



# Acceso a datos

ActUT04. Herramientas de  
mapeo Objeto-Relacional

Francisco José García Cutillas | 2FPGS\_DAM



## Índice

Ejercicio 1 .....	3
-------------------	---

## Ejercicio 1

Adjunto a la tarea tienes un script fct.sql que te va a generar la base de datos.

Utilizando Hibernate, genera un menú que realice las siguientes sobre la base de datos.

Para mostrar los resultados de cada apartado, vamos a desactivar la muestra de las consultas realizadas a la base de datos desde el fichero de configuración de Hibernate.

```
<property name="hibernate.show_sql">false</property>
```

### 1. Consultar todos alumnos

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

1
Alumnos presentes en la base de datos:
Alumno con id 1, Nombre= Marina Garcia, Edad= 21, Titulo= DAM, Empresa de prácticas= NTT DATA
Alumno con id 2, Nombre= Alejandro Ruiz, Edad= 43, Titulo= DAW, Empresa de prácticas= NEORIS
Alumno con id 3, Nombre= Laura Martínez, Edad= 32, Titulo= DAW, Prácticas sin asignar
Alumno con id 4, Nombre= Javier López, Edad= 24, Titulo= DAM, Prácticas sin asignar
Alumno con id 5, Nombre= Sara Rodríguez, Edad= 21, Titulo= ASIR, Prácticas sin asignar
```

### 2. Consultar todas empresas

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

2
Empresas presentes en la base de datos:
Empresa con id= 1, Nombre= NTT DATA, Sector= Tecnología, Prácticas asignadas= 1
Empresa con id= 2, Nombre= NEORIS, Sector= Data Science, Prácticas asignadas= 1
Empresa con id= 3, Nombre= METAENLACE, Sector= Seguridad Informática, Prácticas asignadas= 1
Empresa con id= 4, Nombre= SLANGO, Sector= Desarrollo Web, Prácticas asignadas= 1
Empresa con id= 5, Nombre= AYESA, Sector= Cloud Computing, Prácticas asignadas= 1
```

### 3. Consultar todas prácticas

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

3
Prácticas presentes en la base de datos:
Practica con id= 1, Fecha de inicio= 2022-02-01, Fecha de fin= 2022-04-30, Descripción= Desarrollo Web en Tech Innovators, ID de la empresa= 1, ID del alumno= 1
Practica con id= 2, Fecha de inicio= 2022-03-15, Fecha de fin= 2022-06-15, Descripción= Análisis de Datos en Data Wizards, ID de la empresa= 2, ID del alumno= 2
Practica con id= 3, Fecha de inicio= 2022-04-10, Fecha de fin= 2022-06-30, Descripción= Seguridad de Redes en SecureNet, ID de la empresa= 3, Alumno sin asignar
Practica con id= 4, Fecha de inicio= 2022-05-01, Fecha de fin= 2022-07-31, Descripción= Desarrollo de Software en Web Solutions, ID de la empresa= 4, Alumno sin asignar
Practica con id= 5, Fecha de inicio= 2022-06-15, Fecha de fin= 2022-09-15, Descripción= Cloud Computing en Cloud Systems, ID de la empresa= 5, Alumno sin asignar
```

### 4. Consultar los alumnos candidatos de una práctica

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

4
Introduce el ID de la práctica
1
Alumnos candidatos a la práctica con ID: 1
Alumno con id 1, Nombre= Marina García, Edad= 21, Titulo= DAM, Empresa de prácticas= NTT DATA
Alumno con id 2, Nombre= Alejandro Ruiz, Edad= 43, Titulo= DAW, Empresa de prácticas= NEORIS
```

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

4
Introduce el ID de la práctica
8
No existe una práctica con id: 8, No entity found for query
```

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

4
Introduce el ID de la práctica
5
La práctica no tiene alumnos candidatos
```

## 5. Alta alumno

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.
```

5

Introduce el nombre del alumno:

Antonio

Introduce la edad del alumno:

25

Introduce el título del alumno:

ASIR

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.
```

1

Alumnos presentes en la base de datos:

Alumno con id 1, Nombre= Marina García, Edad= 21, Título= DAM, Empresa de prácticas= NTT DATA

Alumno con id 2, Nombre= Alejandro Ruiz, Edad= 43, Título= DAW, Empresa de prácticas= NEORIS

Alumno con id 3, Nombre= Laura Martínez, Edad= 32, Título= DAW, Prácticas sin asignar

Alumno con id 4, Nombre= Javier López, Edad= 24, Título= DAM, Prácticas sin asignar

Alumno con id 5, Nombre= Sara Rodríguez, Edad= 21, Título= ASIR, Prácticas sin asignar

Alumno con id 6, Nombre= Antonio, Edad= 25, Título= ASIR, Prácticas sin asignar

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.
```

5

Introduce el nombre del alumno:

Pepe

Introduce la edad del alumno:

15

Introduce el título del alumno:

DAW

No ha sido posible dar de alta el alumno. La edad introducida debe ser mayor o igual a 16 años.

## 6. Alta empresa

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

6
Introduce el nombre de la empresa:
Libelium
Introduce el sector de la empresa:
Sector ambiental

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

2
Empresas presentes en la base de datos:
Empresa con id= 1, Nombre= NTT DATA, Sector= Tecnología, Prácticas asignadas= 1
Empresa con id= 2, Nombre= NEORIS, Sector= Data Science, Prácticas asignadas= 1
Empresa con id= 3, Nombre= METAENLACE, Sector= Seguridad Informática, Prácticas asignadas= 1
Empresa con id= 4, Nombre= SLANGO, Sector= Desarrollo Web, Prácticas asignadas= 1
Empresa con id= 5, Nombre= AYESA, Sector= Cloud Computing, Prácticas asignadas= 1
Empresa con id= 6, Nombre= Libelium, Sector= Sector ambiental, sin prácticas asignadas
```

