

J2EE: UNIVERSAL CMIS CLIENT

Miguel Segura Anaya
ETIG / ETIS

Jose Juan Rodriguez

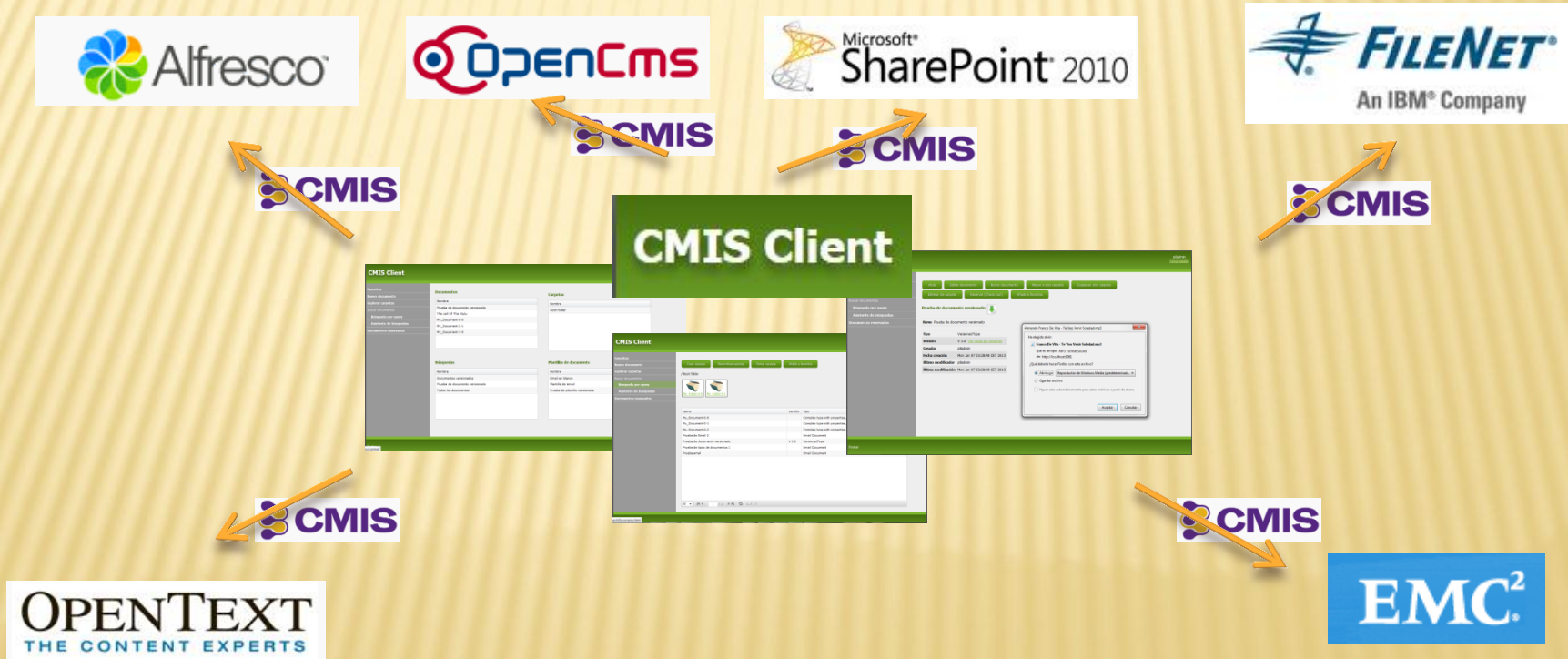
14 de Enero de 2013

INTRODUCCIÓN

- ✘ La gestión de la documentación ha sido durante muchos años un quebradero de cabeza para las organizaciones.
- ✘ Los gestores documentales permiten que toda esa información almacenada, clasificada y parametrizada sea fácilmente localizable por el usuario.
- ✘ Cada fabricante dota cada uno de ellos con su arquitectura, funcionalidades y características propias.
- ✘ Surge la necesidad de integrar estos gestores de contenidos con las herramientas de trabajo diarias, ERPs, BPMs o CRMs.
- ✘ Las organizaciones se “casan” de por vida con los gestores documentales ya que un cambio de los mismos puede llegar a afectar a todos los demás sistemas.
- ✘ El estándar CMIS, siglas de Content Management Interoperability Services, es un estándar diseñado por los principales fabricantes de gestores de contenido empresarial para la su propio ámbito.
- ✘ CMIS permite el acceso de forma estándar y única a los sistemas de gestión de contenidos que lo implementan.

