



Eusko Jaurlaritzaren Informatika Elkartea
Sociedad Informática del Gobierno Vasco

Pliego de Bases Técnicas:

Anexo UDA

Fecha: 20/07/2011

Referencia:

EJIE S.A.
Mediterráneo, 14
Tel. 945 01 73 00*
Fax. 945 01 73 01
01010 Vitoria-Gasteiz
Posta-kutxatila / Apartado: 809
01080 Vitoria-Gasteiz
www.ejie.es

Control de documentación

Título de documento: PBT-Anexo UDA

Histórico de versiones

Código:

Versión:

Fecha: 20/07/2011

Resumen de cambios:

Cambios producidos desde la última versión

Primera versión.

Control de difusión

Responsable: Ander Martinez

Aprobado por: Begoña Gutierrez Lizarralde

Firma:

Fecha:

Distribución:

Referencias de archivo

Autor: Consultoría de Areas de Conocimiento

Nombre archivo:

Localización:

Contenido

	Capítulo/sección	Página
	1 Introducción	4
	2 Arquitectura Tecnológica	5
	3 Patrones de interacción	7
	4 Generadores de código	9
	5 Kit de desarrollo	10

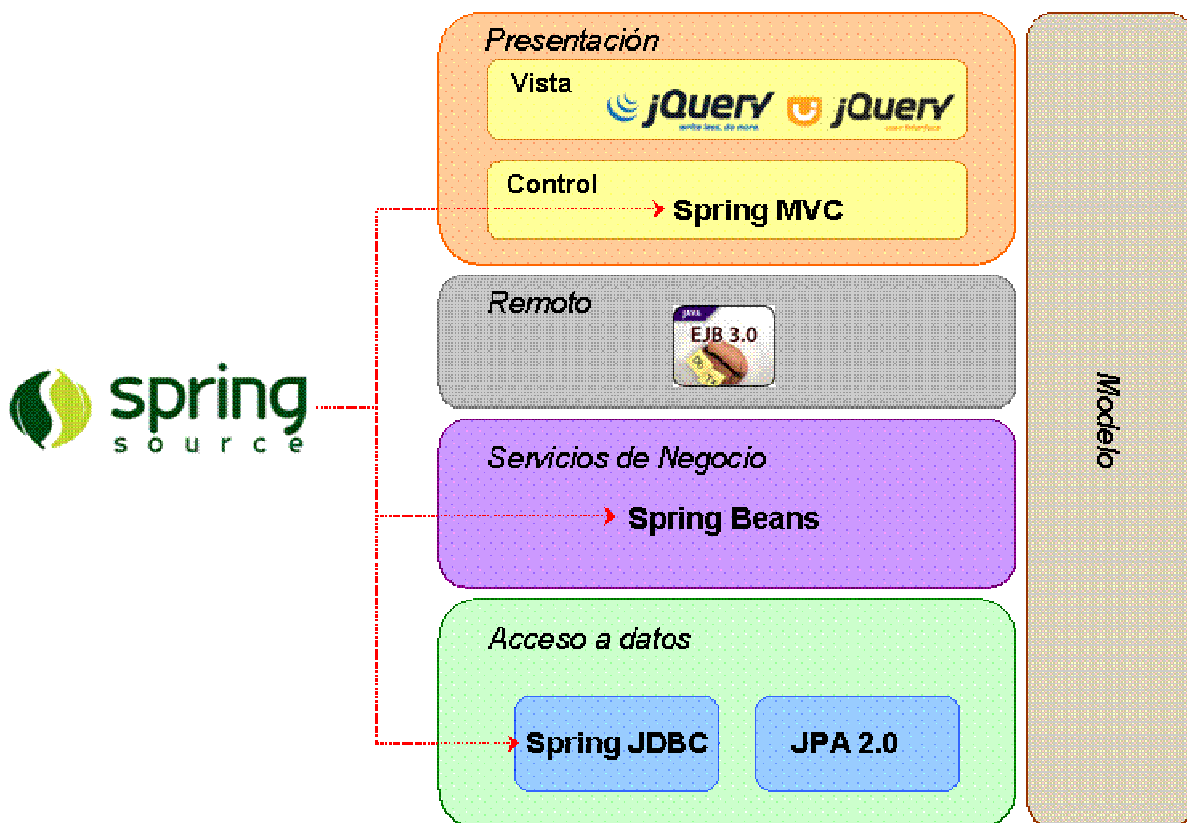
1 Introducción

UDA es un conjunto de herramientas, tecnologías, componentes y normativas funcionales y técnicas que permiten acelerar y normalizan el proceso de construcción de aplicaciones JEE en el ámbito de Gobierno Vasco – EJIE

- Determina la arquitectura conceptual básica de los nuevos sistemas JEE, así como la selección y configuración de tecnologías que implementan sus componentes
- Potencia la ayuda al desarrollo, posibilitando mediante herramientas la generación de forma automatizada y asistida del código de las nuevas aplicaciones a construir, en coherencia con la arquitectura definida
- Proporciona componentes reutilizables y adaptables que implementan patrones de interacción para la construcción de aplicaciones de Internet enriquecidas (RIA)

2 Arquitectura Tecnológica