## 7. Alta práctica

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

7
Introduce la fecha de inicio de la práctica en formato (dd/MM/yyyy):
05/03/2024
Introduce la fecha de fin de la práctica en formato (dd/MM/yyyy):
10/06/2024
Introduce la descripción de la práctica:
Programación de aplicaciones móviles
Introduce el ID de la empresa donde se realizará la práctica:
3
¿Tiene la práctica algún alumno asignado?(S/N)
s
Introduce el ID del alumno:
3
```

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

3
Prácticas presentes en la base de datos:
Practica con id= 1, Fecha de inicio= 2022-02-01, Fecha de fin= 2022-04-30, Descripción= Desarrollo Web en Tech Innovators, ID de la empresa= 1, ID del alumno= 1
Practica con id= 2, Fecha de inicio= 2022-03-15, Fecha de fin= 2022-06-15, Descripción= Análisis de Datos en Data Wizards, ID de la empresa= 2, ID del alumno= 2
Practica con id= 3, Fecha de inicio= 2022-04-10, Fecha de fin= 2022-06-30, Descripción= Seguridad de Redes en SecureNet, ID de la empresa= 3, Alumno sin asignar
Practica con id= 4, Fecha de inicio= 2022-05-01, Fecha de fin= 2022-07-31, Descripción= Desarrollo de Software en Web Solutions, ID de la empresa= 4, Alumno sin asignar
Practica con id= 5, Fecha de inicio= 2022-06-15, Fecha de fin= 2022-09-15, Descripción= Cloud Computing en Cloud Systems, ID de la empresa= 5, Alumno sin asignar
Practica con id= 6, Fecha de inicio= 2024-03-05, Fecha de fin= 2024-06-10, Descripción= Programaci?n de aplicaciones m?viles, ID de la empresa= 3, ID del alumno= 3
```

```
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

7
Introduce la fecha de inicio de la práctica en formato (dd/MM/yyyy):
03/03/2024
Introduce la fecha de fin de la práctica en formato (dd/MM/yyyy):
25/05/2024
Introduce la descripción de la práctica:
Reparar impresoras
Introduce el ID de la empresa donde se realizará la práctica:
7
¿Tiene la práctica algún alumno asignado?(S/N)
n

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

3
Prácticas presentes en la base de datos:
Practica con id= 1, Fecha de inicio= 2022-02-01, Fecha de fin= 2022-04-30, Descripción= Desarrollo Web en Tech Innovators, ID de la empresa= 1, ID del alumno= 1
Practica con id= 2, Fecha de inicio= 2022-03-15, Fecha de fin= 2022-06-15, Descripción= Análisis de Datos en Data Wizards, ID de la empresa= 2, ID del alumno= 2
Practica con id= 3, Fecha de inicio= 2022-04-10, Fecha de fin= 2022-06-30, Descripción= Seguridad de Redes en SecureNet, ID de la empresa= 3, Alumno sin asignar
Practica con id= 4, Fecha de inicio= 2022-05-01, Fecha de fin= 2022-07-31, Descripción= Desarrollo de Software en Web Solutions, ID de la empresa= 4, Alumno sin asignar
Practica con id= 5, Fecha de inicio= 2022-06-15, Fecha de fin= 2022-09-15, Descripción= Cloud Computing en Cloud Systems, ID de la empresa= 5, Alumno sin asignar
Practica con id= 6, Fecha de inicio= 2024-03-05, Fecha de fin= 2024-06-10, Descripción= Programaci?n de aplicaciones m?viles, ID de la empresa= 3, ID del alumno= 3
Practica con id= 7, Fecha de inicio= 2024-03-03, Fecha de fin= 2024-05-25, Descripción= Reparar impresoras, ID de la empresa= 7, Alumno sin asignar
```

```

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

7
Introduce la fecha de inicio de la práctica en formato (dd/MM/yyyy):
10/03/2024
Introduce la fecha de fin de la práctica en formato (dd/MM/yyyy):
02/06/2024
Introduce la descripción de la práctica:
Montaje de equipos informáticos
Introduce el ID de la empresa donde se realizará la práctica:
10
No existe una empresa con id: 10, No entity found for query

```

## 8. Asignar un alumno a una práctica como candidato.

```

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

8
Introduce el ID de la práctica:
2
Introduce el ID del alumno a asignar:
3
**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

4
Introduce el ID de la práctica
2
Alumnos candidatos a la práctica con ID: 2
Alumno con id 2, Nombre= Alejandro Ruiz, Edad= 43, Titulo= DAW, Empresa de prácticas= NEORIS
Alumno con id 3, Nombre= Laura Martínez, Edad= 32, Titulo= DAW, Empresa de prácticas= METAENLACE

```



```

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

8
Introduce el ID de la práctica:
1
Introduce el ID del alumno a asignar:
1
feb 03, 2024 9:45:39 P. M. org.hibernate.engine.jdbc.spi.SqlExceptionHelper logExceptions
WARN: SQL Error: 1062, SQLState: 23000
feb 03, 2024 9:45:39 P. M. org.hibernate.engine.jdbc.spi.SqlExceptionHelper logExceptions
ERROR: Duplicate entry '1-1' for key 'alumnosandpracticass.PRIMARY'
feb 03, 2024 9:45:39 P. M. org.hibernate.engine.jdbc.batch.internal.AbstractBatchImpl release
INFO: HHH000010: On release of batch it still contained JDBC statements
El alumno con ID 1 ya está como candidato de la práctica con ID 1

```

## 9. Asignar un alumno a una práctica (alumno que realiza la práctica).

```

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

9
Introduce el ID de la práctica a asignar:
4

Introduce el ID del alumno que quieres asignar a la práctica:
4

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

3
Prácticas presentes en la base de datos:
Practica con id= 1, Fecha de inicio= 2022-02-01, Fecha de fin= 2022-04-30, Descripción= Desarrollo Web en Tech Innovators, ID de la empresa= 1, ID del alumno= 1
Practica con id= 2, Fecha de inicio= 2022-03-15, Fecha de fin= 2022-06-15, Descripción= Análisis de Datos en Data Wizards, ID de la empresa= 2, ID del alumno= 2
Practica con id= 3, Fecha de inicio= 2022-04-10, Fecha de fin= 2022-06-30, Descripción= Seguridad de Redes en SecureNet, ID de la empresa= 3, ID del alumno= 3
Practica con id= 4, Fecha de inicio= 2022-05-01, Fecha de fin= 2022-07-31, Descripción= Desarrollo de Software en Web Solutions, ID de la empresa= 4, ID del alumno= 4
Practica con id= 5, Fecha de inicio= 2022-06-15, Fecha de fin= 2022-09-15, Descripción= Cloud Computing en Cloud Systems, ID de la empresa= 5, Alumno sin asignar