OBJETIVOS DEL TFC

- ✘ El objetivo de este proyecto es desarrollar una aplicación J2EE que permita a un usuario tipo, realizar las operaciones más comunes en el uso de un gestor documental.
- ✘ Establecer un framework de desarrollo que nos permita afrontar cualquier otra aplicación con las máximas garantías.



ENFOQUE Y METODOLOGÍA

- ✘ Modelo lineal basado en etapas e hitos fijados por los entregables
- ✘ Plan de trabajo
- ✘ Entrega documento plan
- ✘ Preparación del entorno
- ✘ Análisis funcional
- ✘ Diseño técnico
- ✘ Entrega documento análisis y diseño
- ✘ Implementación
- ✘ Entrega maqueta
- ✘ Implementación y documentación
- ✘ Entrega final

[-] Cliente CMIS J2EE	81 días	lun 24/09/12	lun 14/01/13	
Plan de trabajo	8 días	lun 24/09/12	mié 03/10/12	
Entrega plan de trabajo	0 días	mié 03/10/12	mié 03/10/12	
[-] Instalación entorno	5 días	mié 03/10/12	mar 09/10/12	5
Instalacion SO Suse	1 día	mié 03/10/12	mié 03/10/12	
Instalación BBDD DB2	1 día	jue 04/10/12	jue 04/10/12	7
Instalación Servidor Aplicaciones W	1 día	vie 05/10/12	vie 05/10/12	8
Instalación Filenet	1 día	lun 08/10/12	lun 08/10/12	9
Instalación CMIS	1 día	mar 09/10/12	mar 09/10/12	10
[-] Analisis funcional	11 días	mié 03/10/12	mié 17/10/12	5
Actores	1 día	mié 03/10/12	mié 03/10/12	5
Casos de uso	3 días	jue 04/10/12	lun 08/10/12	13
Procesos	7 días	mar 09/10/12	mié 17/10/12	14
[-] Diseño	16 días	jue 18/10/12	jue 08/11/12	12
Diagrama de datos	1 día	jue 18/10/12	jue 18/10/12	12
Diagrama de clase	3 días	vie 19/10/12	mar 23/10/12	17
Diseño aplicación	2 días	mié 24/10/12	jue 25/10/12	18
Diseño formularios	5 días	vie 26/10/12	jue 01/11/12	19
Diseño listados	5 días	vie 02/11/12	jue 08/11/12	20
Entrega documento analisis y diseño	0 días	jue 08/11/12	jue 08/11/12	
[-] Implementación	32 días	jue 08/11/12	vie 21/12/12	22
Implementación core	4 días	jue 08/11/12	mar 13/11/12	22
Alta documento	6 días	mié 14/11/12	mié 21/11/12	24
Exploración carpetas	3 días	jue 22/11/12	lun 26/11/12	25
Busqueda de documentos	8 días	mar 27/11/12	jue 06/12/12	26
Detalle documento	4 días	vie 07/12/12	mié 12/12/12	27
Gestión favoritos	7 días	jue 13/12/12	vie 21/12/12	28
Entrega maqueta	0 días	lun 17/12/12	lun 17/12/12	
Documentación	16 días	lun 24/12/12	lun 14/01/13	23
Pruebas y correcciones	16 días	lun 24/12/12	lun 14/01/13	23
Entrega final	0 días	lun 14/01/13	lun 14/01/13	

ANALISIS - ACTORES Y CASOS DE USO

- ✘ Usuario no autenticado
 - + Identificación en la aplicación
- ✘ Usuario autenticado
 - + Crear un nuevo documento
 - + Buscar documentos
 - + Explorar carpetas
 - + Consultar documento
 - + Modificar metadatos de un documento
 - + Eliminar documento
 - + Crear carpeta
 - + Añadir documento a carpeta
 - + Extraer documento de carpeta
 - + Eliminar carpeta
 - + Renombrar carpeta
 - + Reservar documento
 - + Actualizar documento
 - + Cancelar una reserva
 - + Consultar versiones
 - + Consultar detalle de versión
 - + Eliminar versión
 - + Consultar favoritos
 - + Añadir documento a favoritos
 - + Añadir carpeta a favoritos
 - + Añadir búsqueda a favoritos
 - + Añadir formulario a favoritos
 - + Utilizar favorito
 - + Eliminar favorito
 - + Consultar documentos reservados
 - + Buscar documentos por query

ANALISIS – PROTOTIPADO DE PANTALLAS

- ✘ Diseño de las pantallas de la compacto y coherente.



