añadir documentación a la obra (sólo Coordinador de Seguridad y Salud)


Una vez dentro de la pantalla *Modificación de obra*, para añadir un documento a la obra pulse el botón seleccionar archivo de la documentación que quiera añadir y seleccione el documento. Tenga en cuenta que un fichero demasiado pesado podría ralentizar demasiado el proceso de grabación.

Documentación

Aviso Previo: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	TC2 Trabajadores: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	Entrega de EPIs: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado
Aprobación del plan de seguridad: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	Apertura del centro de trabajo: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	Libro incidencias: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado
Libro de subcontrataciones: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	Plan de Obra: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	ITV Vehiculos: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado
Hoja de Hospitales y Emergencias: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	Otros: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	Otros: <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado

Figura 75: Guía de usuario - Añadir documentación

Crear visita de control (sólo Coordinador de Seguridad y Salud)

Para crear una nueva visita de control, desde el Listado de obras, pulse el icono  de la obra a la cual quiere añadir una visita de control. Accederá a un formulario donde debe introducir una fecha y pulse el botón *Crear nueva visita*, la visita se creara y accederá a la ventana de edición de la Visita de Control.



Protocolo de Seguridad y Salud

user3 -

Inicio
Listado
Contacto

OB14-004 - Limpieza de cunetas

Nueva visita de control

Fecha de cierre

Figura 76: Guía de usuario - Crear visita de control

Editar visita de control (sólo Coordinador de Seguridad y Salud)

Dentro del Detalle de la obra puede observar el listado de visitas de control, pulse el enlace *Modificar* de la visita que quiera modificar y accederá a la ventana de Edición de Visita de Control.



Figura 77: Guía de usuario - Editar visita de control

Añadir fotos a la visita de control (sólo Coordinador de Seguridad y Salud)

Para añadir imágenes a la visita de control, pulse el botón que hay debajo de la imagen que quiere añadir y seleccione la imagen, la imagen se añadirá en el recuadro.



Figura 78: Guía de usuario - Añadir fotografías a visita de control

Borrar Visita de control (sólo Coordinador de Seguridad y Salud)

Las visitas de control únicamente pueden ser borradas y editadas, sino han sido enviadas. Para borrarla pulse el botón borrar que hay en la ventana de Edición de Visita de Control.

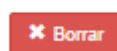


Figura 79: Guía de usuario - Borrar visita de control


Enviar visita de control (sólo Coordinador de Seguridad y Salud)

Para enviar la visita de control a todos los intervinientes, pulse el botón enviar que hay en la ventana de Edición de Visita de Control. Una vez enviada la visita esta no puede ser modificada.



Figura 80: Guía de usuario - Enviar visita de control

Cerrar Obra (sólo Coordinador de Seguridad y Salud)

Para cerrar la obra, desde el Listado de obras, pulse el icono  de la obra que desea cerrar, le aparecerá el formulario de cierre donde tiene que introducir la fecha, y un resumen del curso de la obra. Las obras cerradas no podrán ser editadas, ni volverse a abrir.

Protocolo de Seguridad y Salud user3 ▾

Inicio Listado Contacto

OB14-004 - Limpieza de cunetas

Cierre de Obra

Fecha de cierre

Observaciones

Aceptar

Figura 81: Guía de usuario - Cerrar obra

Anexo 6. Glosario

Data center: Es un centro de datos almacenados, tratados y distribuidos para su consulta. Los servidores que contienen todos esos datos se mantienen en un entorno seguro y con un correcto funcionamiento.

BBDD: Conjunto de datos relacionados que se almacenan de forma que se pueda acceder a ellos de manera sencilla, con la posibilidad de relacionarlos, ordenarlos en base a diferentes criterios, etc.

CSS (Cascading Style Sheets): Es una colección de reglas que afectan a la apariencia de un documento. Estas reglas se refieren al modo en que aparecerá un documento en pantalla cuando el usuario utilice un navegador o explorador gráfico, controlando por ejemplo el color, el fondo, tipo de fuente, apariencia de los bordes, márgenes, alineación y espacio entre caracteres.

