



# Introducción a Javato

Fº. Javier Pereñiguez

Steria Iberica

20/02/2008



# Índice

- Introducción
- Arquitectura
  - Ejemplo arquitectura
- Plataforma
- Desarrollo
  - Ejemplo de entorno de desarrollo
- Vías futuras
- Casos de éxito



# Introducción

- **JAVATO** es un framework libre para entornos Web, basado en tecnología JEE.
- Con el desarrollo de este framework se pretendía:
  - Todos los proyectos Java siguieran la misma estructura y comportamiento
  - Compartir y reutilizar código entre proyectos
  - Facilitar el mantenimiento y extensión de los proyectos
  - Agilizar los desarrollos
  - Aumentar la productividad

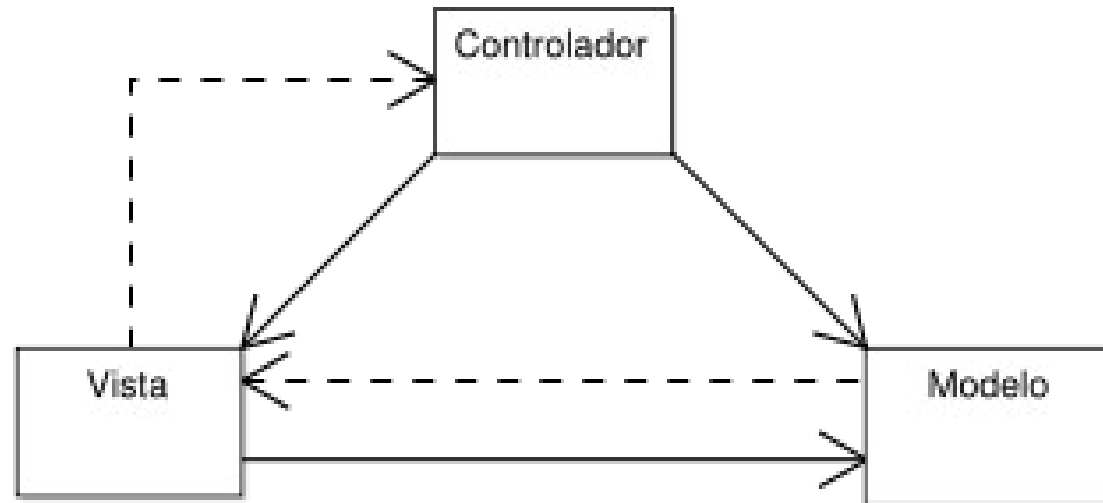


# Arquitectura (i)

- Javato esta basado en el patrón Modelo Vista Controlador (MVC)
  - **Modelo:** Esta es la representación específica de la información con la cual el sistema opera
  - **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario
  - **Controlador:** Este responde a eventos, usualmente acciones del usuario
  
- En terminología Web:
  - **Modelo:** es el Sistema de gestión de BBDD
  - **Vista:** son las paginas Web
  - **Controlador:** representa la lógica de negocio



# Arquitectura (ii)





## Arquitectura - beneficios (i)

- Menor acoplamiento
  - Desacopla las vistas de los modelos.
  - Desacopla los modelos de la forma en que se muestran e ingresan los datos.
- Mayor cohesión.
  - Cada elemento del patrón esta altamente especializado en su tarea
- Las vistas proveen mayor flexibilidad y agilidad.
  - Se puede crear múltiples vistas de un modelo.



