

SEPROSO.
Modelo de Implementación.

Francisco Javier Delgado del Hoyo

Yuri Torres de la Sierra

Rubén Martínez García

Abel Lozoya de Diego

Enero, 2009

Revisiones del documento

Historial de revisiones del documento

VERSIÓN	FECHA	DESCRIPCIÓN	AUTOR
0.1	2/01/09	Objetivos y visión general.	Rubén.

Indice

Revisiones del documento	i
1 Introducción.	1
1.1 Propósito.	1
1.2 Ámbito.	1
1.3 Definiciones.	1
1.4 Referencias.	2
1.5 Visión general.	3
2 Vista general del Modelo de Implementación.	4
3 Elaboración	5
3.1 Componnetes y subsistemas.	5
3.1.1 Componnetes y subsistemas a implementar.	5
3.1.2 Diagramas del modelo de Implementación.	6
3.2 Producto.	6
4 Construcción 1	8
4.1 Componnetes y subsistemas.	8
4.1.1 Componnetes y subsistemas a implementar.	8
4.1.2 Diagramas del modelo de Implementación.	10
4.2 Producto.	10
5 Construcción 2	12
5.1 Componnetes y subsistemas.	12
5.1.1 Componnetes y subsistemas a implementar.	12
5.1.2 Diagramas del modelo de Implementación.	12
5.2 Producto.	13
6 Producción	15
6.1 Componnetes y subsistemas.	15
6.1.1 Componnetes y subsistemas a implementar.	15
6.1.2 Diagramas del modelo de Implementación.	16
6.2 Producto.	16

Indice de figuras

Indice de tablas

Capítulo 1

Introducción.

1.1 Propósito.

Este documento presenta el plan detallado de desarrollo de componentes e integración de los mismos dentro de cada iteración del proyecto.

1.2 Ámbito.

Este documento se aplica sobre todas las iteraciones de SEPROSO. Su contenido irá creciendo a medida que avance el desarrollo con cada nueva iteración. El documento va creándose según evoluciona el modelo de implementación del proyecto.

1.3 Definiciones.

Véase el Glosario.

- **PRADO (Php Rapid Application Development Object-oriented)**: framework de PHP que provee gran cantidad de componentes que facilitan el desarrollo e implementación de aplicaciones Web.
- **PRADO PHP Class**: clase PHP de PRADO que contiene los métodos y atributos definidos en ella, además de los componentes declarados en la página PRADO Page del mismo nombre. Su extensión es .php.
- **PRADO Page**: documento XHTML donde se especifica la presentación o interfaz de usuario para una página web asociada a una clase PRADO PHP Class con el mismo nombre. Su extensión es .page.

Toda PRADO page necesita tener su propia PRADO PHP Class aunque no defina comportamiento en ella.

- **PRADO Template Control:** plantilla para alojar páginas Web o componentes de PRADO. Se utiliza para definir esquemas de contenidos para índices o menús. Es muy similar a un componente PRADO Page sólo que ésta se puede insertar en otra PRADO Page. También lleva asociada una PRADO PHP Class del mismo nombre pero diferente extensión que define su comportamiento. Su extensión es .tpl.
- **XML SQL Map:** descripción en XML de las consultas SQL que se realizan a una base de datos y del mapeo automático que se realiza de los resultados obtenidos como entradas de tablas hacia objetos de la aplicación (también llamado "binding"). A estas consultas se las identifica con un nombre que será utilizado por un DAO para ejecutar la consulta y devolver los objetos correspondientes según mapeo.
- **DAO (Data Access Object):** objeto que permite enlazar la lógica de negocio orientada a objeto con una base de datos relacional, ejecutando las consultas como métodos gracias a ficheros XML SQL Map.

1.4 Referencias.

- Glosario, Grupo 3, Universidad de Valladolid.
- Especificación de Casos de Uso, Grupo 3, Universidad de Valladolid..
- Especificación Suplementaria, Grupo 3, Universidad de Valladolid..
- Plan de Desarrollo Software, Grupo 3, Universidad de Valladolid..
- Plan de Iteración, Grupo 3., Universidad de Valladolid.
- UPEDU. <http://www.upedu.org/upedu/>.
- Modelo de análisis, Grupo 3, Universidad de Valladolid.
- Modelo de diseño, Grupo 3, Universidad de Valladolid.
- PRADO Documentation, www.pradosoft.com/documentation, PRADO.

1.5 Visión general.

En el primer apartado de este documento se presenta una visión general del Modelo de Implementación final desarrollado para el proyecto SEPROSO. Esto incluye sus productos, paquetes, componentes y los diagramas UML que los relacionan detalladamente.

Después el documento contiene varios apartados, uno por cada iteración con actividades de implementación, donde se identifican los componentes y subsistemas que van a ser implementados, junto a sus respectivos diagramas. También se especifica cómo construir, probar y evaluar el producto generado.

