

Diseño e implementación
framework de presentación
para J2EE

Miguel Arias Fernández
Ingeniería en Informática

Josep María Camps Riba

Enero 2013

ÍNDICE

- ✓ **INTRODUCCIÓN**
 - Justificación y contexto
 - Objetivos
 - Metodología empleada
 - Planificación del proyecto
- ✓ **ANÁLISIS INICIAL**
 - Plataforma J2EE
 - Patrones de diseño
 - Frameworks MVC J2EE: Strut2, JSF y Spring
- ✓ **ANÁLISIS Y DISEÑO**
 - Análisis y diseño del Framework
- ✓ **IMPLEMENTACIÓN Y PRUEBAS**
 - Aplicación ejemplo
 - Desarrollo e implantación
- ✓ **CONCLUSIONES**

JUSTIFICACIÓN Y CONTEXTO

Punto de partida

- Marco de trabajo de la especificación J2EE
- J2EE (Java2 Enterprise Edition) conjunto de especificaciones y técnicas que proporcionan soluciones para desarrollo, despliegue y gestión aplicaciones
- J2EE define estándar para desarrollo de aplicaciones de múltiples niveles
- Importancia de capa de presentación en aplicaciones web complejas J2EE
- Estándar el patrón de diseño Modelo-Vista-Controlador (MVC) en capa presentación
- Frameworks que proporcionan implementación de este patrón de diseño
- Evaluación de distintas propuestas de frameworks existentes
- Realizar el diseño y la implementación de un framework de este estilo

OBJETIVOS

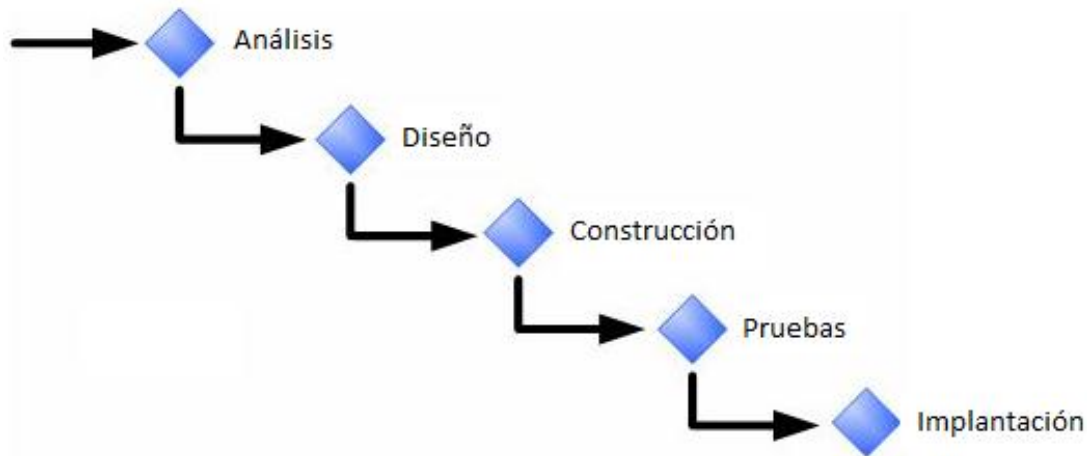
Metas a alcanzar

- Objetivo general
 - Desarrollo de un framework propio para la capa de presentación
 - Buen conjunto de buenas prácticas
 - Patrones de diseño para su elaboración
- Objetivos específicos
 - Análisis estado del arte
 - Estudio capa de presentación en arquitectura J2EE
 - Estudio patrones de diseño enfocados capa de presentación
 - Evaluación y análisis principales frameworks de presentación
 - Elaboración de especificaciones de desarrollo
 - Análisis requerimientos framework de presentación propio
 - Diseño técnico arquitectura
 - Puesta en marcha de un framework de presentación propio
 - Implementación framework según análisis y diseño previos
 - Implementación aplicativo para uso del framework
 - Normas de uso genérico del framework en otros contextos

METODOLOGÍA

Procedimientos a emplear

- Maneras de alcanzar de forma exitosa los objetivos
- Empleo de un desarrollo en cascada



PLANIFICACIÓN



Entrega 1: del 19 de septiembre al 10 de octubre

Entregables:

- Plan de Proyecto

Entrega 2: del 11 de octubre al 8 de noviembre

Entregables:

- Documento Investigación y Análisis carácter general

Entrega 3: del 9 de noviembre al 17 de diciembre

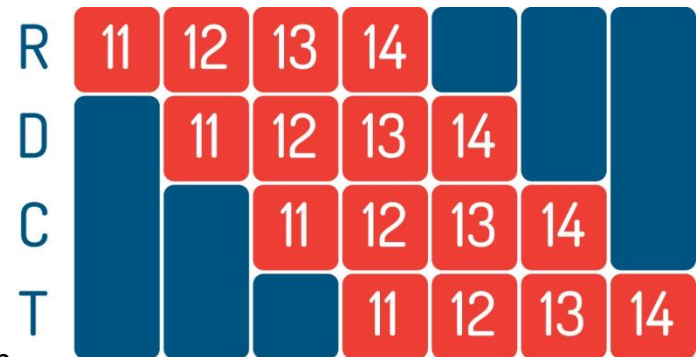
Entregables:

- Documento Análisis y Diseño Framework presentación

Entrega 4: del 18 de diciembre al 14 de enero

Entregables:

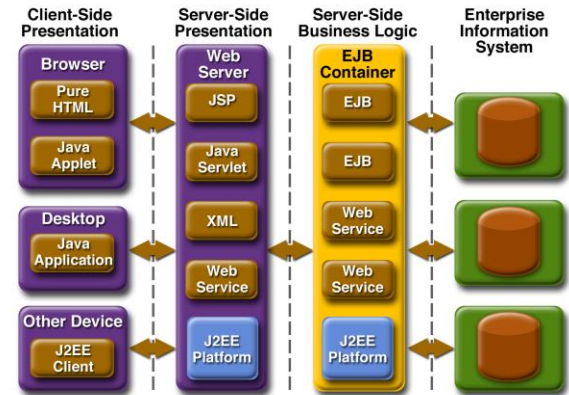
- Producto software final: Código Fuente, JARs, WAR, Javadoc
- Memoria del proyecto
- Presentación del proyecto



PLATAFORMA J2EE

Java Platform, Enterprise Edition

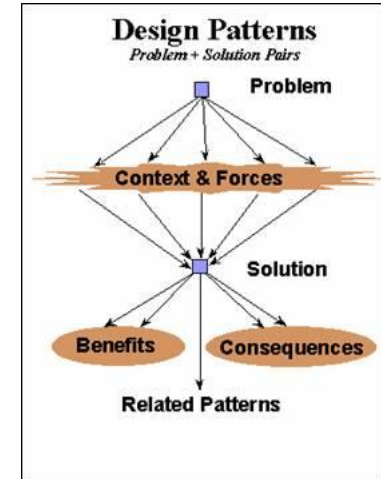
- Características
 - Plataforma abierta y estándar
 - Modelo de aplicaciones distribuido
 - Componentes modulares y estandarizados
- Elementos
 - Componentes (cliente, web y negocio)
 - Contenedores (aplicaciones, applets, web y EJB)
 - Servicios (configurables y no configurables)
 - Servidores de aplicaciones
- Arquitectura multicapa
 - Capa cliente, intermedia y EIS
 - Capa presentación, negocio e integración



PATRONES DE DISEÑO

Solucionando problemas

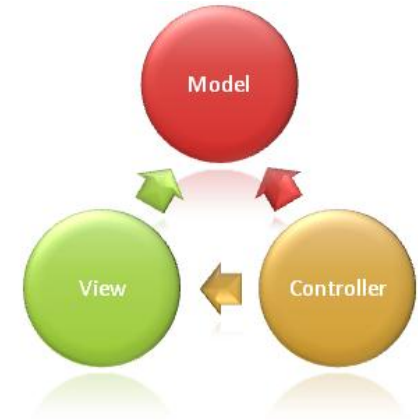
- Ventajas de los patrones
 - Reutilizar las soluciones, encapsular conocimiento, etc.
- Uso de patrones
 - Nombre, contexto, problema, solución y consecuencias
- Tipos de patrones
 - Propósito general: creacionales, estructurales y de comportamiento
 - Etapa del ciclo de vida: análisis, arquitectónicos, asignación de responsabilidades, diseño y programación
- Catálogo de patrones J2EE
 - Capa presentación
 - Capa negocios
 - Capa integración