DISEÑO – BASE ARQUITECTÓNICA J2EE

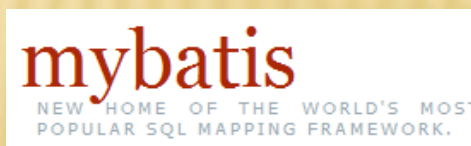
- ✘ J2EE es una plataforma Java, creada para la construcción de aplicaciones orientadas al cliente web.
- ✘ Cubre prácticamente todas las necesidades que este tipo de aplicaciones requiere.
- ✘ da solución a necesidades del desarrollador como son la seguridad, la persistencia, la intercomunicación entre sistemas, la transaccionalidad, etc.
- ✘ Establece estándares como el de portlet, servlet , EJB o web services.

DISEÑO – PATRÓN MVC

- ✘ Patrón de diseño de aplicaciones que especifica cómo debe ser estructurada una aplicación.
- ✘ Separa las aplicaciones Web en tres componentes distintos con el objetivo de independizar los.
- ✘ La lógica de negocio de la aplicación, incluyendo el acceso a los datos y su manipulación, está encapsulada dentro del modelo.
- ✘ Las respuestas HTML que se enviarán al navegador de usuario, serán generadas por la vista.
- ✘ El controlador es el elemento de la aplicación al cual se dirigen todas las peticiones a la capa intermedia que se realicen desde el cliente.
- ✘ Su misión es determinar las acciones a realizar para cada una de estas peticiones e invocar al resto de los componentes de la aplicación (Modelo y Vista)

DISEÑO - MODELO

- ✘ Dos partes, la dedicada a comunicarse con el gestor documental, y la que almacenará las preferencias y configuraciones del usuario y que estará en base de datos.
- ✘ La comunicación con el gestor documental, se llevará a cabo utilizando las librerías OpenCMIS de Apache Chemistry.
- ✘ La comunicación con el gestor de base de datos, se utilizará el framework MyBatis.
- ✘ Se ha utilizado la herramienta MyBatis generator para generar de forma automática beans y Daos a partir de las tablas de la base de datos.



DISEÑO - VISTA

- ✘ Para la elaboración de la interface de usuario, se ha utilizado como base de construcción de las páginas los JSPs, ampliando la funcionalidad básica con etiquetas JSTL .
- ✘ Las validaciones y las ayudas de los formularios se han realizado mediante javascript, más concretamente utilizando Mootools.
- ✘ Las funcionalidades básicas de Mootools se han ampliado con bibliotecas adicionales para los selectores de fecha, tablas de datos y ventanas modales.
- ✘ El envío de los datos a las tablas de datos se ha realizado utilizando el formato JSON, utilizando la librería Jackson para la conversión entre JSON y Java.
- ✘ El diseño final se ha realizado utilizando ficheros css, un fichero principal para el diseño de la aplicación mas otros incorporados por las extensiones a Mootools.
- ✘ La maquetación de las páginas se ha realizado con Divs.



DISEÑO - CONTROLADOR

- ✘ El controlador que utilizamos para construir la aplicación es el de Spring, concretamente en su modalidad de anotaciones.
- ✘ Enlaza las urls de las páginas directamente a los métodos de las clases controladoras.
- ✘ Controla los parámetros obligatorios de llamada a las páginas también con anotaciones.
- ✘ Genera los contenedores de información que enviaremos a los JSP de presentación mediante objetos ModelAndView compatibles con jstl.
- ✘ Da soporte al “upload” de ficheros de manera sencilla y práctica.
- ✘ Configuración básica en xml, el resto en anotaciones.