Uno de los principios básicos en la definición de toda arquitectura para posibilitar gran parte de los requisitos establecidos (Extensibilidad, mantenibilidad, facilidad de manejo) es sin duda alguna la división de responsabilidades o la creación de módulos con una finalidad en concreto, evitando mezclar diferentes aspectos de la creación. A su vez, la arquitectura tecnológica está basada en estándares.



Se han seleccionado una serie de tecnologías para utilizar en los diferentes módulos, que se detallan a continuación:

- En la capa Presentación utiliza componentes RIA
 - Cliente: JavaScript, CSS y AJAX
 - Vista: jQuery + jQuery UI → Componentes RIA [JSON]
 - Servidor: JSP
 - Controlador: Spring MVC
- Modelo
 - POJOs que definen entidades
- Remoto:

- EJB 3.0 para la comunicación remota
- Implementación mediante interfaces remotas
- Wrappers para el despliegue
- Servicios de negocio:
 - POJOS para la implementación
 - Fachada para las peticiones
 - Locales
 - Remotas
- Acceso a datos
 - POJOS para la implementación
 - APIs de persistencia
 - JDBC → SpringJDBC
 - JPA 2.0

Además, la arquitectura de UDA se caracteriza por utilizar

- Componente de funciones comunes (x38)
 - Se integra en las aplicaciones
 - Solventa requerimientos estructurales de Ejje
 - Podrá evolucionar según se detecten nuevos requerimientos
 - Bajo acoplamiento con las aplicaciones desarrolladas con UDA (AOP)
 - Funcionalidades incluidas:
 - Seguridad (incluye adaptador XLNetS)
 - Validación (facilita validación de campos en servidor)
 - Trazas (aplica formato estándar para Bitácora)
 - Utilidades y complementos
- Apache Maven (resolución de dependencias / librerías)
 - Librerías a nivel de EAR [pom.xml]

3 Patrones de interacción

UDA ofrece una serie de componentes (RUP) que implementan un grupo de patrones de interacción de usuario identificados.

Cada componente implementado está orientado a cubrir casos de uso concretos para los cuales se ha diseñado. El uso de dichos patrones de interacción está destinado a homogeneizar en lo posible la capa de presentación de las aplicaciones.