PATRONES DE DISEÑO

Patrón Modelo-Vista-Controlador

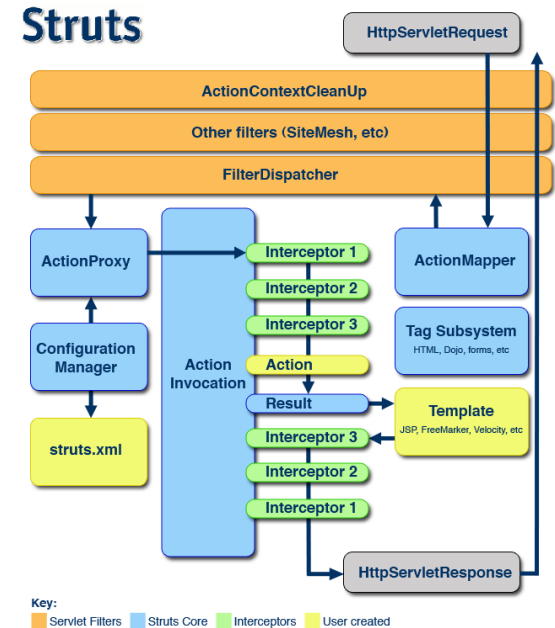
- Patrón MVC en J2EE
 - Patrones de tipología de arquitectura
 - Desacoplar interfaz gráfica resto sistema
- Modelo 1
 - Características de JSP
 - JavaBeans para ejecutar lógica de negocio y determinar página siguiente
- Modelo 2 – MVC
 - Controlador (Controller)
 - Servlet central recibe peticiones, guarda resultado petición y da control a JSP
 - Modelo (Model)
 - JavaBeans o EJBs desempeña rol de modelo
 - Vista (View)
 - Rol ejecutado por JSPs



FRAMEWORKS MVC J2EE

Struts 2

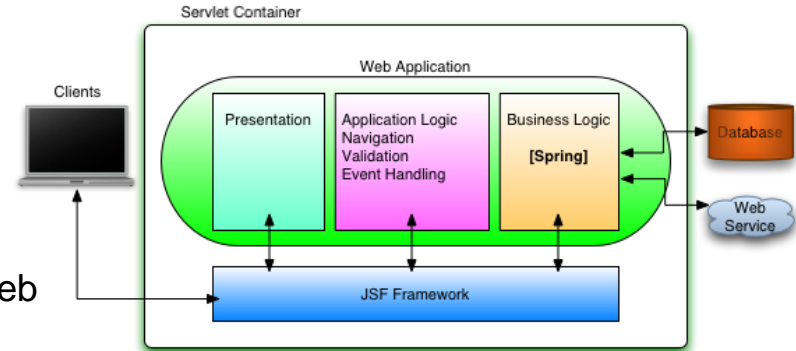
- Arquitectura
 - FilterDispatcher, corazón de Struts2
 - Punto entrada framework
 - Lanza ejecución peticiones
 - Interceptores
 - Ejecutan algún proceso a aplicar a Actions
 - Acciones
 - Realizan la lógica para servir una petición
 - Results
 - Envían la respuesta de regreso al usuario
- Funcionamiento
 - Procesado de las peticiones de una aplicación



FRAMEWORKS MVC J2EE

JavaServer Faces

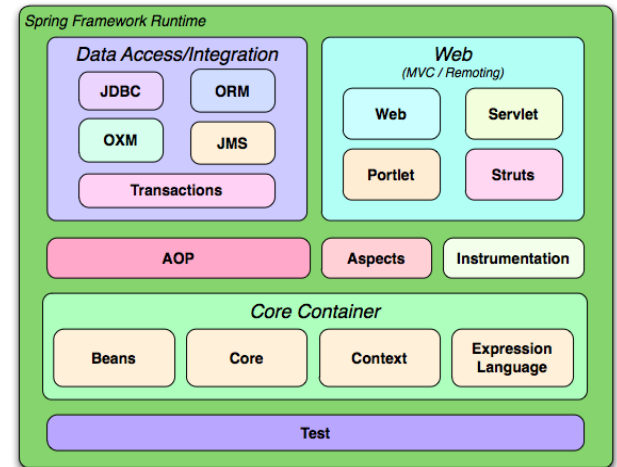
- Arquitectura
 - Beans y Managed Beans
 - Bean: contiene la lógica de negocio
 - Managed Bean: representa formulario web
 - Navegación
 - Conversión y Validación
 - Eventos
- Funcionamiento
 - Respuestas y Peticiones
 - Faces y No-Faces
 - Situaciones del ciclo de vida
 - Petición No-Faces genera una respuesta Faces
 - Petición Faces genera una respuesta no-Faces
 - Petición Faces genera una respuesta Faces
 - Procesado de las peticiones de una aplicación



FRAMEWORKS MVC J2EE

Spring MVC

- Arquitectura
 - El contenedor
 - Núcleo, objetos bean, contexto y lenguaje de expresiones
 - Integración y acceso de Datos
 - Integración módulos JDBC, ORM, OXM, JMS y de transacción
 - Web
 - Módulos Web, Web-Servlet, Web-Struts y Web-Portlet
 - AOP
 - Programación orientada a aspectos
 - Instrumentación
 - Test
 - Permite probar aplicaciones de Spring y componentes
- Funcionamiento
 - Procesado de las peticiones de una aplicación