```

```

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.
9
Introduce el ID de la práctica a asignar:
4
La práctica con ID 4 ya tiene alumno asignado

```

```

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.
9
Introduce el ID de la práctica a asignar:
6
No existe una práctica con id: 6, No entity found for query

```

```

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.
9
Introduce el ID de la práctica a asignar:
5
Introduce el ID del alumno que quieres asignar a la práctica:
8
No existe un alumno con id: 8, No entity found for query

```

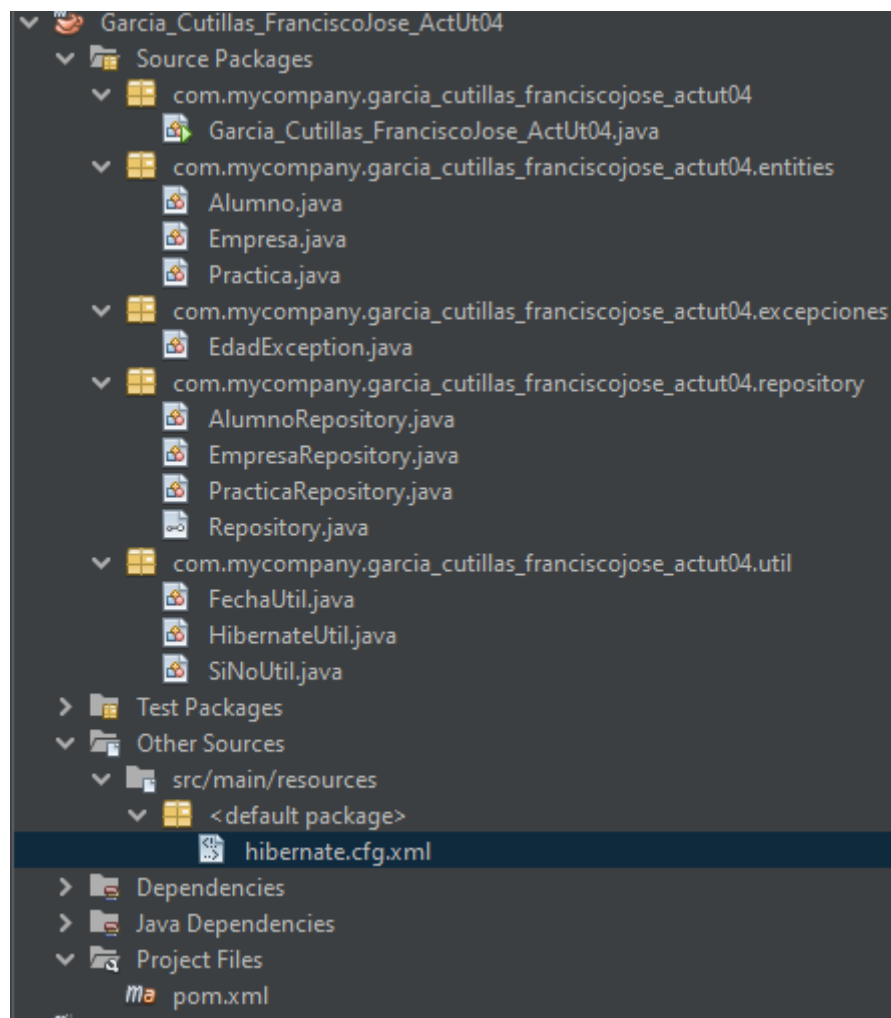
```

**** Aplicación para gestionar la FCT de los alumnos ****
Opciones:
- Pulsa 1 para consultar todos los alumnos.
- Pulsa 2 para consultar todas las empresas.
- Pulsa 3 para consultar todas las prácticas.
- Pulsa 4 para consultar los alumnos candidatos de una práctica.
- Pulsa 5 para dar de alta un alumno.
- Pulsa 6 para dar de alta una empresa.
- Pulsa 7 para dar de alta una práctica.
- Pulsa 8 para asignar un alumno a una práctica como candidato.
- Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).
- Pulsa cualquier tecla para salir.

0
¡Gracias por utilizar nuestra herramienta!
feb 03, 2024 10:00:34 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost/fct]
-----
BUILD SUCCESS