Diagrama de Gantt: es una herramienta que le permite al usuario modelar la planificación de las tareas necesarias para la realización de un proyecto

Diseño físico de BBDD: descripción de la implementación de una base de datos en memoria secundaria, describiendo las estructuras de almacenamiento y los métodos de acceso a esos datos.

Diseño lógico de BBDD: Es una descripción de la estructura de la base de datos que puede procesar un SGBD.

Esquema conceptual: Representación gráfica o simbólica de un concepto.

Framework: Estructura conceptual y tecnológica de soporte definido, normalmente con módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

HTML (Hypertext Markup Language): Es un lenguaje de programación que está diseñado para crear páginas web con hipervínculos. HTML se usa para formatear texto (es decir, para designar cierto texto como título, párrafo, lista, etc.), además de estructurar la información de una manera determinada.

Java: Lenguaje de programación orientado a objetos que se popularizó a partir del lanzamiento de su primera versión comercial de amplia difusión, la JDK 1.0 en 1996. Actualmente es uno de los lenguajes más usados para la programación en todo el mundo.

JDK (Java Development Kit): Es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.

jUnit: Es un framework java que permite la realización de la ejecución de clases de manera controlada, para poder comprobar que los métodos realizan su cometido de forma correcta.

Linux: Sistema operativo completo, robusto, disponible libremente, que fue desarrollado originalmente por Linus Torvalds.

PHP(Hypertext Preprocessor): Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Servidor: Sistema informático (ordenador) que presta ciertos servicios y recursos (de comunicación, aplicaciones, ficheros, etc.) a otros ordenadores (denominados clientes), los cuales están conectados en red a él.

POJO: Un objeto POJO es una instancia de una clase que no extiende ni implementa nada en especial. Por ejemplo, un Servlet tiene que extender de HttpServlet y sobrescribir sus métodos, por lo

tanto no es un POJO. En cambio, si se define una clase 'Persona', con sus atributos privados y sus correspondientes getters y setters públicos, una instancia de esta simple clase es un objeto POJO.

SGBD: Sistema de gestión de base de datos.

Versión beta: Es una versión de software a la cual se le han realizado las pruebas internas y se ha lanzado a los usuarios para que realicen las pruebas públicas. Es el prototipo del producto final.

Anexo 7. Bibliografía

Nita Sáenz Higuera; Rut Vidal Oltra (2014) *Redacción de textos científico técnicos*. Barcelona.

Universitat Oberta de Catalunya. [Fecha de consulta: Septiembre de 2014]

Jeff Linwood; Dave Minter (2010) *Beggining Hibernate*. Apress.[Fecha de consulta:Noviembre de 2014]

Wikipedia (2013). Plain Old Java Object. Desde http://es.wikipedia.org/wiki/Plain_Old_Java_Object

[Fecha de consulta: Noviembre 2014]

Hosting y Dominios. Glosario. Desde: http://hosting.astalaweb.net/Glosario.asp#.VLOY6SuG_e4

[Fecha de consulta: Noviembre de 2014]

mkyong (2011) *Hibernate – Many-To-Many Example (Annotation)*. Desde:

<http://www.mkyong.com/hibernate/hibernate-many-to-many-relationship-example-annotation/> [Fecha

de consulta:Noviembre de 2014]

Rogers Candenhead (2012) *Programación Java 7*. Madrid. ANAYA. [Fecha de consulta: Noviembre

de 2014]

Desconocido (2009) *Struts 2*. Desde:

<https://code.google.com/p/prog2-60g-20121/downloads/detail?name=Manual-Struts2-Espanol.pdf>

[Fecha de consulta: Noviembre de 2014]

mkyong (2010) *Struts 2 Date Tag Example*. Desde:

<http://www.mkyong.com/struts2/struts-2-date-tag-example/> [Fecha de consulta:Noviembre de 2014]

mkyong (2010) *Struts2 File Upload Example*. Desde:

<http://www.mkyong.com/struts2/struts-2-file-upload-example/> [Fecha de consulta:Noviembre de 2014]