## Arquitectura – beneficios (ii)

- Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos y canales
  - Una vista para cada dispositivo que puede variar según sus capacidades
- Más claridad de diseño
- Facilita el mantenimiento
- Mayor escalabilidad



## Arquitectura – capas MVC

- En Javato cada capa del patrón MVC esta implementada con un framework
  - El modelo esta ligado a Hibernate
  - La vista esta ligada a JSF
  - Y el controlador a Spring
- Cada uno de estos framework esta especializado o se amolda perfectamente a la capa con la que se ha ligado en Javato





# Arquitectura - Hibernate

- ORM, Mapeo Objeto Relacional
- Configuración y mapeos sencillos (ficheros XML)
- Lenguaje de consultas orientadas a objetos (HQL)
- Permite SQL nativo
- Libera al desarrollador de tareas de programación de persistencia relacional
- Se integra con la mayoría de las BBDD conocidas
- Soporta XDoclet (generación automática)
- Gestión de transacciones
- Pooling de conexiones



# Arquitectura - Spring

- Framework para desarrollo de aplicaciones Web
- Bien organizado (Modular)
- Facilidades para integrarlo con otros frameworks
- Proporciona un completo contenedor de peso ligero
- Incorpora una implementación de patrón MVC
- Proporciona una capa de abstracción de JDBC
- Incluye funcionalidad de Programación Orientada a Aspectos (AOP)
- Basado en “best-practices” aceptadas por la industria

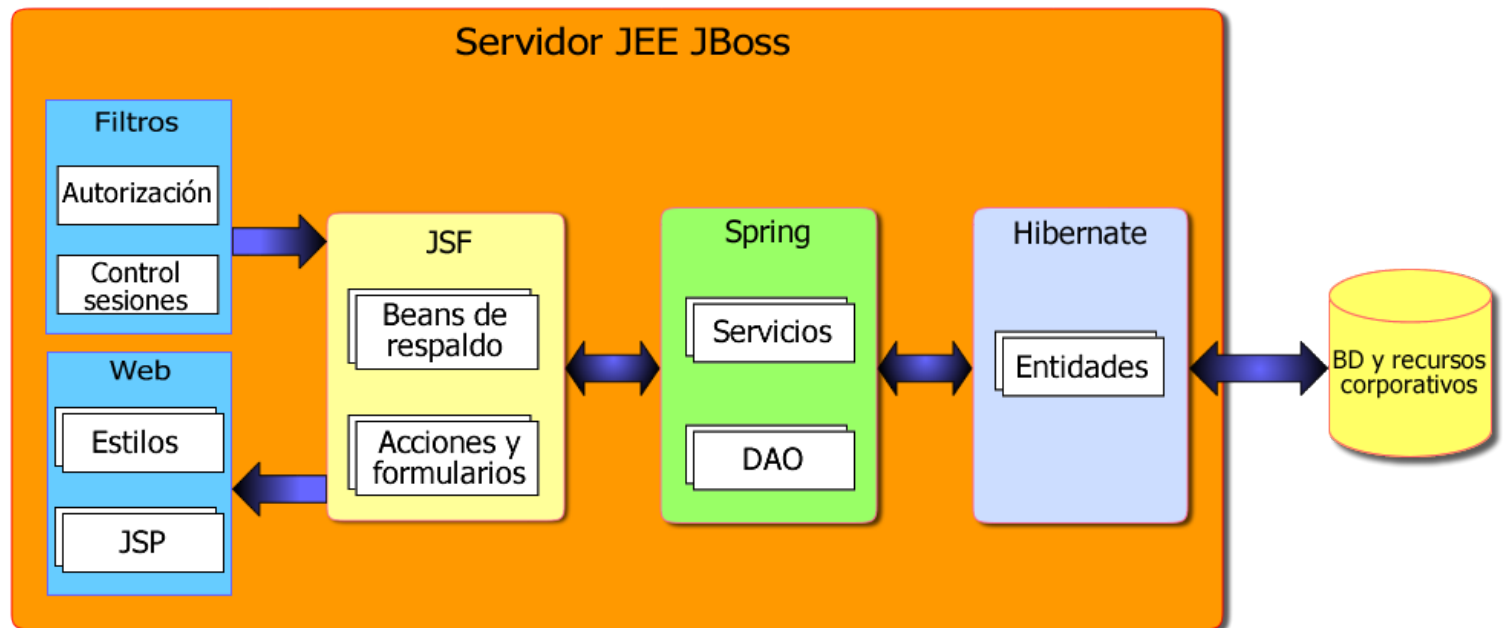


# Arquitectura - JSF

- Es una especificación, existen diferentes implementaciones (Myfaces, Trinidad, RichFaces, etc.)
- Basado en el patrón de diseño Modelo-Vista-Controlador
- Basado en componentes (fácil integración con IDE)
- Flexible (permite crear nuestros propios componentes)
- Orientado al desarrollo visual rápido (drag & drop)
  - Poco Maduro
- Generación automática de presentación en función del navegador cliente
- Generación automática de presentación cumpliendo estándares de Accesibilidad (WAI)



