

## Ejercicio 1. Primer ejemplo con framework de persistencia – JPA – Hibernate.

Dado el siguiente script de Oracle, definido en el fichero `schema.ddl` contenido en el subdirectorio `sql` de un proyecto JPA para Eclipse:

```
DROP TABLE ALUMNO CASCADE CONSTRAINT;
```

```
CREATE TABLE ALUMNO (  
    NIF VARCHAR(9),  
    EDAD SMALLINT NOT null,  
    NOMBRE VARCHAR(30),  
    APELLIDO VARCHAR(30),  
    PRIMARY KEY (NIF)  
);
```

### Realizar las siguientes tareas.

1. Crear un proyecto JPA en Eclipse, con *Hibernate* como proveedor de persistencia y utilizando una conexión con nombre *Oracle XE* (una vez creado el proyecto se puede reemplazar el fichero `persistence.xml` con el disponible en UBUVirtual).
2. Ejecutar el script SQL **desde Eclipse** (no usar SQLDeveloper, SQL\*Plus, APEX o similares) para la creación de la tabla. *Nota: se proporciona el fichero `schema.ddl` en el subdirectorio `sql` dentro del directorio raíz de nuestro proyecto Eclipse.*
3. Escribir el código de la entidad JPA correspondiente a la tabla `ALUMNO` que se acaba de crear. Realizar los ajustes necesarios en el proyecto para su correcta compilación. Se puede usar la herramienta de ingeniería inversa vista en el guión de prácticas para generar la entidad a partir de la tabla o bien editar el código manualmente (atributos, *getters* y *setters*).
4. Añadir la clase `Main` (disponible en UBUVirtual) que utilizando la entidad previamente construida, permite realizar los siguientes pasos:
  1. Crear un alumno
  2. Establecer sus datos e insertar en la base de datos
  3. Guardar el alumno en la base de datos
  4. Listar toda la lista de alumnos en la base de datos
  5. Actualizar los datos del alumno
  6. Listar toda la lista de alumnos en la base de datos
  7. Eliminar el alumno
  8. Listar toda la lista de alumnos en la base de datos

**Aclaración:** este apartado 4 es un mero ejemplo de uso de JPA, sin integrar ningún patrón de diseño y resolviendo el ejercicio de manera similar a como se ha realizado con JDBC. Se delega todo el trabajo fundamentalmente sobre el framework de persistencia. Se utilizan algunos conceptos, que se irán viendo con mayor detalle en sucesivas semanas.

5. Ejecutar el método `main` de la clase anterior y comprobar que funciona correctamente sin errores en pantalla.

6. Incluir un fichero de configuración para *log* (fichero `log4j2.properties`) y cambiar el nivel de *log*, comprobando las diferentes salidas en la consola de Eclipse (Ej: cambiar de ALL a OFF y viceversa). Recordad que un fichero `.properties` es un fichero de texto con pares clave y valor que puede editarse. En desarrollo ese fichero se ubica en el directorio `src`.

7. Activar las opciones de traza de *Hibernate*, en el fichero `persistence.xml` (cambiar opciones `hibernate.show_sql` y `hibernate.format_sql` a **true**) y comprobar los efectos en consola, al ejecutar de nuevo. Copiar la traza de la ejecución (con OFF en el `log4j2.properties`) e intenta deducir la correspondencia de las sentencias SQL generadas por Hibernate con el código de la clase `Main`. Por ejemplo: "*La SQL <texto de la sentencia SQL> se corresponde con la línea de código Java <texto de la línea de Java>*".

### ***Tarea opcional avanzada***

8. Cambiar de proveedor de persistencia a *EclipseLink*, modificando el fichero `persistence.xml`, comentando las líneas de *Hibernate* y descomentando las correspondientes a *EclipseLink*. Recordad que es necesario además incluir la biblioteca de usuario *EclipseLink*. Ejecutar una vez realizados los cambios, comprobando el correcto funcionamiento de la clase `Main` y compara resultados con la ejecución con *Hibernate*. El resultado final es el mismo en la base de datos pero cada proveedor de persistencia puede trabajar de forma diferente.

## **Bibliotecas**

Se utilizarán las bibliotecas de usuario del proyecto `user_lib_JPA`, disponible en UBUVirtual y explicadas en el guión de configuración:

- **Oracle**
- **Hibernate**
- **SLF4J** (si utilizamos logging)
- **EclipseLink** (solo si se intenta resolver el ejercicio 8)

## **Entregable**

Adjuntar un documento `ejercicio_01_JPA.pdf` con el resultado de la **tarea 7**.