```

## Estructura del proyecto



## Clase Main

```

1 package com.myccompany.garcia_cutillas_franciscojose_actut04;
2
3 import com.myccompany.garcia_cutillas_franciscojose_actut04.entities.Alumno;
4 import com.myccompany.garcia_cutillas_franciscojose_actut04.entities.Empresa;
5 import com.myccompany.garcia_cutillas_franciscojose_actut04.entities.Practica;
6 import com.myccompany.garcia_cutillas_franciscojose_actut04.excepciones.EdadException;
7 import com.myccompany.garcia_cutillas_franciscojose_actut04.repository.AlumnoRepository;
8 import com.myccompany.garcia_cutillas_franciscojose_actut04.repository.EmpresaRepository;
9 import com.myccompany.garcia_cutillas_franciscojose_actut04.repository.PracticaRepository;
10 import com.myccompany.garcia_cutillas_franciscojose_actut04.util.FechaUtil;
11 import com.myccompany.garcia_cutillas_franciscojose_actut04.util.HibernateUtil;
12 import com.myccompany.garcia_cutillas_franciscojose_actut04.util.SiNoUtil;
13 import java.util.ArrayList;
14 import java.util.InputMismatchException;
15 import java.util.List;
16 import java.util.Scanner;
17 import javax.persistence.PersistenceException;
18
19 /**
20  *
21  * @author Fran
22  */
23 public class Garcia_Cutillas_FranciscoJose_ActUt04 {
24
25     private static Scanner sc = new Scanner(System.in, charsetName:"ISO-8859-1");
26     private static final PracticaRepository practicaRepository = new PracticaRepository();
27     private static final EmpresaRepository empresaRepository = new EmpresaRepository();
28     private static final AlumnoRepository alumnoRepository = new AlumnoRepository();
29
30     public static void main(String[] args) {
31
32         boolean salir = false;
33         do {
34
35             salir = menu();
36
37         } while (salir == false);
38     }
39
40     private static boolean menu() {
41
42         HibernateUtil.buildSessionFactory();
43
44         int menu = 0;
45
46         System.out.println("***** Aplicación para gestionar la FCT de los alumnos *****");
47         System.out.println("\tOpciones:");
48         System.out.println("\t - Pulsa 1 para consultar todos los alumnos.");
49         System.out.println("\t - Pulsa 2 para consultar todas las empresas.");
50         System.out.println("\t - Pulsa 3 para consultar todas las prácticas.");
51         System.out.println("\t - Pulsa 4 para consultar los alumnos candidatos de una práctica.");
52         System.out.println("\t - Pulsa 5 para dar de alta un alumno.");
53         System.out.println("\t - Pulsa 6 para dar de alta una empresa.");
54         System.out.println("\t - Pulsa 7 para dar de alta una práctica.");
55         System.out.println("\t - Pulsa 8 para asignar un alumno a una práctica como candidato.");
56         System.out.println("\t - Pulsa 9 para asignar un alumno a una práctica(alumno que realiza la práctica).");
57         System.out.println("\t - Pulsa cualquier tecla para salir.");
58     }
59

```

```

60     try {
61         menu = sc.nextInt();
62     } catch (InputMismatchException ex) {
63         menu = 0;
64     }
65     sc.nextLine();
66     boolean salir = false;
67
68     switch (menu) {
69         case 1:
70             mostrarAlumnos();
71             break;
72         case 2:
73             mostrarEmpresas();
74             break;
75         case 3:
76             mostrarPracticas();
77             break;
78         case 4:
79             mostrarAlumnosCandidatos();
80             break;
81         case 5:
82             altaAlumno();
83             break;
84         case 6:
85             altaEmpresa();
86             break;
87         case 7:
88             altaPractica();
89             break;
90         case 8:
91             asignarAlumnoCandidato();
92             break;
93         case 9:
94             asignarAlumnoPractica();
95             break;
96         default:
97             System.out.println(x: "Gracias por utilizar nuestra herramienta!");
98             salir = true;
99     }
100
101     HibernateUtil.closeSessionFactory();
102
103     return salir;
104 }
105
106 //1. Consultar todos los alumnos
107 private static void mostrarAlumnos() {
108     ArrayList<Alumno> listaAlumnos = (ArrayList<Alumno>) alumnoRepository.findAll();
109
110     System.out.println(x: "Alumnos presentes en la base de datos:");
111
112     for (Alumno a : listaAlumnos) {
113         System.out.println(" " + a);
114     }
115 }
116
117 }

```

```

126 //2. Consultar todas las empresas
127 private static void mostrarEmpresas() {
128
129     ArrayList<Empresa> listaEmpresas = (ArrayList<Empresa>) empresaRepository.findAll();
130
131     System.out.println(:"Empresas presentes en la base de datos:");
132
133     for (Empresa e : listaEmpresas) {
134         System.out.println(" " + e);
135     }
136
137 }
138
139 //3. Consultar todas las prácticas
140 private static void mostrarPracticas() {
141
142     ArrayList<Practica> listaPracticas = (ArrayList<Practica>) practicaRepository.findAll();
143
144     System.out.println(:"Prácticas presentes en la base de datos:");
145
146     for (Practica p : listaPracticas) {
147         System.out.println(" " + p);
148     }
149
150 }
151
152 //4. Consultar alumnos candidatos de una práctica
153 private static void mostrarAlumnosCandidatos() {
154
155     Practica practica = new Practica();
156
157     System.out.println(:"Introduce el ID de la práctica");
158
159     int idPractica = sc.nextInt();
160     sc.nextLine();
161
162     practica = practicaRepository.findOneById(idPractica);
163     List<Alumno> listaAlumnos = null;
164
165     if (practica != null) {
166
167         listaAlumnos = practica.getListaAlumnos();
168
169         if (listaAlumnos.size() > 0) {
170
171             System.out.println("Alumnos candidatos a la práctica con ID: " + idPractica);
172
173             for (Alumno a : listaAlumnos) {
174                 System.out.println(" " + a);
175             }
176
177         } else {
178
179             System.out.println(:"La práctica no tiene alumnos candidatos");
180
181         }
182
183     }
184
185 }

```

```

187 //5. Alta alumno
188 private static void altaAlumno() {
189
190     System.out.println("Introduce el nombre del alumno:");
191     String nombre = sc.nextLine();
192
193     System.out.println("Introduce la edad del alumno:");
194     int edad = sc.nextInt();
195     sc.nextLine();
196
197     System.out.println("Introduce el titulo del alumno:");
198     String titulo = sc.nextLine();
199
200     try {
201
202         Alumno alumno = new Alumno(nombre, edad, titulo);
203         HibernateUtil.getCurrentSession().beginTransaction();
204         HibernateUtil.getCurrentSession().save(sc, alumno);
205         HibernateUtil.getCurrentSession().getTransaction().commit();
206
207     } catch (EdadException ex) {
208
209         System.out.println("No ha sido posible dar de alta el alumno. " + ex.getMessage());
210
211     }
212
213 }
214
215 //6. Alta empresa
216 private static void altaEmpresa() {
217
218     System.out.println("Introduce el nombre de la empresa:");
219     String nombre = sc.nextLine();
220
221     System.out.println("Introduce el sector de la empresa:");
222     String sector = sc.nextLine();
223
224     Empresa empresa = new Empresa(nombre, sector);
225
226     ArrayList<Empresa> empresas = (ArrayList<Empresa>) empresaRepository.findAll();
227
228     boolean empresaBase = false;
229     for (Empresa e : empresas) {
230
231         if (e.getNombre().toUpperCase().contains(sc, empresa.getNombre().toUpperCase())) {
232
233             empresaBase = true;
234
235         }
236
237     }
238
239     /*Comprobación de que la empresa no se encuentra ya en la base de datos. Como no vamos a controlar
240     desde aquí los id de las empresas, vamos a controlar que no haya dos empresas con el mismo nombre*/
241     if (!empresaBase) {
242
243         HibernateUtil.getCurrentSession().beginTransaction();
244         HibernateUtil.getCurrentSession().save(sc, empresa);
245         HibernateUtil.getCurrentSession().getTransaction().commit();
246
247     } else {
248
249         System.out.println("La empresa " + empresa.getNombre() + " ya se encuentra en la base de datos, "
250             + "por lo que no es posible añadirla de nuevo");
251
252     }
253
254 }