Capítulo 2

Vista general del Modelo de Implementación.

En la siguiente sección se presenta el Diagrama Global UML del Modelo de Implementación (con componentes y paquetes) y los diagramas UML con el contenido de los paquetes a mayor nivel de detalle. Esto provee una visión general de todos los componentes que serán implementados en SEPROSO.

Capítulo 3

Elaboración

3.1 Componnetes y subsistemas.

3.1.1 Componnetes y subsistemas a implementar.

En esta iteración las actividades de implementación se centran en obtener un prototipo no funcional en el que se presenten las líneas guía de la Interfaz Gráfica de Usuario al cliente. Esto implica implementar los siguientes paquetes y componentes:

1. Básicos

- MainLayout (PRADO Template Control y PRADO PHP Class)
- SiteMap (PRADO Template Control y PRADO PHP Class)
- Welcome (PRADO Page)
- Contact (PRADO Page)

2. Paquete User

- " Login (PRADO page y PRADO PHP Class)
- Logout (PRADO page)
- UserList (PRADO page y PRADO PHP Class)
- UserCreate (PRADO page y PRADO PHP Class)

3. Paquete Project

- " Templates (PRADO page)
- Phases (PRADO page)
- Activities (PRADO page)
- AddActivity (PRADO page)
- Workers (PRADO page)

- WorkRegister (PRADO page)
 - ReviewAct (PRADO page)
4. Paquete Report
- " ReportList (PRADO page)
 - WorkRegisters (PRADO page)
 - FutureActs (PRADO page)
 - PlannedActs (PRADO page)
 - Summary (PRADO page)
5. Paquete Theme
- Images (GIF) ' Iconos: ok, error, menú, logo, imágenes decorativas.
 - Report (CSS) ' Estilo de presentación de informes y periodos.
 - Site (CSS) ' Estilo decorativo principal del sitio web.

La numeración de esta lista establece el orden en que deben ser implementados los componentes para conseguir que estén listos al mismo tiempo y facilitar su integración.

3.1.2 Diagramas del modelo de Implementación.

Los diagramas de componentes y paquetes del Modelo de Implementación asociados con esta iteración son los siguientes: figura 1 (sección 2.1).

En las siguientes figuras se muestra un Diagrama de Implementación a diferentes niveles de detalle del producto a construir como resultado de esta iteración (Prototipo GUI).

3.2 Producto.

Se *construye* integrando todos los componentes y subsistemas listados en la sección 3.1.1 de este documento generando una herramienta básica no funcional: login y logout, informe básico con selección de periodo y formulario de introducción de datos de fases y actividades.

Este producto se *utiliza* para verificar si el proyecto es viable y si las líneas guía definidas del mismo se ven afectadas. Este prototipo ayuda a establecer la arquitectura principal del sistema ya que todos los subsistemas

posteriormente implementados serán integrados dentro de este prototipo básico.

Su *evaluación* consistirá en determinar si un usuario se siente cómodo y atraído por la interfaz, y encuentra fáciles de usar las funcionalidades supuestas en el sistema. Además se comprobará si la arquitectura empleada es extensible fácilmente para completar la interfaz de usuario y la funcionalidad más adelante.

Para su *instalación* solo es necesario ejecutar el script de instalación sobre un servidor web con PHP y abrir una página de SEPROSO con un navegador web.

Para *probarlo* se siguen los apartados de todas las pruebas del Plan de Pruebas relacionados con la Interfaz de Usuario. Sólo hay que comprobar la fluida y fácil interacción entre el usuario y la herramienta en tiempo real (tiempo empleado en encontrar un caso de uso o en hacer algo que desea). Si los resultados no fuesen los esperados se debería rediseñar la interfaz, de acuerdo a los gustos del usuario, minimizando el impacto que esto suponga sobre la arquitectura inicial (esto es, modificar el estilo y la presentación de los contenidos pero no cambiar el diseño realizado)

Capítulo 4

Construcción 1

4.1 Componnetes y subsistemas.

4.1.1 Componnetes y subsistemas a implementar.

En esta iteración las actividades de implementación se centran en conseguir un producto que contenga funcionalidad básica para los casos de uso críticos descritos en el Plan de Iteración: Construcción I. Esto implica implementar los siguientes paquetes y componentes:

1. Básicos

- MainLayout (PRADO PHP Class)
- SeprosoExceptions (PRADO PHP Class)
- Exceptions (TXT)

