

# *Trabajo fin de carrera*

UOC

**José María González Vázquez**

Enero 2011

---

---

# *Presentación*

- Contexto del problema
  - Motivación del proyecto
  - Introducción a la gestión de contenidos
  - Introducción a Spring
  - Patrones de diseño utilizados
  - Funcionamiento de la aplicación
  - Posibles ampliaciones
  - Conclusiones
- 
-

# *Contexto del problema*

- Empresa para la comercialización de productos alimentarios
  - Amplia cartera de clientes y productos
  - Necesidad de mantener una información precisa y actualizada
  - Pequeña empresa sin personal informático
  - Productos de temporada o que solo se dan un año (p. ej. un vino de una añada específica)
- 
-

# *Motivación del proyecto*

- Dotar de una herramienta de publicación de contenidos
  - Estudio de Spring como base para una aplicación en un entorno web
  - Estudio del Modelo-Vista-Controlador en Spring (SpringMVC)
  - Estudio de la persistencia en Spring (SpringJDBC)
- 
-

# *Introducción a la gestión de contenidos (I)*

## ⇒ Modelo clásico de desarrollo web

- Cada página con contenidos es única
  - Uno o varios desarrolladores crean cada página a petición de uno o varios creadores de contenidos
  - Las modificaciones de contenido pueden necesitar la intervención de desarrolladores
  - No hay una separación clara entre contenidos y presentación
  - Coste de actualización alto al necesitar trabajadores cualificados para el mantenimiento de la aplicación
  - Complejidad en el mantenimiento del sitio web
- 
-