```

```

253 }
254
255 //7. Alta práctica
256 private static void altaPractica() {
257     System.out.println("Introduce la fecha de inicio de la práctica en formato (dd/MM/yyyy):");
258     String fechaInicio = sc.nextLine();
259
260     System.out.println("Introduce la fecha de fin de la práctica en formato (dd/MM/yyyy):");
261     String fechaFin = sc.nextLine();
262
263     System.out.println("Introduce la descripción de la práctica:");
264     String descripcion = sc.nextLine();
265
266     System.out.println("Introduce el ID de la empresa donde se realizará la práctica:");
267     int idEmpresa = sc.nextInt();
268     sc.nextLine();
269
270     //Extraer la empresa, para pasarla como parámetro a la práctica.
271     Empresa empresa = empresaRepository.findOneById(id: idEmpresa);
272
273     //Comprobamos que la empresa está en la base de datos
274     if (empresa != null) {
275         boolean correcto = false;
276         String seleccion;
277         System.out.println("¿Tiene la práctica algún alumno asignado?(S/N)");
278         seleccion = sc.nextLine();
279
280         do {
281             //Comprobamos que se introduce bien el valor S/N
282             if (SiNoUtil.comprobarSiNo(entrada: seleccion)) {
283                 correcto = true;
284             } else {
285                 System.out.println("Tienes que elegir entre (S/N)");
286             }
287         } while (!correcto);
288
289         Practica practica;
290         Alumno alumno;
291
292         //Si tenemos alumno para asignar a la práctica, la creamos con alumno
293         if (seleccion.equalsIgnoreCase(anotherString: "S")) {
294             System.out.println("Introduce el ID del alumno:");
295             int idAlumno = sc.nextInt();
296             sc.nextLine();
297
298             //Comprobar el id del alumno y meter alumno
299             alumno = alumnoRepository.findOneById(id: idAlumno);
300
301         }
302     }
303 }

```



```

312 //Si el alumno no tiene práctica asignada, creamos práctica
313 if (!comprobarAlumnoPractica(a: alumno)) {
314
315     practica = new Practica( fechaInicio: FechaUtil.convierteFecha( fechaSpain: fechaInicio),
316                             fechaFin: FechaUtil.convierteFecha( fechaSpain: fechaFin), descripcion, empresa, alumno);
317
318     HibernateUtil.getCurrentSession().beginTransaction();
319     HibernateUtil.getCurrentSession().save(o: practica);
320     HibernateUtil.getCurrentSession().getTransaction().commit();
321
322 } else {
323
324     System.out.println("El alumno con id " + alumno.getId() + " ya tiene práctica asignada");
325
326 }
327
328 //Si no tenemos alumno, la creamos sin él
329 } else {
330
331     practica = new Practica( fechaInicio: FechaUtil.convierteFecha( fechaSpain: fechaInicio),
332                             fechaFin: FechaUtil.convierteFecha( fechaSpain: fechaFin), descripcion, empresa);
333
334     HibernateUtil.getCurrentSession().beginTransaction();
335     HibernateUtil.getCurrentSession().save(o: practica);
336     HibernateUtil.getCurrentSession().getTransaction().commit();
337
338 }
339
340 }
341
342 }
343
344 //8. Asignar un alumno a una práctica como candidato
345 private static void asignarAlumnoCandidato() {
346
347     //Compruebo que la práctica existe.
348     System.out.println(x: "Introduce el ID de la práctica:");
349     int idPractica = sc.nextInt();
350     sc.nextLine();
351     Practica practica = practicaRepository.findOneById(id: idPractica);
352
353     if (practica != null) {
354
355         System.out.println(x: "Introduce el ID del alumno a asignar:");
356         int idAlumno = sc.nextInt();
357         sc.nextLine();
358         Alumno alumno = alumnoRepository.findOneById(id: idAlumno);
359
360         //Compruebo que existe el alumno
361         if (alumno != null) {
362
363             //Le asigno como candidato a la práctica
364             try {
365
366                 alumnoRepository.saveCandidato(id_alumno: idAlumno, id_practica: idPractica);
367
368             } catch (PersistenceException ex) {
369
370                 System.out.println("El alumno con ID " + alumno.getId() + " ya está como candidato de la práctica con ID "
371                                     + practica.getId());
372
373             }
374
375         }
376
377     }
378
379 }