# Ejemplo arquitectura



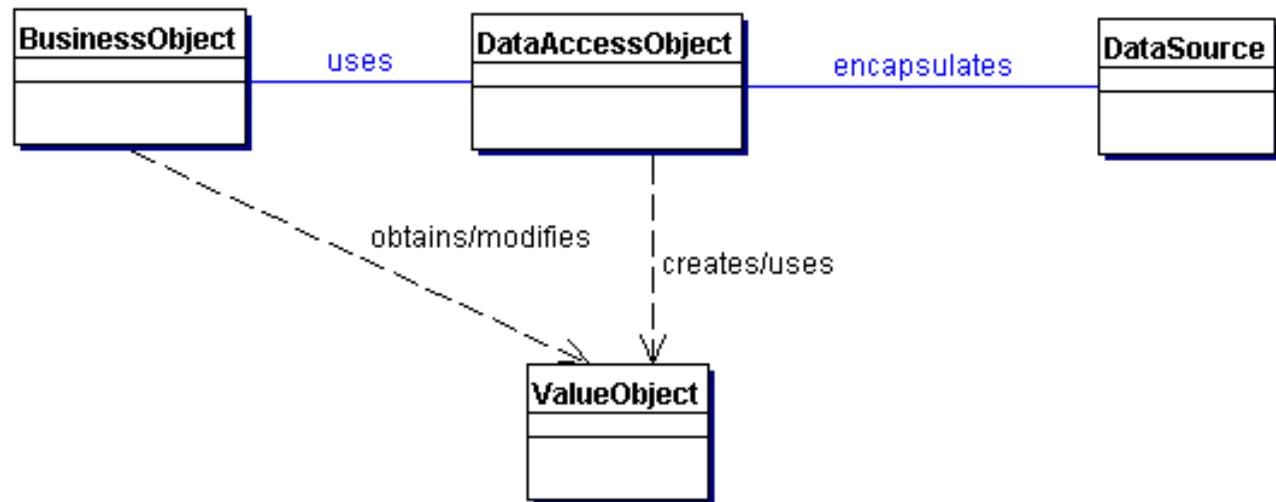


## Arquitectura – patrón DAO (i)

- DAO oculta completamente los detalles de implementación de la fuente de datos a sus clientes
- El interface expuesto por DAO no cambia cuando cambia la implementación de la fuente de datos subyacente