2. Paquete User

3. Paquete Project

- Activities (PRADO PHP Class)
- AddActivities (PRADO PHP Class)
- ApproveAct (PRADO page y PRADO PHP Class)
- Phases (PRADO PHP Class)
- ProjectCreate (PRADO page y PRADO PHP Class)
- ProjectList (PRADO page y PRADO PHP Class)
- ReviewAct (PRADO PHP Class)
- Templates (PRADO PHP Class)
- Workers (PRADO PHP Class)
- WorkRegister (PRADO PHP Class)

4. Paquete Report

- ActiveActs (PRADO page y PRADO PHP Class)
- ActiveWorkers (PRADO page y PRADO PHP Class)
- Activities (PRADO page y PRADO PHP Class)
- ApproveActs (PRADO page y PRADO PHP Class)
- AssignedActs (PRADO page y PRADO PHP Class)
- DelayedActs (PRADO page y PRADO PHP Class)
- FutureActs (PRADO PHP Class)
- FutureWorkers (PRADO page y PRADO PHP Class)
- PlannedActs (PRADO PHP Class)
- ReviewedActs (PRADO page y PRADO PHP Class)
- Summary (PRADO PHP Class)
- Worker (PRADO page y PRADO PHP Class)
- WorkRegisters (PRADO PHP Class)

5. Paquete Theme

- Project (CSS)
- Work-record (CSS)

6. Paquete DAO

- DaoManager (PRADO PHP Class)
- ProjectDao (PRADO PHP Class)
- ReportsDao (PRADO PHP Class)
- WorkerDao (PRADO PHP Class)
- WorkRecordDao (PRADO PHP Class)
- UserDao (PRADO PHP Class)
- ProjectRecord (PRADO PHP Class)
- WorkerRecord (PRADO PHP Class)
- WorkRecord (PRADO PHP Class)

7. Paquete Login

- SeprosoAuthManager (PRADO PHP Class)
- UserManager (PRADO PHP Class)
- SeprosoUser (PRADO PHP Class)

8. Paquete DataBase

- SeprosoUserTypeHandler (PRADO PHP Class)
- Project (XML SQL Map)
- Worker (XML SQL Map)
- WorkRecord (XML SQL Map)
- User (XML SQL Map)
- Reports (XML SQL Map)

La numeración de esta lista establece el orden en que deben ser implementados los componentes para conseguir que estén listos al mismo tiempo y facilitar su integración.

4.1.2 Diagramas del modelo de Implementación.

Los diagramas de componentes y paquetes del Modelo de Implementación asociados a esta iteración son los siguientes: figura 1 (sección 2.1).

En las siguientes figuras se muestra un Diagrama de Implementación a diferentes niveles de detalle del producto a construir como resultado de esta iteración (Versión Alpha).

4.2 Producto.

Se *construye* integrando todos los componentes y subsistemas listados en la sección 4.1.1 de este documento produciendo una herramienta con la siguiente funcionalidad: sesión de usuario, gestión de proyecto, generación de informes (algunos, no todos), y registro de informes de actividad.

Este producto se *utiliza* para que los probadores comiencen las pruebas de requisitos y casos de uso implementados. El producto como tal es aún inestable y contiene errores a depurar, además de que sólo implementa una parte de la funcionalidad, aunque satisface la mayoría de los requisitos. La *evaluación consiste* en comprobar si la herramienta permite a los usuarios crear y planificar un proyecto, registrar informes de trabajo y generar algún tipo de informe de forma adecuada y sin errores.

La *instalación se realiza* del mismo modo que en la sección 3.2.

Las *pruebas* se realizan comprobando el correcto comportamiento de la interfaz de usuario, de la base de datos y del código PHP para todos los requisitos a probar indicados en el Plan de Pruebas. Además se pasan todos

los casos de prueba de los casos de uso implementados en esta iteración.

Capítulo 5

Construcción 2

5.1 Componnetes y subsistemas.

5.1.1 Componnetes y subsistemas a implementar.

En esta iteración se pretende conseguir un producto que proporcione funcionalidad completa de los casos de uso críticos y una funcionalidad parcial en los casos de uso añadidos respecto a la iteración anterior que se especifican en el *Plan de Iteración: Construcción II*. Esto implica implementar los siguientes nuevos paquetes y componentes.

El orden de esta lista determina implícitamente el orden en que deben ser implementados los componentes para conseguir que estén listos al mismo tiempo y facilitar su integración.

5.1.2 Diagramas del modelo de Implementación.

Los diagramas de componentes y paquetes del Modelo de Implementación asociados con esta iteración son los siguientes: figura 1 (sección 2.1).