```

```

376         } else {
377
378             System.out.println(x: "La práctica a la que le quieres asignar un alumno, no existe");
379
380         }
381     }
382 }
383
384 //9. Asignar un alumno a una práctica (alumno que realiza la práctica)
385 private static void asignarAlumnoPractica() {
386
387     System.out.println(x: "Introduce el ID de la práctica a asignar:");
388     int idPractica = sc.nextInt();
389     sc.nextLine();
390
391     //Recupero la práctica a la que asignar el alumno, comprobando que existe y que no tiene alumno asignado
392     Practica practica = practicaRepository.findOneById(id: idPractica);
393
394     if (practica != null) {
395
396         if (practica.getAlumno() == null) {
397
398             System.out.println(x: "Introduce el ID del alumno que quieres asignar a la práctica:");
399             int idAlumno = sc.nextInt();
400             sc.nextLine();
401
402             Alumno alumno = alumnoRepository.findOneById(id: idAlumno);
403
404             if (alumno != null) {
405
406                 //Asigno alumno
407                 practica.setAlumno(alumno);
408
409                 try {
410                     practicaRepository.update(p: practica);
411                 } catch (PersistenceException ex) {
412                     System.out.println(x: "El alumno ya tiene practica asignada");
413                 }
414             }
415         }
416         else {
417
418             System.out.println("La práctica con ID " + idPractica + " ya tiene alumno asignado");
419
420         }
421     }
422 }
423
424 //Comprueba si el alumno tiene una práctica asignada
425 private static boolean comprobarAlumnoPractica(Alumno a) {
426
427     ArrayList<Practica> practicas = (ArrayList<Practica>) practicaRepository.findAll();
428
429     for (Practica p : practicas) {
430
431         if (p.getAlumno() != null) {
432
433             if (p.getAlumno().getId() == a.getId()) {
434                 return true;
435             }
436         }
437     }
438
439     return false;
440 }
441
442 }
443
444 return false;
445
446 }
447
448 }
449

```

## Clase Alumno.java

```

1  package com.mycompany.garcia_cutillas_franciscojose_actut04.entities;
2
3  import com.mycompany.garcia_cutillas_franciscojose_actut04.excepciones.EdadException;
4  import java.io.Serializable;
5  import java.util.List;
6  import javax.persistence.Column;
7  import javax.persistence.Entity;
8  import javax.persistence.GeneratedValue;
9  import javax.persistence.GenerationType;
10 import javax.persistence.Id;
11 import javax.persistence.JoinColumn;
12 import javax.persistence.JoinTable;
13 import javax.persistence.ManyToMany;
14 import javax.persistence.OneToOne;
15 import javax.persistence.Table;
16
17 /**
18  *
19  * @author Fran
20  */
21 @Entity
22 @Table(name = "Alumnos", catalog = "fct")
23 public class Alumno implements Serializable {
24
25     @Id
26     @GeneratedValue(strategy = GenerationType.IDENTITY)
27     @Column(name = "alumno_id")
28     private int id;
29
30     @Column(name = "nombre_alumno")
31     private String nombre;
32
33     @Column(name = "edad")
34     private int edad;
35
36     @Column(name = "titulo")
37     private String titulo;
38
39     @OneToOne(mappedBy = "alumno")
40     private Practica practica;
41
42     @ManyToMany
43     @JoinTable(name = "AlumnosCandPracticas",
44         joinColumns = @JoinColumn(name = "alumno_id"),
45         inverseJoinColumns = @JoinColumn(name = "practica_id"))
46     private List<Practica> listaPracticas;
47
48     public Alumno() {
49     }
50
51     public Alumno(String nombre, int edad, String titulo) throws EdadException {
52         this.nombre = nombre;
53         setEdad(edad);
54         this.titulo = titulo;
55     }
56
57     public int getId() {
58         return id;
59     }
60
61     public void setId(int id) {
62         this.id = id;
63     }
64
65     public String getNombre() {
66         return nombre;
67     }

```

```

68
69     public void setNombre(String nombre) {
70         this.nombre = nombre;
71     }
72
73     public int getEdad() {
74         return edad;
75     }
76
77     public void setEdad(int edad) throws EdadException {
78         if (edad < 16) {
79             throw new EdadException();
80         } else {
81             this.edad = edad;
82         }
83     }
84
85     public String getTitulo() {
86         return titulo;
87     }
88
89     public void setTitulo(String titulo) {
90         this.titulo = titulo;
91     }
92
93     public Practica getPractica() {
94         return practica;
95     }
96
97     public void setPractica(Practica practica) {
98         this.practica = practica;
99     }
100
101     public List<Practica> getListasPracticas() {
102         return listaPracticas;
103     }
104
105     public void setListasPracticas(List<Practica> listaPracticas) {
106         this.listaPracticas = listaPracticas;
107     }
108
109     @Override
110     public int hashCode() {
111         int hash = 5;
112         hash = 37 * hash + this.id;
113         return hash;
114     }
115
116     @Override
117     public boolean equals(Object obj) {
118         if (this == obj) {
119             return true;
120         }
121         if (obj == null) {
122             return false;
123         }
124         if (getClass() != obj.getClass()) {
125             return false;
126         }
127         final Alumno other = (Alumno) obj;
128         return this.id == other.id;
129     }
130
131     @Override
132     public String toString() {
133         if (practica != null) {
134
135             return "Alumno con id " + id + ", Nombre= " + nombre + ", Edad= " + edad
136                 + ", Titulo= " + titulo + ", Empresa de prácticas= " + practica.getEmpresa().getNombre();
137
138         } else {
139
140             return "Alumno con id " + id + ", Nombre= " + nombre + ", Edad= " + edad
141                 + ", Titulo= " + titulo + ", Prácticas sin asignar";
142
143         }
144     }
145 }
146
147 }
148

```

## Clase Empresa.java

```

1 package com.mycompany.garcia_cutillas_franciscojose_actut04.entities;
2
3 import java.io.Serializable;
4 import java.util.List;
5 import javax.persistence.Column;
6 import javax.persistence.Entity;
7 import javax.persistence.GeneratedValue;
8 import javax.persistence.GenerationType;
9 import javax.persistence.Id;
10 import javax.persistence.OneToMany;
11 import javax.persistence.Table;
12
13 /**
14  *
15  * @author Fran
16  */
17 @Entity
18 @Table(name = "Empresas", catalog = "fct")
19 public class Empresa implements Serializable {
20
21     @Id
22     @GeneratedValue(strategy = GenerationType.IDENTITY)
23     @Column(name = "empresa_id")
24     private int id;
25
26     @Column(name = "nombre_empresa")
27     private String nombre;
28
29     @Column(name = "sector")
30     private String sector;
31
32     @OneToMany(mappedBy = "empresa")
33     private List<Practica> listaPracticas;
34
35     public Empresa() {
36     }
37
38     public Empresa(String nombre, String sector) {
39         this.nombre = nombre;
40         this.sector = sector;
41     }
42
43     public int getId() {
44         return id;
45     }
46
47     public void setId(int id) {
48         this.id = id;
49     }
50
51     public String getNombre() {
52         return nombre;
53     }
54
55     public void setNombre(String nombre) {
56         this.nombre = nombre;
57     }
58
59     public String getSector() {
60         return sector;
61     }
62
63     public void setSector(String sector) {
64         this.sector = sector;
65     }
66

