## **Ejercicio agendaV2 (hibernate,JPQL,Maven)**

### Proyecto j16\_agendaV2

#### Notas previas:

**JPQL** es el lenguaje de consulta estándar definido por JPA. Te permite escribir consultas de manera portable y orientada a objetos.

**Hibernate** es una implementación o una herramienta concreta de código abierto para Java (un framework ORM - Object-Relational Mapping) que sigue y cumple con las normas definidas por la especificación JPA.

**Maven** es una herramienta fundamental en el desarrollo de Java, actuando como un gestor de proyectos y una herramienta de automatización de construcción (build automation tool). Referencia en archivos **pom.xml** que indica versión de hibernate y que usamos MySQL, así como nombre del archivo y su id además de la versión de Maven.

#### Estructura del proyecto:

Las carpetas principales serán(con dominio com.curso getafe.agenda (esto asegura que no hay ningún proyecto igual en el mundo):

##### com.cursogetafe.agenda.config:

contiene el archivo Config, crea usando singleton(asegura que sea único y no haya copias), un EntityManagerFactory (Equivale a la gestión de conexiones, es decir un DataSource o Pool).

Tiene 2 métodos :

* getDataSource() crea el data source con las propiedades que necesita para conectarse(URL, Driver,usuario, password)
* getProperties() ayuda a obtener los valores para el data source leyendo del archivo properties (app.properties usando FileReader de Java)

##### **com.cursogetafe.agenda.inicio:**

Inicia la aplicación de agenda, bien por consola o bien por Vista usando las clases de la carpeta **8. com.cursogetafe.agenda.vista** (lo que hacen las opciones del programa)

##### com.cursogetafe.agenda.modelo:

Contiene las clases que gestionan las tablas de la BBDD, deberán estar mapeadas para ello con las diferentes etiquetas(@diferentes\_etiquetas)

##### com.cursogetafe.agenda.negocio:

Contiene la interface que obliga a implementar los métodos que usaremos en nuestra aplicación agenda (Agenda.java) y un archivo con los métodos de interface implementando esa interface (AgendaImpl.java)

##### com.cursogetafe.agenda.persistencia:

Utiliza el patrón **DAO**(Data Access Object), separa la lógica de negocio de la lógica de persistencia de datos.

Usa interfaces para las operaciones con las BBDD (CRUD: Create, Read, Actualice, Delete/Drop) de una entidad específica (ContactoDAO.java será la interface y las clases que lo implementan en su nombre especifican lo que cumplen en los diferentes casos:

###### **ContactoDaoJDBC.java:**

###### Implementa de la interface de la clase que representa a la entidad, los métodos que actúan en la base de datos Usando JDBC :

###### Driver que indica el software SQL(mySQL, postgres,etc)

###### Conexión,usa el Driver,y gestiona conexiones DataSource.

###### Usando una consulta SQL sobre la BBDD y utilizando modo transaccional(commit y rollback) si hace modificaciones

###### **ContactoDaoJPA.java:**

###### Implementa de la interface de la clase que representa a la entidad, los métodos que actúan en la base de datos Usando JPA :

Con el mapeo establecido en las clases, se realiza mediante un EntityManager que usa ese mapeo, una consulta JPQL que referencia las clases independizando la consulta del software gestor de SQL (mySQL, postgres,Oracle,etc) respetando la norma de modo transaccional para modificaciones en la BBDD(commit y rollback).

###### 

###### **ContactoDaoMem.java**

###### **ContactoDaoMemSerial.java:**

##### com.cursogetafe.agenda.tests:

Contiene las diferentes pruebas que se han hecho con las clases y sus métodos

##### com.cursogetafe.agenda.util:

Contiene clases que:

* permiten rellenar la BBDD (Contactos.java con el método generaContactos())
* Conjunto de **métodos estáticos y reutilizables** para realizar tareas(leer diferentes tipos de entrada del usuario desde la consola y operaciones relacionadas con fechas y números aleatorios)(Util.java)

##### com.cursogetafe.agenda.vista:

Contiene los métodos que se ejecutarán desde la capa de vista (por consola)

##### com.cursogetafe.agenda.vista.swing:

Contiene los métodos que se ejecutarán desde la capa de vista (mediante aplicación desarrollada en Swing)

##### /j16\_agendaV2/src/main/java/META-INF:

Contiene el archivo **persistence.XML** que apunta a las clases que se están usando con hibernate, además de contener valores en uso de maven y MySQL(entre las etiquetas <properties>)