Los patrones implementados actualmente son:

- **Feedback:** Canal informativo por el que se comunica al usuario los posibles errores o problemas acaecidos, los resultados de su interacción con los elementos de la aplicación y las posibles medidas a tomar para solucionar ciertas inconsistencias
- **Mensaje:** Tiene como objetivo mostrar al usuario de forma homogénea, clara y llamativa, los posibles mensajes que pueden desencadenar las acciones en la aplicación. Estos mensajes predefinidos pueden ser de diferente tipología: error, confirmación, aviso o alerta
- **Diálogo:** Permite lanzar un subproceso o un mensaje de confirmación dentro de un proceso principal sin salirse de este. Es una evolución del patrón mensaje
- **Idioma:** El componente de idioma está diseñado para permitir al usuario elegir de forma intuitiva el idioma en el que se le presenta la aplicación
- **Menú:** Menú de la aplicación mantenido a lo largo de todas las páginas de forma consistente que muestra entradas directas a secciones clave de la aplicación
- **Migas de pan:** Muestra a los usuarios los pasos de navegación seguidos durante su interacción con la aplicación y les facilita enlaces para volver a páginas precedentes o la página de inicio
- **Combo:** Permite al usuario recuperar un elemento de una gran lista de elementos o de varias listas dependientes de forma sencilla y ocupando poco espacio en la interfaz
- **Autocomplete:** En cuanto el usuario comienza a escribir una búsqueda se le sugieren búsquedas relacionadas con lo que ha escrito que pueden ser de su interés
- **Fecha:** Permite al usuario introducir y seleccionar una fecha, tanto de forma manual como visual, moviéndose fácilmente por días, meses y años. Además, para minimizar las posibilidades de introducir una fecha incorrecta, ofrece al usuario ayudas y sugerencias de formato
- **Hora:** El usuario puede introducir y seleccionar una hora tanto de forma manual como visual, moviéndose fácilmente por las horas y los minutos, recibiendo ayudas y sugerencias para minimizar las posibilidades de introducir una hora incorrecta
- **Botonera:** Se les presenta a los usuarios una barra de botones con diversas funcionalidades relacionadas con elementos de la página. Gracias a este componente se presentan, ordenan y agrupan las distintas funcionalidades gestionadas por las aplicaciones.
- **Tabla:** Se les presenta a los usuarios los datos tabulados para que la información se visualice de manera ágil y rápida, facilitando así su comprensión y manejo
- **Mantenimiento:** Implementa un nuevo patrón definido para facilitar la lógica necesaria en las acciones básicas, denominadas CRUD (create, read, update y delete), sobre una tabla

- Pestañas: Permiten dar acceso de forma compacta a grupos de contenidos mutuamente excluyentes pudiendo ser integradas en zonas muy reducidas de la interfaz
- Tooltip: Todas las aplicaciones deben tener un sistema de ayuda aunque algunas necesiten uno más completo y preciso que otras. El objetivo del sistema de ayuda debe ser, por un lado ayudar al impaciente y/o al usuario ocasional tan extensamente como sea posible y, por otro, ayudar a los usuarios expertos

4 Generadores de código

UDA provee asistentes para generación de proyectos, código (en base al modelo relacional) y de vista (interfaces de usuario), mediante plugins del IDE Eclipse:

- Crear una nueva aplicación (estructura de proyectos)
- Generar código fuente aplicación (operaciones CRUD)
- Definir un mantenimiento básico (sobre una entidad de BD)
- Añadir proyectos a la aplicación:
 - Proyectos de aplicaciones web (WAR)
 - Proyectos de negocio (EJB)
- Generación de interfaces de comunicación con otras aplicaciones:
 - Cliente
 - Servidor

5 Kit de desarrollo

UDA se presenta como un Kit de desarrollo, el cual contiene:

- Eclipse Helios OEPE
- Apache Maven (resolución de dependencias / librerías)
- Plugin UDA:
 - Plugin Eclipse
 - Plantillas (templates)
- Reglas de validación de la calidad del código:
 - PMD / Checkstyle / FindBugs

Para todo esto se requiere como software de base:

- Servidor de aplicaciones Weblogic 11g
- Java 6
- Acceso a Base de Datos Oracle 10g
- Repositorio SVN