DISEÑO – OTRAS CONSIDERACIONES

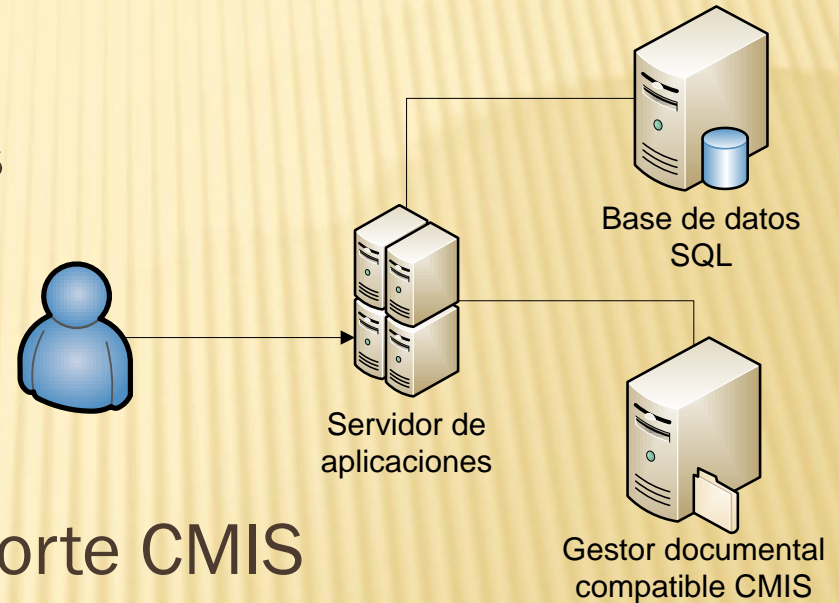
- ✘ Para gestionar los posibles errores que se puedan producir se seguirá una estrategia de envío y no procesado de las excepciones hasta que lleguen al controlador.
- ✘ Se controlará el cierre de las conexiones de base de datos en la sección finally del bloque try, catch , finally.
- ✘ La trazabilidad de la aplicación se realizará utilizando la librería log4j.
- ✘ Todo código que requiera alguna lógica de negocio compleja se organizará mediante métodos estáticos de clases servicio.
- ✘ Las clases se organizarán en paquetes java. Bean, controller, dao, service, servlet y util.
- ✘ El control de seguridad se realizará mediante un filtro que verificará la información de login en la sesión.

IMPLEMENTACIÓN - DECISIONES

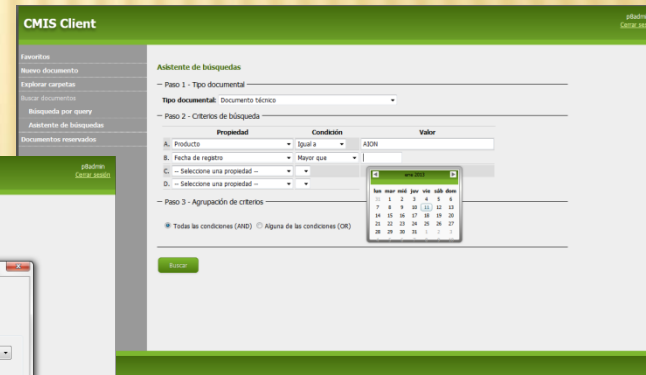
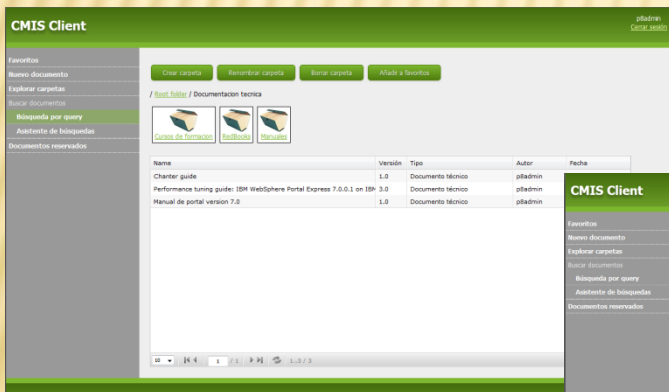
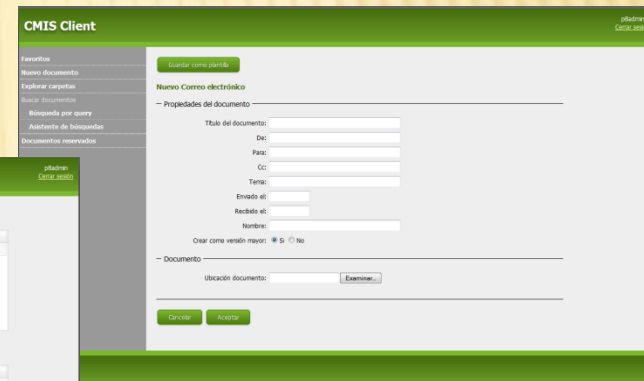
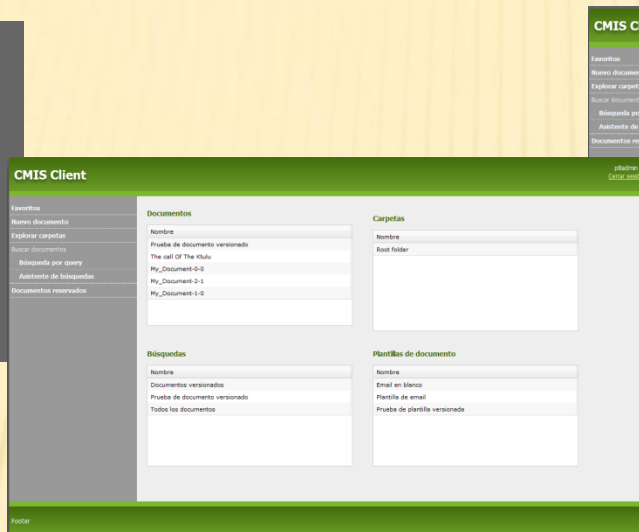
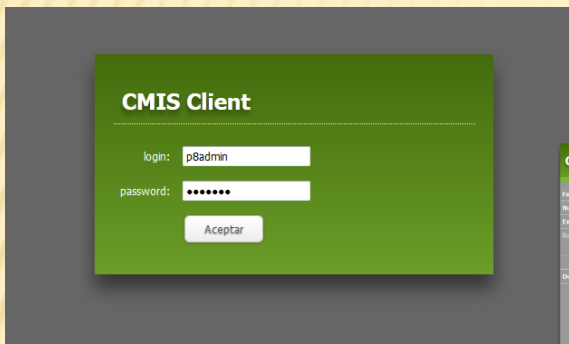
- × Muy fiel al diseño original, salvo algunas correcciones.
- × Corrección en el modelo de datos.
- × Corrección en la gestión de conexiones.
- × Nuevas funcionalidades:
 - + Documentos reservados
 - + Búsqueda por query
- × Parámetros de conexión en ficheros de propiedades
- × Generación automática de objetos de MyBatis
- × Librerías adicionales Mootools
 - + Arian Mootools datepicker
 - + mBox
 - + Omnigrid
- × Carga de listados con JSON
- × Construcción de JSON con librería Jackson
- × Nuevo diseño basado CSS

