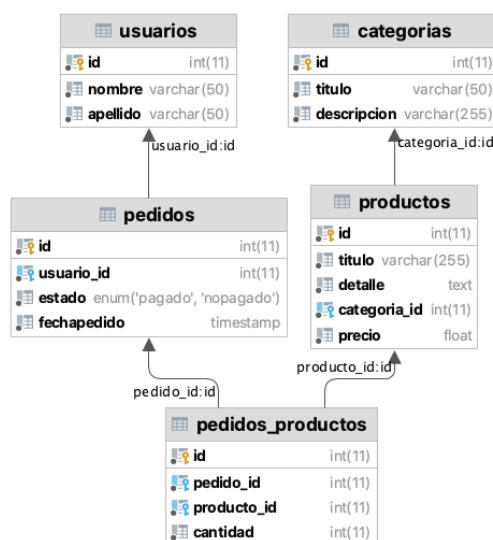


Nombre y Apellidos

EJERCICIO 1 TIENDA

El proyecto se enmarca dentro de una aplicación que será una tienda online tipo Amazon. Esta tienda tiene productos de distintas categorías. Los usuarios pueden hacer pedidos distintos, y cada pedido tendrá al menos un producto.

Teniendo eso en cuenta os presento el modelo de datos.



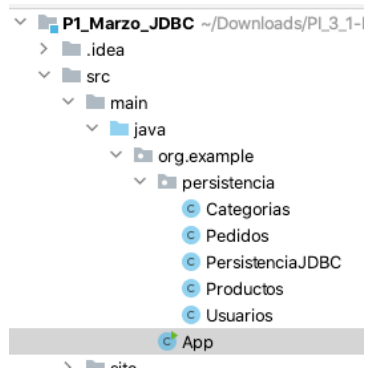
PASOS PREVIOS

NOTA: los siguientes pasos los realizarás junto con el profesor. No te adelantes a sus explicaciones. Una vez terminadas sus indicaciones, se dará paso al examen.

1. Lanza el script proporcionado que te creará la BBDD AD_tienda.
2. Deberás importar los dos proyectos que se te proporcionan, uno para JDBC (Ejercicio 1.1→P1_Marzo_JDBC), y el otro para JPA-Hibernate (Ejercicio 1.2→P1_Marzo_JPA). Al final del examen tendrás que comprimir los proyectos a en un fichero llamado **P1_Marzo_NombreApellidos.zip**.
3. Configura las **dependencias** adecuadas al gestor de base de datos en cada uno de los proyectos.
4. Configura el fichero de configuración de persistencia de acuerdo con las características del gestor de base de datos en el proyecto JPA.

EJERCICIO 1.1 – JDBC (3,25 PUNTOS)

Dentro del proyecto JDBC tendrás la siguiente estructura de paquetes:



- a. org.example.persistencia: paquete con las clases de la capa de persistencia.
- b. App.java: fichero de prueba con el método main correspondiente.

1. **(0.5 pto.)** Crea los POJO de forma conveniente.
2. **(2.75 ptos.)** Implementa el siguiente método dentro de la clase PersistenciaJDBC atendiendo a la siguiente descripción:

(2.75 ptos.) aniadirProducto

```
public boolean aniadirProducto (Productos producto) {
```

Añadirá el objeto producto pasado por parámetro, siempre y cuando, el id de la categoría asociada sea válido. En caso de serlo, y de realizar correctamente la inserción, devolverá true. En caso contrario false

Parametros:

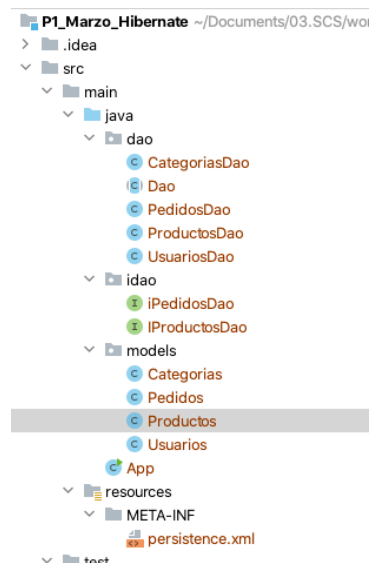
producto– objeto de tipo Productos.

Returns:

Devuelve true en caso de que se haya podido añadir el objeto Productos. False en caso contrario.

EJERCICIO 1.2 – JPA-HIBERNATE (6.75 PUNTOS)

Dentro del proyecto JPA tendrás la siguiente estructura de paquetes:



- a. models: paquete con las clases correspondientes a las entidades del dominio
 - b. iDao: paquete con las interfaces DAO
 - c. dao: paquete con las clases de la capa de persistencia que deberán usar una clase abstracta *DAO* que proporcionará el acceso a datos mediante el objeto *EntityManager* y además tendrá implementados los métodos básicos del CRUD.
 - d. App.java: donde deberás probar TODOS los métodos dao implementados convenientemente.
1. (0.25 ptos.) Crea una clase Dao abstracta dentro del paquete dao. Utilízala en el resto de clases de persistencia convenientemente.
 2. (2 ptos.) Crea TODAS las anotaciones de persistencia JPA necesarias para las entidades Categorías, Pedidos, Productos y Usuarios. Debes representar tanto el mapeo de campos con las tablas de BBDD así como las relaciones entre las entidades.
 3. (0.25 ptos.) Modifica el fichero de configuración persistence.xml para configurar la conexión convenientemente. También deberá mostrar por consola cada sentencia SQL ejecutada.
 4. (4.25 ptos.) Implementa los siguientes métodos conforme a la especificación siguiente:

(1.5 ptos.) aniadirProducto

```
public boolean aniadirProducto(Productos producto)
```

El objeto de este método es añadir el Objeto pasado por parámetro a la BBDD. Para ello antes deberás asegurarte que el identificador de la categoría pasado como parámetro existe.

Parámetros:

producto – Objeto Productos que se desea añadir a la base de datos.

Returns:

Devolverá true en caso de que la inserción haya sido exitosa. False en caso contrario.

(1.25 pto.) listarPedidosUsuario

```
public List<Pedidos> listarPedidosUsuario(Integer usuarioId)
```

Este método listará los pedidos que tiene asociados el usuario pasado por parámetro. Para ello antes deberás asegurarte que el identificador de usuario pasado como parámetro existe.

Parámetros

usuarioId– identificador del usuario para el cual se quiere listar los pedidos.

Returns:

Devuelve el listado de Pedidos asociados al usuario.

(1.5 pto.) listarProductosPorCategoria

```
public List<Productos> listarProductosCategoria(Integer categoriaId)
```

Este método listará los productos para una determinada categoría pasado por parámetro. Para ello antes deberás asegurarte que el identificador de la categoría pasado como parámetro existe.

Parámetros

categoryId– identificador de la categoría para el cual se quiere listar los productos.

Returns:

Devuelve el listado de Productos asociados a dicha categoría.