

Arquitectura Java

Clase “3”

Agenda:

- **Arquitectura Java.**
- **Distintos tipos de Arquitecturas.**
- **Arquitectura legacy.**
- **Ejemplo**
- **Arquitectura actual**
- **Ejemplo**
- **Ejercicio**

Arquitectura

Introducción

- La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema.
- Generalmente, no es necesario inventar una nueva arquitectura de software para cada sistema de información.
- Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto.

En el mundo Java las arquitecturas mas conocidas son:

- MVC
- SOA
- Microservicios

En esta clase veremos ejemplos de MVC y dejaremos las otras dos para las próximas clases del curso.

MVC - Separación de capas
























- La programación por capas se refiere a un estilo de programación que tiene como objetivo separar la lógica de diseño de la lógica de negocios.
- Una de las ventajas que podemos destacar sobre este estilo es que el desarrollo del software se puede llevar a cabo en varios tipos de niveles, así, cuando suceda algún cambio solo nos iremos sobre el nivel requerido.

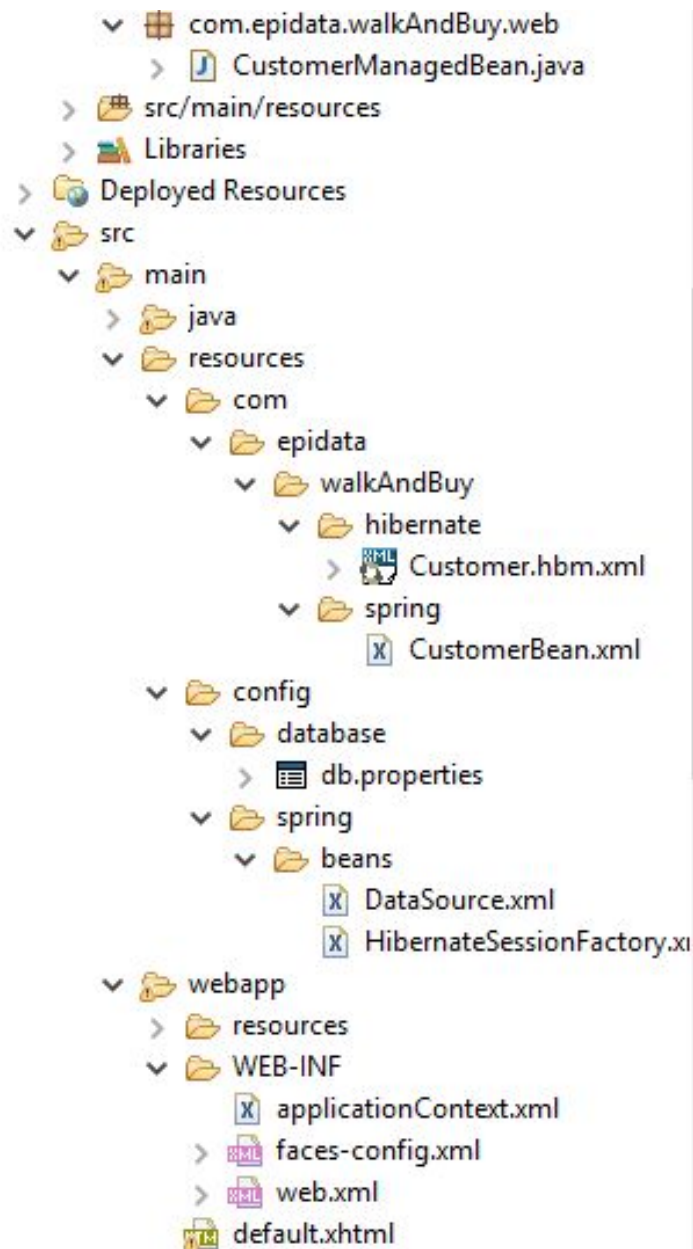
Programación por capas

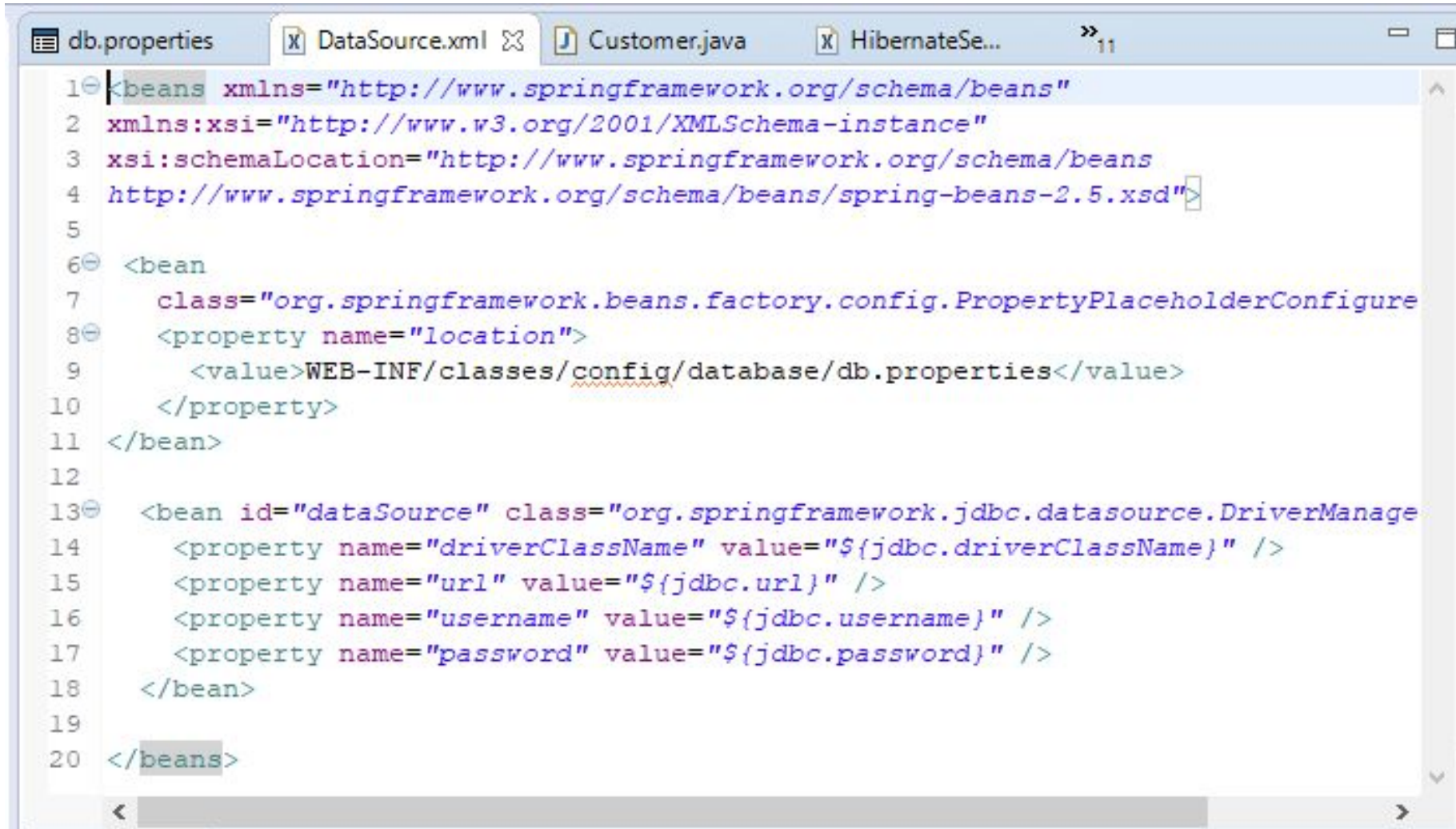
- En el diseño de sistemas informáticos actual se suelen usar las arquitecturas multinivel o programación por capas.
- En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

Ejemplo

- Vamos a ver una aplicación típica con dos arquitecturas distintas, una legacy basada en xml y otra en annotations.
- Vamos a utilizar JSF para la capa web, hibernate para base de datos y spring para la capa de negocios.

- ▼  JavaServerFaces
 - >  Deployment Descriptor: JavaServerFaces
 - >  JAX-WS Web Services
 - ▼  Java Resources
 - ▼  src/main/java
 - ▼  com.epidata.walkAndBuy.dao
 - >  CustomerDao.java