ANÁLISIS DEL FRAMEWORK

Análisis Framework de presentación MAF

- Análisis realizado en etapas anteriores
- Frameworks MVC analizados cumplen el modelo
 - Basado en acciones utilizado por Struts y Spring
 - Basado en componentes empleado por JSF
- Framework MAF basado en acciones
- Buenas prácticas y combinación de patrones.
 - Uso de Service to Worker
 - Varios patrones Front Controller, Application Controller,...
- Ventajas de utilizar este tipo de frameworks
 - Separación de la vista y el modelo
 - Control centralizado de las acciones
 - Fácil desarrollo y mantenimiento aplicaciones
 - Reutilización de componentes

DISEÑO DEL FRAMEWORK

Diseño Framework de presentación MAF (I)

- Componentes
 - Action context
 - Componente encargado de encapsular el contexto de la acción
 - Form Bean y Form Error
 - Guarda y valida los parámetros que el cliente envíe en la petición
 - Action
 - Ejecuta la lógica de negocio en función de la petición
 - ResultType
 - Genera la vista a partir del contexto de la acción
 - Error Handler
 - Da soporte a las distintas excepciones que se puedan producir
 - Etiquetas
 - Inyectan texto en páginas JSP que conformen la vista
- Configuración
 - Lectura de archivo de configuración
 - ResultTypeConfig
 - FormBeanConfig
 - ActionConfig
 - ErrorHandlerConfig

DISEÑO DEL FRAMEWORK

Diseño Framework de presentación MAF (II)

- Internacionalización
 - Pares clave-valor para distintos idiomas
- Controlador
 - Punto de acceso para las peticiones del cliente
 - Servlet Front Strategy
- Procesado de las peticiones de una aplicación
 - Recepción
 - Obtención de la acción
 - Procesado del formulario
 - Ejecución de la acción
 - Resultado
 - Errores

APLICACIÓN EJEMPLO

Diseño aplicación ejemplo

- Simulación Banca electrónica
 - Usa el framework MAF
- Formularios
 - Extienden del objeto BaseFormBean
- Acciones
 - Uso de la clase BaseAction
- Gestión de errores
 - Implementa el interfaz ErrorHandler
- Tipos de resultado
 - Definido nuevo tipo de resultado
- Resultados globales
 - Resultado global noUsuario para control sesión
- Vistas
 - Páginas JSP con etiquetas del framework

Cuenta destino	Fecha	Cantidad	Saldo	
1111-2222-44-0000055555	01/01/2013 12:00:00	-100,00	9.900,00	Detalle
1111-2222-44-0000055555	01/01/2013 14:00:00	-200,00	9.700,00	Detalle
1111-3333-55-0000066666	01/01/2013 15:00:00	-150,00	9.550,00	Detalle
1111-2222-33-0000044444	01/01/2013 18:00:00	300,00	9.850,00	Detalle
1111-2222-33-0000044444	13/01/2013 13:31:40	45,00	9.895,00	Detalle

[Volver Nueva](#)



DESARROLLO E IMPLANTACIÓN

Framework MAF y aplicación ejemplo

- Desarrollo con herramientas libres
 - Maven, SVN, Eclipse, etc.
- Framework MAF
 - maf-core: módulo principal y compone el núcleo del framework
 - maf-framework módulo adicional
 - Clases base de interfaces descritos.
 - Implementa configuración del framework basada en JAXB
 - Dependencias
- Aplicación de ejemplo
 - Dependencias
 - Basadas en maf-framework y maf-core
 - Configuración
 - Parámetros, resultados, resultados globales, errores, errores globales, formularios y acciones
 - Ejecución
 - jetty-maven-plugin y tomcatX-maven-plugin
 - Definición
 - Acceso a datos y Logs

CONCLUSIONES

Conclusiones obtenidas (I)

- Plataforma J2EE
 - Múltiples posibilidades para creación de sistemas
 - Gran integración con patrones de diseño
 - Numerosa documentación y conocimiento
- Análisis Frameworks
 - Cumplen con su cometido a la perfección.
 - Conjunto de patrones que colaboran para alcanzar objetivo común
 - Buen complemento de los patrones dentro del desarrollo de software
 - Dan respuesta a problemas en base a soluciones ya probadas
 - Forma muy similar de funcionamiento



CONCLUSIONES

Conclusiones obtenidas (II)

- Nuestro framework MAF
 - Posibilidad de extensiones y mejoras
 - Detectadas en la fase de pruebas (aplicación ejemplo)
 - Consecución de objetivos, no sin esfuerzo
 - Máxima colaboración, integración y reutilización
 - Adecuada transparencia con los sistemas cliente
 - Fácil configuración para su uso por terceros
 - Facilidad de extensibilidad y evolución en el futuro
 - Buena modularidad para su mantenimiento
 - Versatilidad para adecuarse al entorno (internacionalización, logs, etiquetas propias, etc.)
- Adquisición de conocimientos J2EE y patrones
 - Se puede abordar estudio de otro componente
 - Se puede descubrir su funcionamiento para replicarlo