IMPLEMENTACIÓN - SOFTWARE

- ✘ Servidor aplicaciones J2EE
 - + Definición de origen de datos
 - + Despliegue aplicación
- ✘ Base de datos SQL
 - + Creación de tablas
- ✘ Gestor documental con soporte CMIS
 - + Conexión Servicios Web o ATOM
 - + OpenCMIS InMemory Repository para pruebas
- ✘ Navegador Firefox



IMPLEMENTACIÓN - CMIS CLIENT



CONCLUSIONES - FUNCIONALES

- ✘ Cumplimiento de todos los requerimientos establecidos en la fase de análisis.
- ✘ Pocas correcciones.
- ✘ Casos de uso como guía de desarrollo simplifica la gestión y el control del grado de avance.
- ✘ Aparición de nuevas funcionalidades de tres tipos:
 - + Necesarias, funcionales.
 - + Bajo impacto, de interface.
 - + A desarrollar en nuevas versiones, nuevas funcionalidades de alta complejidad
- ✘ Objetivo de creación de un cliente de CMIS que permita a un usuario realizar las tareas básicas independientemente del gestor utilizado, cumplido.

CONCLUSIONES - TÉCNICAS

- ✘ Diseño muy efectivo. Desarrollo, rápido, modular y con muy bajo nivel de errores.
- ✘ CMIS consigue la abstracción completa del gestor. Enfoque práctico.
- ✘ Apache Chemistry, implementación clara y simple. Uso intuitivo.
- ✘ Grandes resultados con el MVC.
 - + Modelo de construcción rápida y libre de errores.
 - + Controlador versátil y con utilidades de uso inmediato.
 - + Vista, rápida adaptación a los cambios de diseño.
- ✘ Utilidades de Mootools, prácticas y de gran calidad visual.
- ✘ Objetivo de definición de Framework, cumplido

CONCLUSIONES - PERSONALES

- ✘ Aportación de nuevos conocimientos técnicos
 - + JavaScript
 - + Spring controller
 - + JSON
 - + Anotaciones
 - + Apache Chemistry / CMIS
- ✘ Puesta en práctica de la teoría de gestión de proyectos

EVOLUCIONES FUTURAS DEL PRODUCTO

- ✘ Compatibilidad con otros navegadores.
- ✘ Gestión del control de acceso a carpetas y documentos.
- ✘ Mejoras en la construcción de los formularios de metadatos.
- ✘ Integración con dispositivos de escaneado.
- ✘ Incorporación masiva de documentos.
- ✘ Conexión simultánea a múltiples repositorios.