 - ▼  com.epidata.walkAndBuy.dao.impl
 - >  CustomerDaoImpl.java
 - ▼  com.epidata.walkAndBuy.model
 - >  Customer.java
 - ▼  com.epidata.walkAndBuy.service
 - >  CustomerService.java
 - ▼  com.epidata.walkAndBuy.service.impl
 - >  CustomerServiceImpl.java
 - ▼  com.epidata.walkAndBuy.web
 - >  CustomerManagedBean.java
 - >  src/main/resources
 - >  Libraries
 - >  Deployed Resources
 - >  src
 - >  target
 -  pom.xml





```
1 <beans xmlns="http://www.springframework.org/schema/beans"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://www.springframework.org/schema/beans
4     http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">
5
6   <bean
7     class="org.springframework.beans.factory.config.PropertyPlaceholderConfigure
8     <property name="location">
9       <value>WEB-INF/classes/config/database/db.properties</value>
10    </property>
11  </bean>
12
13  <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManage
14    <property name="driverClassName" value="${jdbc.driverClassName}" />
15    <property name="url" value="${jdbc.url}" />
16    <property name="username" value="${jdbc.username}" />
17    <property name="password" value="${jdbc.password}" />
18  </bean>
19
20 </beans>
```

Instrucciones para instalar el arquetipo:

- 1) Bajar el rar de descargas.
- 2) Descomprimir.
- 3) Ir por línea de comando y ejecutar mvn install.
- 4) Cuando termine el proceso, verificar si el arquetipo quedó instalado en maven.
- 5) Ir a la carpeta donde se quiera crear el proyecto.
- 6) Ejecutar: mvn archetype:generate
 - DarchetypeGroupId=ar.com.epidata.walkAndBuy
 - DarchetypeArtifactId=WalkAndBuy-archetype
 - DarchetypeVersion=0.0.1-SNAPSHOT
 - DgroupId=groupid -DartifactId=NombreProyecto -B

Ahora veremos lo mismo pero con una nueva arquitectura.