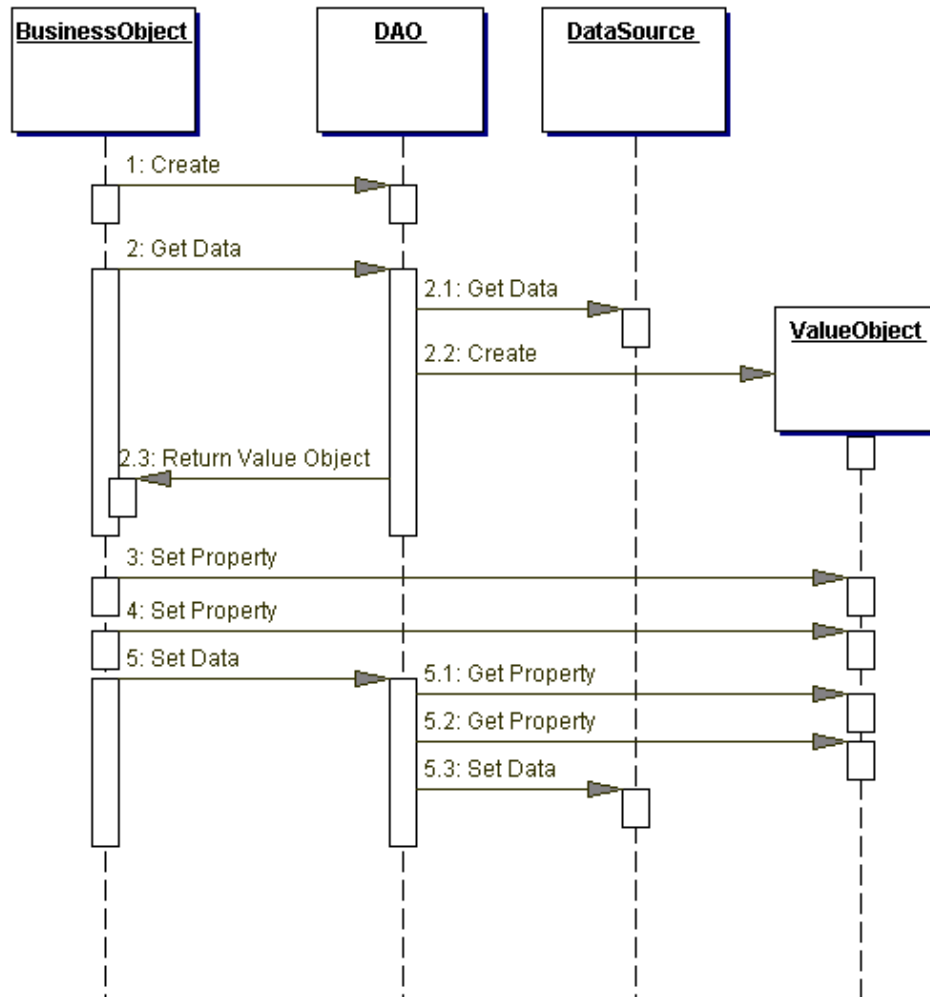


# Arquitectura – patrón DAO (ii)





# Arquitectura – patrón DAO (iii)





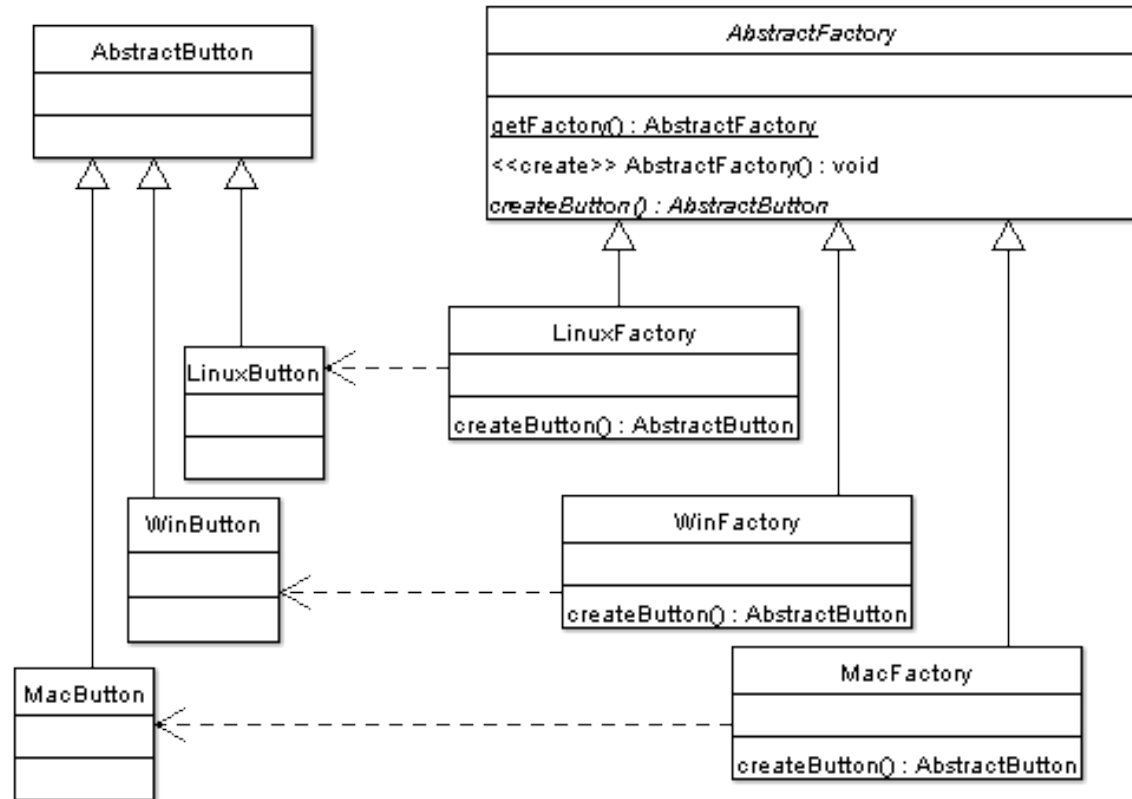
## Arquitectura – patrón DAO (iV)

