

Especialista en Aplicaciones y Servicios Web con Java Enterprise



Alternativas a EJB

Sesión 3: Persistencia: introducción a HIBERNATE

INDICE



- **¿Por qué necesitamos Hibernate?**
- **Arquitectura Hibernate**
- **Configuración de Hibernate**
- **Resumen de pasos de configuración**

¿Por qué necesitamos Hibernate?



- **Programación orientada a objetos**

- Trata con objetos, atributos y relaciones



- **Uso de bases de datos relacionales**

- Trata con relaciones, tuplas y conjuntos



ORM: Object-Relational Mapping

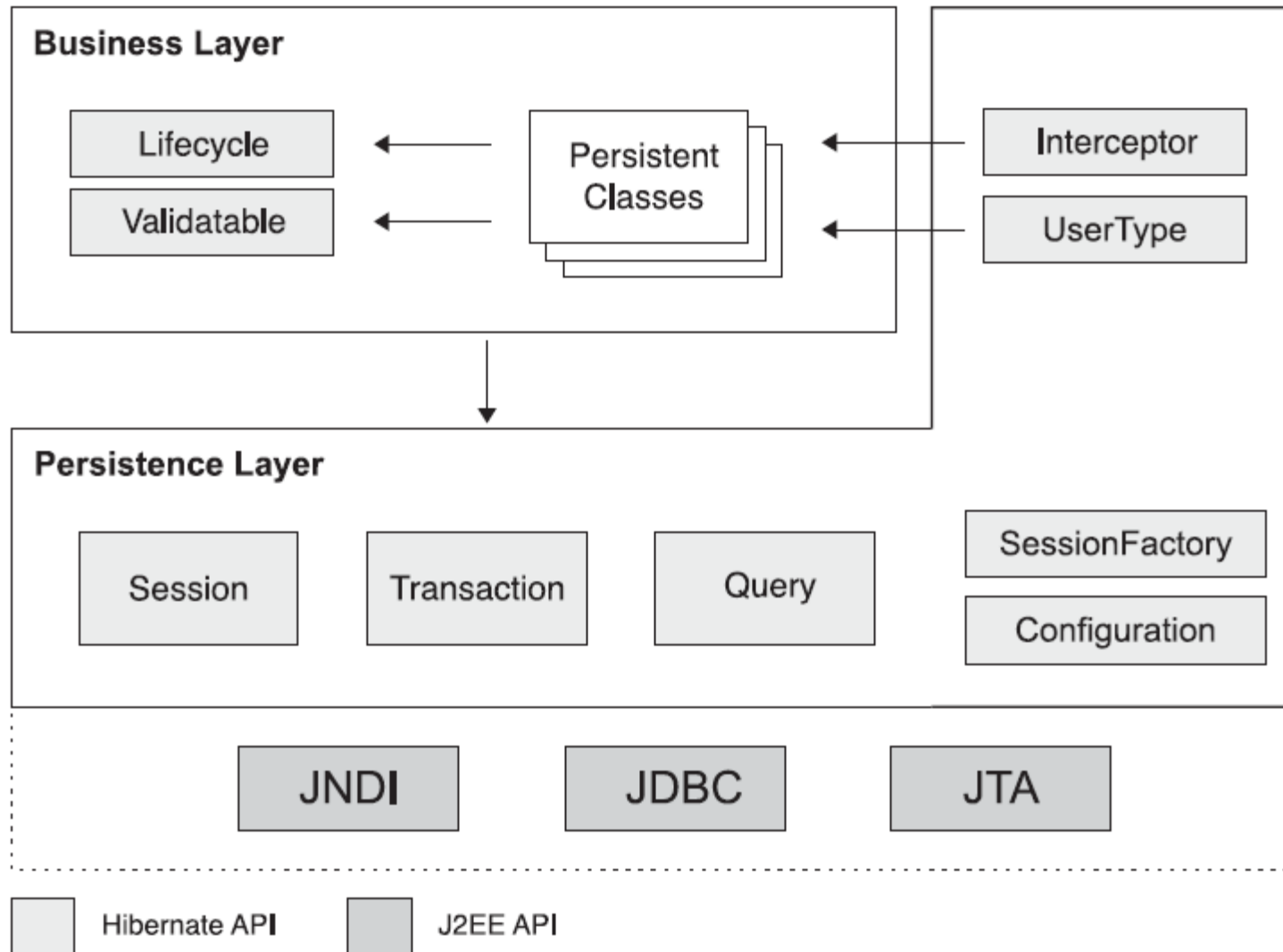
- **Problema:** un 35% del código de una aplicación para realizar la correspondencia $O \leftrightarrow R$
- **Solución:** utilizar una ORM, por ejemplo **Hibernate**

INDICE



- **¿Por qué necesitamos Hibernate?**
- **Arquitectura Hibernate**
- **Configuración de Hibernate**
- **Resumen de pasos de configuración**

Arquitectura Hibernate



INDICE



- **¿Por qué necesitamos Hibernate?**
- **Arquitectura Hibernate**
- **Configuración de Hibernate**
- **Resumen de pasos de configuración**