```

```

67     public List<Practica> getListaPracticas() {
68         return listaPracticas;
69     }
70
71     public void setListaPracticas(List<Practica> listaPracticas) {
72         this.listaPracticas = listaPracticas;
73     }
74
75     @Override
76     public int hashCode() {
77         int hash = 5;
78         hash = 19 * hash + this.id;
79         return hash;
80     }
81
82     @Override
83     public boolean equals(Object obj) {
84         if (this == obj) {
85             return true;
86         }
87         if (obj == null) {
88             return false;
89         }
90         if (getClass() != obj.getClass()) {
91             return false;
92         }
93         final Empresa other = (Empresa) obj;
94         return this.id == other.id;
95     }
96
97     @Override
98     public String toString() {
99         if (listaPracticas.size() != 0) {
100             return "Empresa con id= " + id + ", Nombre= " + nombre + ", Sector= "
101                 + sector + ", Prácticas asignadas= " + listaPracticas.size();
102         } else {
103             return "Empresa con id= " + id + ", Nombre= " + nombre + ", Sector= "
104                 + sector + ", sin prácticas asignadas";
105         }
106     }
107 }
108

```

## Clase Practica.java

```

1  package com.mycompany.garcia_cutillas_franciscojose_actut04.entities;
2
3  import java.io.Serializable;
4  import java.sql.Date;
5  import java.util.List;
6  import javax.persistence.Column;
7  import javax.persistence.Entity;
8  import javax.persistence.GeneratedValue;
9  import javax.persistence.GenerationType;
10 import javax.persistence.Id;
11 import javax.persistence.JoinColumn;
12 import javax.persistence.ManyToMany;
13 import javax.persistence.ManyToOne;
14 import javax.persistence.OneToOne;
15 import javax.persistence.Table;
16
17 /**
18  *
19  * @author Fran
20  */
21 @Entity
22 @Table(name = "Practicas", catalog = "fct")
23 public class Practica implements Serializable {
24
25     @Id
26     @GeneratedValue(strategy = GenerationType.IDENTITY)
27     @Column(name = "practica_id")
28     private int id;
29
30     @Column(name = "fecha_inicio")
31     private Date fechaInicio;
32
33     @Column(name = "fecha_fin")
34     private Date fechaFin;
35
36     @Column(name = "descripcion")
37     private String descripcion;
38
39     @ManyToOne
40     @JoinColumn(name = "empresa_id")
41     private Empresa empresa;
42
43     @OneToOne
44     @JoinColumn(name = "alumno_id")
45     private Alumno alumno;
46
47     @ManyToMany(mappedBy = "listaPracticas")
48     private List<Alumno> listaAlumnos;
49
50     public Practica() {
51     }
52
53     public Practica(Date fechaInicio, Date fechaFin, String descripcion, Empresa empresa, Alumno alumno) {
54         this.fechaInicio = fechaInicio;
55         this.fechaFin = fechaFin;
56         this.descripcion = descripcion;
57         this.empresa = empresa;
58         this.alumno = alumno;
59     }
60
61     public Practica(Date fechaInicio, Date fechaFin, String descripcion, Empresa empresa) {
62         this.fechaInicio = fechaInicio;
63         this.fechaFin = fechaFin;
64         this.descripcion = descripcion;
65         this.empresa = empresa;
66     }
67

```

```
68 public int getId() {
69     return id;
70 }
71
72 public void setId(int id) {
73     this.id = id;
74 }
75
76 public Date getFechaInicio() {
77     return fechaInicio;
78 }
79
80 public void setFechaInicio(Date fechaInicio) {
81     this.fechaInicio = fechaInicio;
82 }
83
84 public Date getFechaFin() {
85     return fechaFin;
86 }
87
88 public void setFechaFin(Date fechaFin) {
89     this.fechaFin = fechaFin;
90 }
91
92 public String getDescripcion() {
93     return descripcion;
94 }
95
96 public void setDescripcion(String descripcion) {
97     this.descripcion = descripcion;
98 }
99
100 public Empresa getEmpresa() {
101     return empresa;
102 }
103
104 public void setEmpresa(Empresa empresa) {
105     this.empresa = empresa;
106 }
107
108 public Alumno getAlumno() {
109     return alumno;
110 }
111
112 public void setAlumno(Alumno alumno) {
113     this.alumno = alumno;
114 }
115
116 public List<Alumno> getListAlumnos() {
117     return listaAlumnos;
118 }
119
120 public void setListaAlumnos(List<Alumno> listaAlumnos) {
121     this.listaAlumnos = listaAlumnos;
122 }
123
124 @Override
125 public int hashCode() {
126     int hash = 5;
127     hash = 13 * hash + this.id;
128     return hash;
129 }
130
```



```

131     @Override
132     public boolean equals(Object obj) {
133         if (this == obj) {
134             return true;
135         }
136         if (obj == null) {
137             return false;
138         }
139         if (getClass() != obj.getClass()) {
140             return false;
141         }
142         final Practica other = (Practica) obj;
143         return this.id == other.id;
144     }
145
146     @Override
147     public String toString() {
148         if (alumno != null) {
149             return "Practica con id= " + id + ", Fecha de inicio= " + fechaInicio
150                 + ", Fecha de fin= " + fechaFin + ", Descripción= " + descripcion
151                 + ", ID de la empresa= " + empresa.getId() + ", ID del alumno= " + alumno.getId();
152         } else {
153             return "Practica con id= " + id + ", Fecha de inicio= " + fechaInicio
154                 + ", Fecha de fin= " + fechaFin + ", Descripción= " + descripcion
155                 + ", ID de la empresa= " + empresa.getId() + ", Alumno sin asignar ";
156         }
157     }
158 }
159
160 }
161
162 }
163
164 }
165