1. Paquete User

- Holidays (PRADO page y PRADO PHP Class)

2. Paquete Project

- AddActivities (PRADO page y PRADO PHP Class)

3. Paquete Report

- Activities (PRADO page y PRADO PHP Class)
- Worker (PRADO page y PRADO PHP Class)
- WorkRegisters (PRADO page y PRADO PHP Class)

4. Paquete DAO

- ActivityDao (PRADO PHP Class)
- ProcessDao (PRADO PHP Class)
- PhaseDao (PRADO PHP Class)
- ReportsDao (PRADO PHP Class)
- ActivityRecord (PRADO PHP Class)
- ProcessRecord (PRADO PHP Class)
- PhaseRecord (PRADO PHP Class)
- Paquete DataBase
- DateTimeTypeHandler (PRADO PHP Class)
- Activity (XML SQL Map)
- Process (XML SQL Map)
- Phase (XML SQL Map)
- Reports (XML SQL Map)

En las siguientes figuras se muestra un Diagrama de Implementación a diferentes niveles de detalle del producto a construir como resultado de esta iteración (Versión Beta).

5.2 Producto.

Se *construye* integrando todos los componentes y subsistemas listados en la sección 5.1.1 de este documento obteniendo una herramienta que presenta toda la funcionalidad especificada en el Documento de Especificación de Casos de Uso, si bien no de forma estable o libre de errores.

Este *producto* se utiliza para que los probadores completen las pruebas de requisitos y casos de uso implementados y para depurar los errores detectados en la iteración anterior. El producto todavía no es estable ya que faltan de corregir los errores en los nuevos casos de uso, pero satisface todos los requisitos.

La *evaluación* consiste en comprobar si la herramienta permite a cualquier tipo de usuario realizar todos los casos de uso sin errores y de la forma esperada. También se pretende conseguir una arquitectura completa y estable, aunque algunos de los componentes estén incompletos.

La *instalación* se realiza como la del Prototipo en la sección 3.2.

Las *pruebas* consisten en pasar todos los casos de prueba diseñados por los probadores y en poner el producto a disposición de los beta-testers para detectar errores que se prevén en los casos de prueba.

Capítulo 6

Producción

6.1 Componnetes y subsistemas.

6.1.1 Componnetes y subsistemas a implementar.

En esta iteración se busca conseguir implementar una versión definitiva de la herramienta que pueda ponerse en producción y que proporcione al usuario toda la funcionalidad establecida en los casos de uso y verifique todos los requisitos establecidos. Hay que completar todos los componentes y dejarlos en una versión estable operativa. Aquí se consiguen ya todos los paquetes y componentes, aunque los que se implementan en mayor medida son:

1. Paquete User
 - (a) Holidays (PRADO page y PRADO PHP Class)
2. Paquete Project
 - AddActivities (PRADO page y PRADO PHP Class)
3. Paquete Report
 - Worker (PRADO page y PRADO PHP Class)
 - WorkRegisters (PRADO page y PRADO PHP Class)
4. Paquete DAO
 - ActivityDao (PRADO PHP Class)
 - ArtifactDao (PRADO PHP Class)
 - ConfigurationDao (PRADO PHP Class)
 - ReportsDao (PRADO PHP Class)
 - ActivityRecord (PRADO PHP Class)

- ArtifactRecord (PRADO PHP Class)
- ConfigurationRecord (PRADO PHP Class)

5. Paquete DataBase

- Activity (XML SQL Map)
- Artifact (XML SQL Map)
- Configuration (XML SQL Map)
- Reports (XML SQL Map)

La numeración de esta lista establece un orden para diseño TOP-DOWN de implementación de los componentes para conseguir que estén listos al mismo tiempo y facilitar su integración.

6.1.2 Diagramas del modelo de Implementación.

Los diagramas de componentes y paquetes del Modelo de Implementación asociados con esta iteración son los siguientes: figura 1 (sección 2.1).

En las siguientes figuras se muestra un Diagrama de Implementación a diferentes niveles de detalle del producto a construir como resultado de esta iteración (Versión Final).

6.2 Producto.

Se *construye* integrando todos los componentes y subsistemas de la lista (sección 6.1.1). De aquí se obtiene una herramienta plenamente funcional y sin errores, lista para ser puesta a disposición del cliente y en fase de mantenimiento. Esto implica que permite desarrollar al usuario todos los casos de uso de forma correcta.

Se *utiliza* para depurar todos los errores que quedaron pendientes en la versión beta, además de añadir pequeños detalles de funcionalidad a los casos de uso que se implementaron parcialmente entonces.

Los criterios de *evaluación* son claros: obtener un producto estable, que cumpla todos los requisitos iniciales y que permita desarrollar todos los casos de uso desde su interfaz gráfica. En resumen: conseguir un producto aceptado por el cliente y listo para poner en producción indefinidamente.

Para su *instalación* se aplica el mismo script que en la sección 3.2.

Para *probarlo* se aplican todos los casos de prueba de nuevo, se evalúan todos los requisitos a probar del Plan de Pruebas y se verifica que no se produzca ningún error o respuesta inesperada en ningún caso.