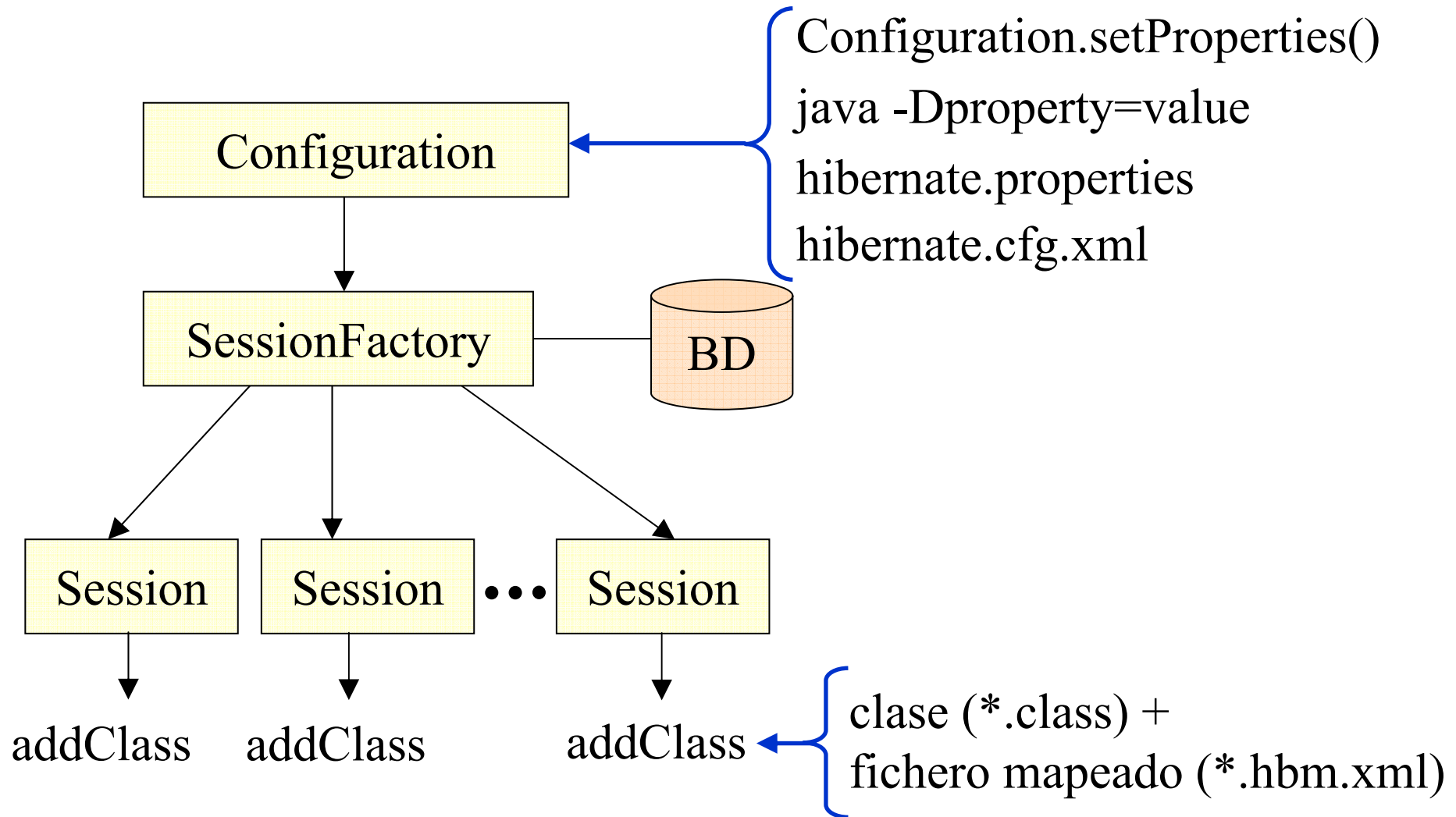
Configuración de Hibernate



- **Vamos a utilizar Hibernate en un entorno NO gestionado:**
 - **La propia aplicación gestiona las conexiones y transacciones**

- **Para utilizar hibernate necesitamos configurarlo**
 - **org.hibernate.cfg.**Configuration**:**
 - **Correspondencias entre tipos Java y tipos de una BD SQL**
 - **Propiedades de configuración (Consultar manual de referencia de Hibernate)**