- Como mejorar el patrón DAO
- Utilizando los patrones de creación abstract factory y method factory
- Se utiliza un factoría abstracta de DAO y tantas factorías concretas como implementaciones distintas tengamos
- Cada método de la factorías concretas a su vez es una implementación del patrón method factory



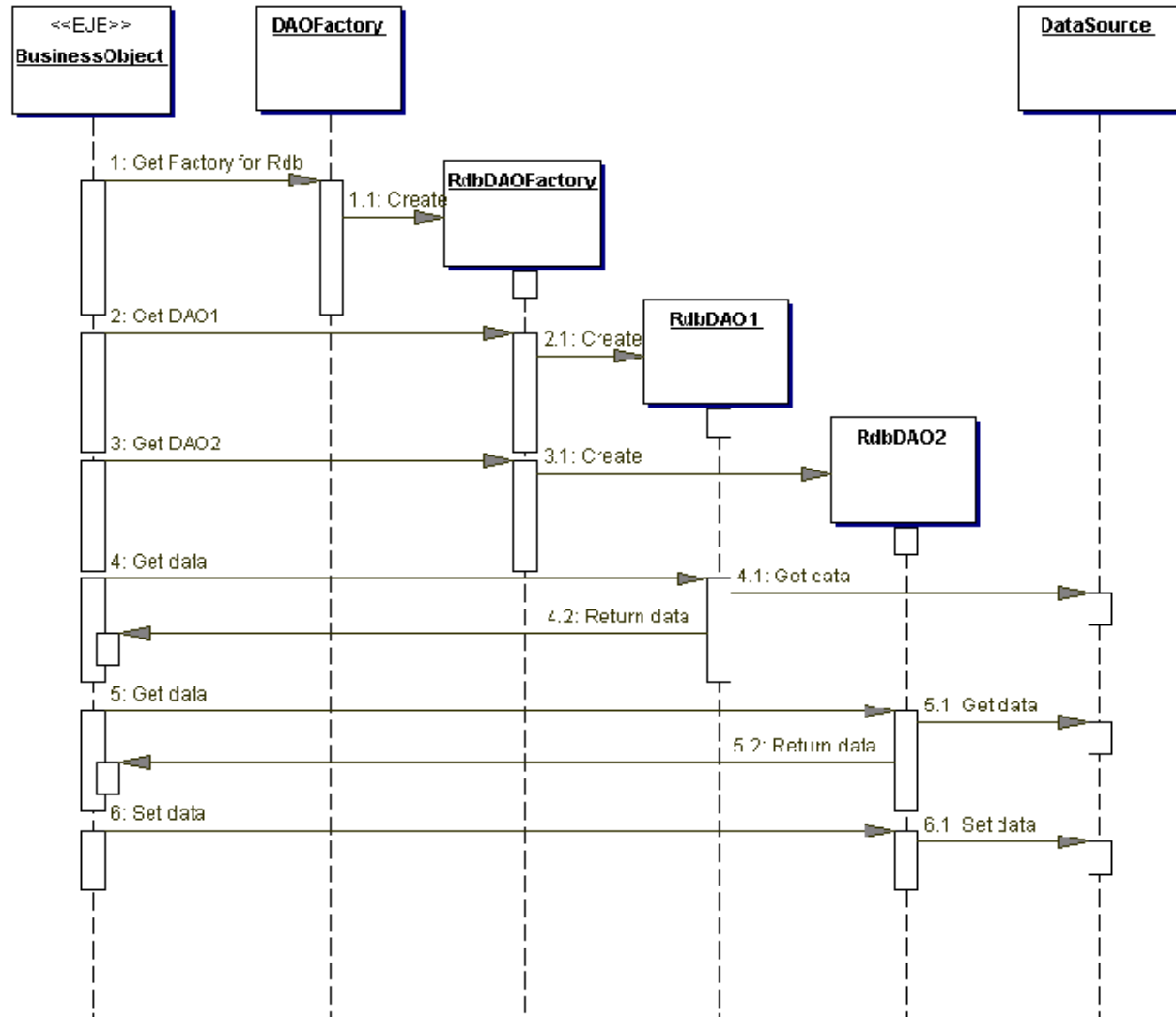


# Arquitectura – patrón DAO (v)





# Arquitectura – patrón DAO (vi)





# Arquitectura – patrón DAO (vii)

- Beneficios patrón DAO
  - Transparencia, los objetos de negocio utilizan la fuente de datos sin conocer los detalles de su implementación
  - Fácil migración, a otra fuente de datos (gracias a los patrones abstract factory y method factory)
  - Reduce la complejidad, como manejan todas las complejidades del acceso a los datos, se simplifica el código de los objetos de negocio
  - Fácil de mantener, toda la funcionalidad de acceso a datos esta localizada en un sitio concreto



# Índice

- Introducción
- Arquitectura
  - Ejemplo arquitectura
- **Plataforma**
- Desarrollo
  - Ejemplo de entorno de desarrollo
- Vías futuras
- Casos de éxito



# Plataforma

- Javato, además del framework de desarrollo, se apoya en una serie de aplicaciones y herramientas
- Se intenta abarcar todo el ciclo desarrollo de una aplicación (desde creación hasta el despliegue en producción)
- El uso de la plataforma es opcional
- Proporciona muchas funcionalidades a las aplicaciones