# *Introducción a la gestión de contenidos (II)*

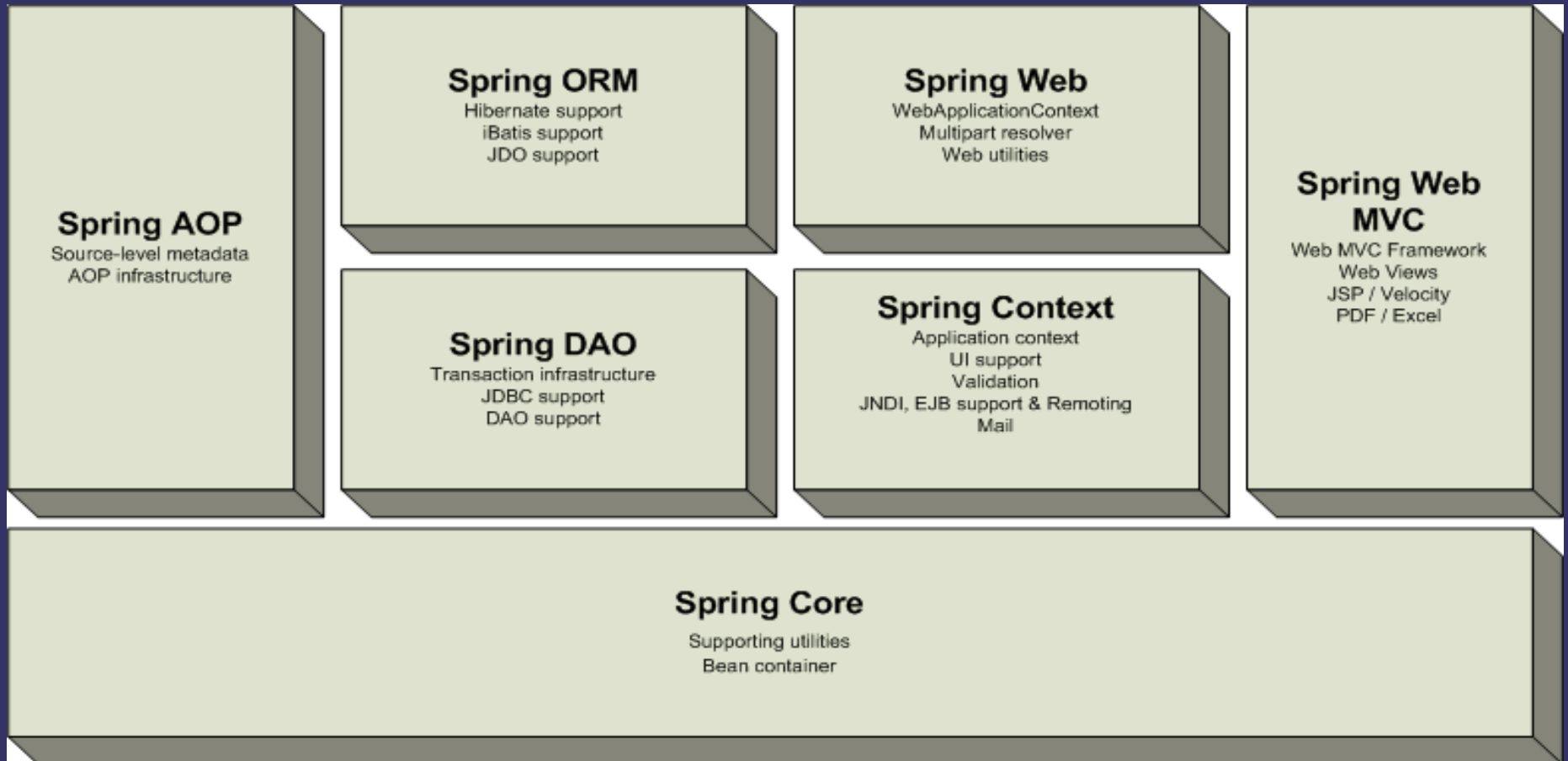
## ⇒ Modelo de gestión de contenidos

- Separación de contenidos y presentación
- El editor de contenidos crea contenidos sin necesidad de la intervención de un desarrollador
- Los contenidos se actualizan, publican y despublican según la evolución de las necesidades del sitio web
- Se delimitan mejor los roles:
  - Desarrolladores
  - Creadores de contenido
  - Administradores de la aplicación
  - Revisores de contenido
  - Etc...

# *Introducción a Spring (I)*

- La primera versión data del 2002
  - Se ha convertido en un estándar de la industria
  - 9 de 10 bancos más rentables utilizan Spring (fuente SpringSource)
  - Posee una comunidad de desarrolladores muy activa
  - Dividido en módulos. No es necesario utilizarlos todos, tan solo los que mejor se adecuen al proyecto
- 
-

# Introducción a Spring (II)

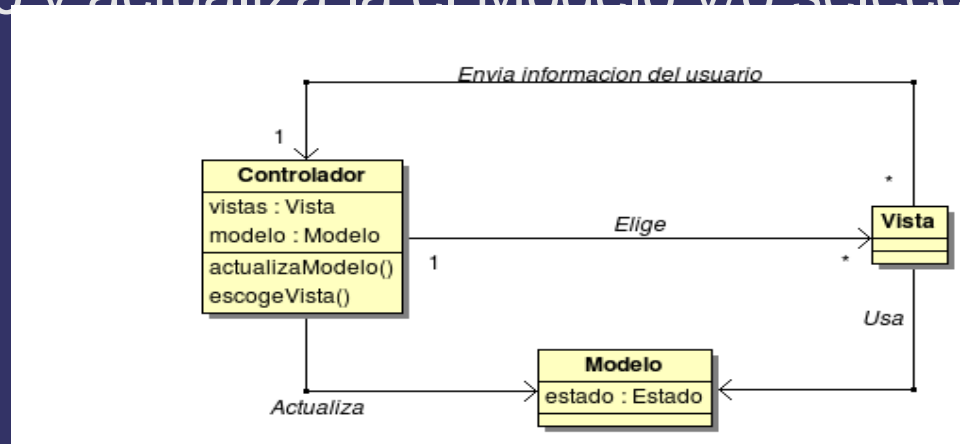




# Patrones de diseño utilizados (I)

## ⇒ Modelo-Vista-Controlador

- Se divide la problemática de la interfaz en tres partes
- El **Modelo** guarda el estado de la aplicación
- La **Vista** interpreta los datos del modelo y los presenta al usuario
- El **Controlador** procesa la información introducida por el usuario y actualiza la el Modelo y/o selecciona la Vista



# *Patrones de diseño utilizados (II)*

- ⇒ Patrón DAO (Data Access Object)
    - Este patrón tiene como objetivo simplificar la lógica de los recursos de acceso a la base de datos
    - Se utilizan interfaces en lugar de clases, de esta manera se hace una abstracción del acceso a base de datos
    - Se pueden cambiar los métodos de acceso a la base de datos sin que influya en el resto de la aplicación
    - Spring se ocupa del mapeado de objetos, así como de la inyección de objetos y propiedades de acceso a la base de datos
- 
-

# *Funcionamiento de la aplicación (I)*

## ⇒ Consola de administración

- El Administrador gestiona productos, canales, proveedores, clientes y pedidos
- El administrador publica productos y canales

## ⇒ Sitio Web

- Al construir la web se pasa como parámetro el id de un canal. Si el canal está publicado se buscan todos los productos publicados asociados al canal y se muestran en la web