Inicio de Hibernate





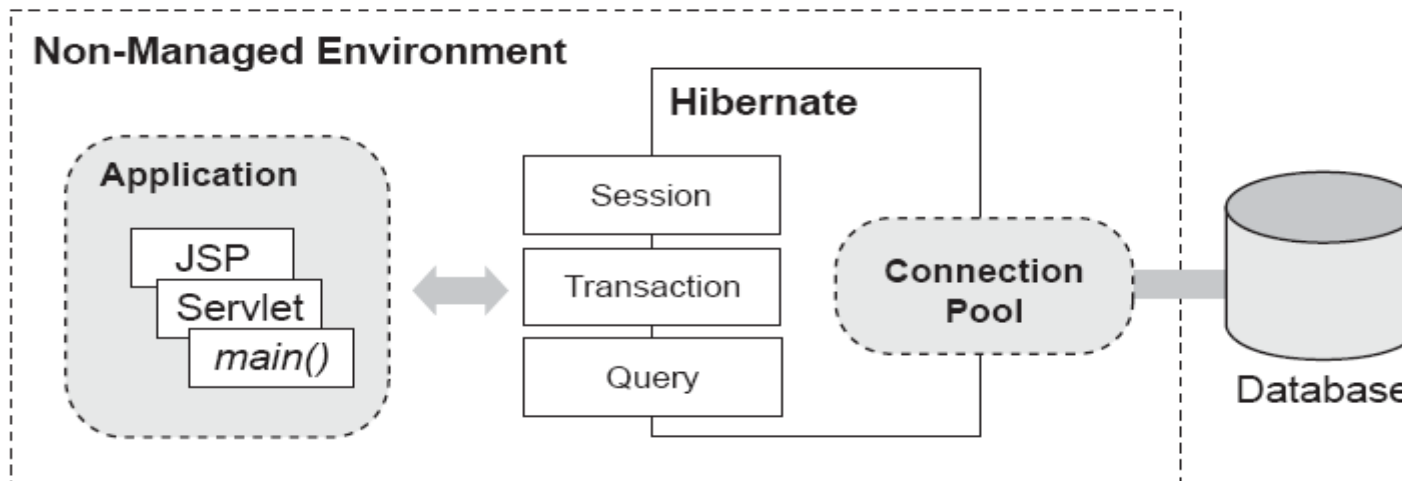
Ejemplos de creación de una SessionFactory

```
Configuration cfg = new Configuration();
cfg.addResource("hello/Message.hbm.xml"); cfg.setProperties
System.getProperties()); SesionFactory sessions =
cfg.buildSessionFactory();
```

```
SessionFactory sessions = new Configuration().
    addResource ("hello/Message.hbm.xml").
    setProperties (System.getProperties()).
    buildSessionFactory();
```

```
SessionFactory sessions = new Configuration().
    addClass(Message.class).
    setProperties(System.getProperties()).
    setProperty ("hibernate.dialect",
"org.hibernate.dialect.MySQLInnoDBDialect").
    setProperty("hibernate.connection.datasource",
"java:comp/env/jdbc/test").buildSesionFactory();
```

Configuración de la BD



- **Es conveniente utilizar un pool de conexiones (C3P0)**

➤ P.ej. librería C3P0

```
hibernate.connection.driver_class=org.postgresql.Driver
hibernate.connection.url = jdbc:postgresql://localhost/auctiondb
hibernate.connection.username = auctionuser
hibernate.conection.password = secret
hibernate.dialect = net.sf.hibernate.dialect.PosgreSQLDialect
hibernate.c3p0.min_size=5 hibernate.c3p0.max_size=20
hibernate.c3p0.timeout=300 hibernate.c3p0.max_elements=50
hibernate.c3p0.idle_test_period=3000
```

Fichero hibernate.cfg.xml



CABECERA

```
<hibernate-configuration>
  <session-factory name="java:/hibernate/HibernateFactory">
    <property name="show_sql">true</property>
    <property name="connection.datasource">
      java:/comp/env/jdbc/AuctionDB
    </property>
    <property name="dialect">
      net.sf.hibernate.dialect.PostgreSQLDialect
    </property>
    <property name="transaction.manager_lookup_class">
      net.sf.hibernate.transaction.JBossTransactionManagerLookup
    </property>
    <mapping resource="auction/Item.hbm.xml"/>
    <mapping resource="auction/Category.hbm.xml"/>
    <mapping resource="auction/Bid.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

Ejemplos



- **Uso de hibernate.hbm.xml**

```
SessionFactory sessions = new  
    Configuration().configure().buildSessionFactory();
```

- **Configuración de *Loggin***

- Ayuda a depurar los programas
- Necesita el fichero log4j.properties en nuestro *classpath*

INDICE



- **¿Por qué necesitamos Hibernate?**
- **Arquitectura Hibernate**
- **Configuración de Hibernate**
- **Resumen de pasos de configuración**

Resumen de pasos de configuración



- Situar el *.jar del *driver* JDBC elegido y el fichero hibernate2.jar en nuestro *classpath*
- Añadir las dependencias de Hibernate (directorio (lib) en el *classpath*. (lib/README.txt contiene una lista de librerías requeridas y opcionales).
- Elegir y configurar un *pool* de conexiones JDBC.
- Determinar las propiedades de Configuration en un fichero hibernate.properties en el *classpath*.
- Crear una instancia de Configuration en nuestra aplicación y cargar los ficheros de mapeado XML utilizando addResource() o addClass().
- Obtener una SessionFactory a partir de Configuration llamando a BuildSessionFactory().