## Plataforma – BBDD Javato

- Proporciona un script para la creación de una BBDD
- Esta BBDD dará respaldo a las aplicaciones Webcomun y Traspaso entre entornos



## Plataforma – Webcomun (i)

- Es una aplicación web
- Gestiona todas las aplicaciones Javato que quieran utilizar la plataforma (incluyéndose ella misma)
- Utiliza la BBDD Javato
- Esta implementada utilizando la arquitectura Javato y la plataforma



## Plataforma – Webcomun (ii)

- Utilidades que ofrece:
  - Gestión de aplicaciones
  - Gestión de menús
  - Gestión de ayuda
  - Gestión de usuarios
  - Gestión de roles
  - Gestión de permisos
  - Gestión de entornos
  - Gestión de informes
  - Motor de informes
  - Visor de Logs





# Plataforma – Traspaso

- Es una aplicación web
- Utiliza la BBDD Javato
- Esta implementada utilizando la arquitectura Javato y la plataforma
- Proporciona la 2 funcionalidades:
  - Traspaso de ejecutables: Hace despliegues de aplicaciones a los entornos creados en Webcomun
  - Traspaso de BBDD: Lanza script de BBDD a los entornos creados en Webcomun



## Plataforma – Cruise Control

- Herramienta de Integración Continua
- Se apoya en Ants
- Se integra con CVS
- Genera informes de los cambios entre versiones
- Mantiene la lista de artefactos generados
- Fácil configuración (ficheros XML)
- Utiliza tecnologías distribuidas (RMI y HTML)
- Utiliza un contenedor de Servlet (Jetty)



## Plataforma – Apache Ants

- Herramienta para la construcción automática de software (Make)
- Compila el proyecto
- Copiar ficheros de recursos al directorio de clases
- Borrar el directorio de clases y temporales
- Empaqueta un proyecto en fichero (jar, war, etc.)
- Despliega los ficheros de ejecución al servidor



## Plataforma - XRadar

- Herramienta que evalúa la calidad del software generado
- Basado en métricas (dependencias, violaciones de código, etc.)
- Arquitectura basada en Java y XML
- Genera gráficos para controlar el estado del proyecto
- Exporta los informes en formato HTML y SVG



# Plataforma - Mantis

- Herramienta colaborativa
- Implementa un sistema gestor de incidencias
- Notificación por correo electrónico
- Integración con CVS
- Independiente del Sistema Operativo
- Integración con Wiki
- Soporta múltiples gestores de BBDD
- Histórico de cambios
- Permite exportar informes a CSV, Microsoft Word y Microsoft Excel
- Sencillo de instalar y intuitiva de utilizar



## Plataforma - Birt

- Gestor de Informes Business Intelligence
- Tiene 2 componentes: un diseñador de informes y un entorno de ejecución (runtime)
- Tiene un API muy rica
  - De gestión de BBDD
  - De diseño de informes
- Integrado totalmente con el IDE Eclipse
- Creación y utilización de plantillas
- Proporciona una gran variedad de gráficos



## Plataforma – CVS

- Servidor de control de versiones
- Repositorio de versiones
- Guarda registros de cambios
- Posibilita la colaboración
- Permite editar el código de forma simultanea
- Asume la tarea de integrar todos los cambios
- Guarda un registro de todos los conflictos



# Índice

- Introducción
- Arquitectura
  - Ejemplo arquitectura
- Plataforma
- **Desarrollo**
  - Ejemplo de entorno de desarrollo
- Vías futuras
- Casos de éxito





## Desarrollo - Eclipse

- **Eclipse** es una plataforma de software de código abierto independiente de la plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido"
- Hecho en Java
- Fácil instalación (solo descomprimir)
- Extensible a través de Plugins
  - WTP, plugin para el desarrollo de aplicaciones web

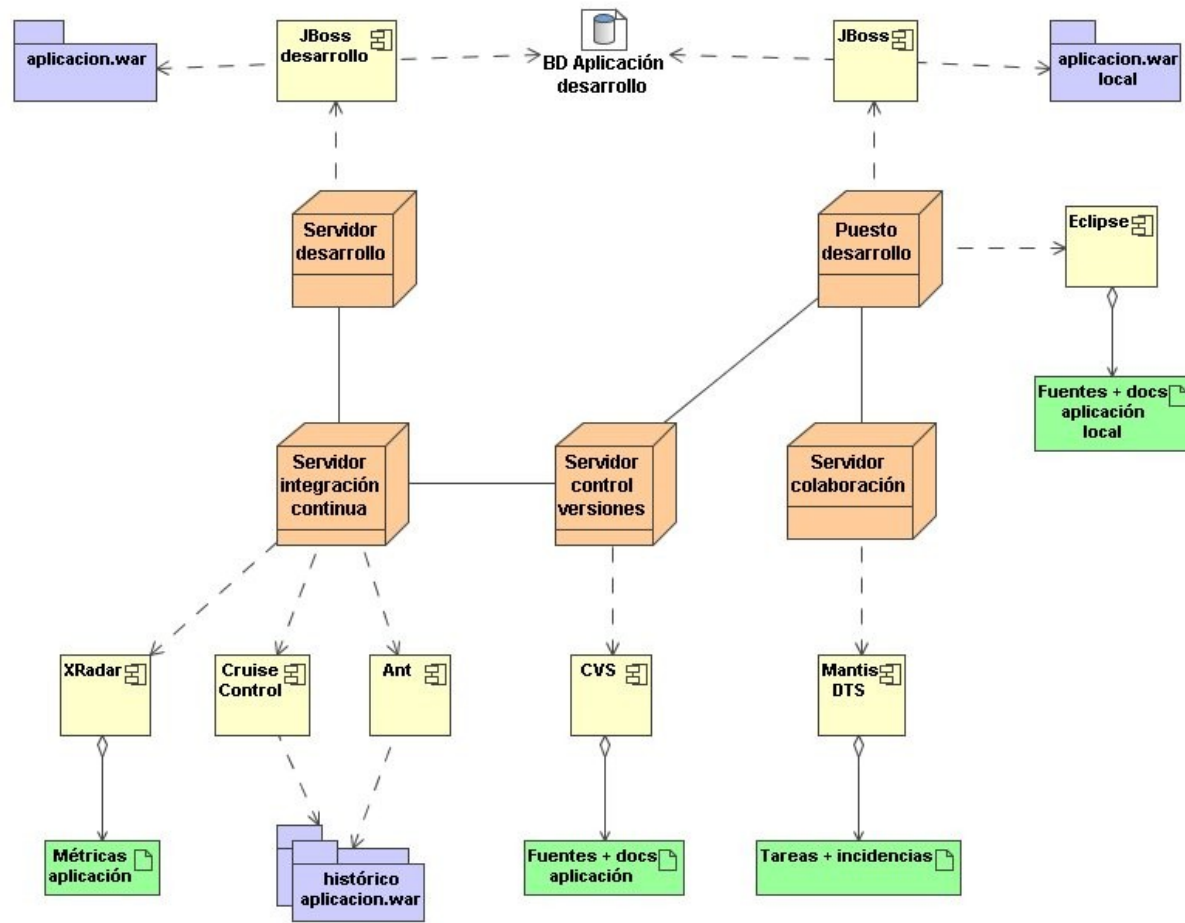


## Desarrollo - Semilla

- Es una aplicación que tiene configurados los 3 framework (Hibernate, Spring y JSF) para que trabajen de forma sincronizada. También tiene configuradas todas las posibilidades que ofrece Javato
- Es una plantilla
- Es el punto de partida para cualquier desarrollo
- Es una aplicación completa, es decir, si la desplegamos en un servidor de aplicaciones funciona



# Ejemplo de un entorno de desarrollo





# Índice

- Introducción
- Arquitectura
  - Ejemplo arquitectura
- Plataforma
- Desarrollo
  - Ejemplo de entorno de desarrollo
- **Vías futuras**
- Casos de éxito



# Vías futuras – la vista

## ■ RichFaces

- Implementación JSF que incluye funcionalidades de AJAX
- Componentes visuales

## ■ Facelets

- Es un sistema de plantillas-vistas para JSF
- Es como Tiles para Struts
- Divide el interfaz de usuario (pantalla) en regiones y a cada región se le asocia una página JSF



## Vías futuras – el modelo

- JPA (Java Persistence API)
  - Es el API de persistencia desarrollado para la plataforma Java EE
  - Busca estandarizar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional
  - Existen varias implementaciones: Hibernate, Toplink, etc.



## Vías futuras – Jasper Report

- El framework Javato proporciona una interfaz para un motor de informes genérico
- Actualmente incorpora una implementación para el motor de informes **Birt**
- Se pretende incorporar una nueva implementación para el motor de **Jasper Report**
- El usuario podrá elegir entre la implementación **Birt** y **Jasper Report**



# Índice

- Introducción
- Arquitectura
  - Ejemplo arquitectura
- Plataforma
- Desarrollo
  - Ejemplo de entorno de desarrollo
- Vías futuras
- **Casos de éxito**





## Casos de éxito

- FAPI, aplicación para el control de fabricación de pienso para animales. Consejería de Agricultura.
- Portal de la imprenta regional.
- FCT, aplicación para formación en centros de trabajo. Consejería de Educación.
- Webcomun y Traspaso entre entornos, aplicaciones para la gestión de aplicaciones Javato. Dirección General de Informática



# Introducción a Javato

Fº. Javier Pereñiguez

Steria Iberica

20/02/2008