```

### Clase EdadException.java

```

1     package com.mycompany.garcia_cutillas_franciscojose_actut04.excepciones;
2
3     /**
4      *
5      * @author Fran
6      */
7     public class EdadException extends Exception {
8
9         @Override
10        public String getMessage() {
11            return "La edad introducida debe ser mayor o igual a 16 años.";
12        }
13
14    }
15

```

## Interfaz Repository.java

```

1  package com.mycompany.garcia_cutillas_franciscojose_actut04.repository;
2
3  import java.util.List;
4
5  /**
6   *
7   * @author Fran
8   */
9
10 public interface Repository<T> {
11
12     public void save(T t);
13
14     public List<T> findAll();
15
16     public T findOneById(int id);
17 }
18

```

## Clase AlumnoRepository.java

```

1  package com.mycompany.garcia_cutillas_franciscojose_actut04.repository;
2
3  import com.mycompany.garcia_cutillas_franciscojose_actut04.entities.Alumno;
4  import com.mycompany.garcia_cutillas_franciscojose_actut04.entities.Practica;
5  import com.mycompany.garcia_cutillas_franciscojose_actut04.util.HibernateUtil;
6  import java.util.List;
7  import javax.persistence.NoResultException;
8  import org.hibernate.query.Query;
9
10 /**
11  *
12  * @author Fran
13  */
14 public class AlumnoRepository implements Repository<Alumno> {
15
16     @Override
17     public void save(Alumno t) {
18         HibernateUtil.getCurrentSession().beginTransaction();
19         HibernateUtil.getCurrentSession().clear();
20         HibernateUtil.getCurrentSession().save(t);
21         HibernateUtil.getCurrentSession().getTransaction().commit();
22     }
23
24     @Override
25     public List<Alumno> findAll() {
26         Query<Alumno> query = HibernateUtil.getCurrentSession()
27             .createQuery("select a from Alumno a", type: Alumno.class);
28
29         return (List<Alumno>) query.getResultList();
30     }
31
32     @Override
33     public Alumno findOneById(int id) {
34         Query<Alumno> query = HibernateUtil.getCurrentSession()
35             .createQuery("select a from Alumno a where id = :idAlu", type: Alumno.class);
36         query.setParameter("idAlu", id);
37
38         try {
39             Alumno a = query.getSingleResult();
40             return a;
41         } catch (NoResultException ex) {
42             System.out.println("No existe un alumno con id: " + id + ", " + ex.getMessage());
43             return null;
44         }
45     }
46
47     public void saveCandidato(int id_alumno, int id_practica) {
48
49         PracticaRepository practicaRepository = new PracticaRepository();
50
51         HibernateUtil.getCurrentSession().beginTransaction();
52
53         Alumno alumno = findOneById(id_alumno);
54         Practica practica = practicaRepository.findOneById(id_practica);
55         alumno.getListPracticas().add(practica);
56
57         HibernateUtil.getCurrentSession().merge(alumno);
58         HibernateUtil.getCurrentSession().getTransaction().commit();
59     }
60 }
61

```

## Clase EmpresaRepository.java

```

1  package com.mycompany.garcia_cutillas_franciscojose_actut04.repository;
2
3  import com.mycompany.garcia_cutillas_franciscojose_actut04.entities.Empresa;
4  import com.mycompany.garcia_cutillas_franciscojose_actut04.util.HibernateUtil;
5  import java.util.List;
6  import javax.persistence.NoResultException;
7  import org.hibernate.query.Query;
8
9  /**
10   *
11   * @author Fran
12   */
13  public class EmpresaRepository implements Repository<Empresa> {
14
15      @Override
16      public void save(Empresa t) {
17          HibernateUtil.getCurrentSession().beginTransaction();
18          HibernateUtil.getCurrentSession().clear();
19          HibernateUtil.getCurrentSession().save(t);
20          HibernateUtil.getCurrentSession().getTransaction().commit();
21      }
22
23      @Override
24      public List<Empresa> findAll() {
25          Query<Empresa> query = HibernateUtil.getCurrentSession()
26              .createQuery("select e from Empresa e", type: Empresa.class);
27
28          return (List<Empresa>) query.getResultList();
29      }
30
31      @Override
32      public Empresa findOneById(int id) {
33          Query<Empresa> query = HibernateUtil.getCurrentSession()
34              .createQuery("select e from Empresa e where id = :idEmp", type: Empresa.class);
35          query.setParameter("idEmp", id);
36
37          try {
38              Empresa e = query.getSingleResult();
39              return e;
40          } catch (NoResultException ex) {
41
42              System.out.println("No existe una empresa con id: " + id + ", " + ex.getMessage());
43              return null;
44          }
45      }
46  }
47
48  }
49

```

## Clase PracticaRepository.java

```

1  package com.mycompany.garcia_cutillas_franciscojose_actut04.repository;
2
3  import com.mycompany.garcia_cutillas_franciscojose_actut04.entities.Practica;
4  import com.mycompany.garcia_cutillas_franciscojose_actut04.util.HibernateUtil;
5  import java.util.List;
6  import javax.persistence.NoResultException;
7  import org.hibernate.query.Query;
8
9  /**
10   *
11   * @author Fran
12   */
13  public class PracticaRepository implements Repository<Practica> {
14
15      @Override
16      public void save(Practica p) {
17          HibernateUtil.getCurrentSession().beginTransaction();
18          HibernateUtil.getCurrentSession().clear();
19          HibernateUtil.getCurrentSession().save(o: p);
20          HibernateUtil.getCurrentSession().getTransaction().commit();
21      }
22
23      @Override
24      public List<Practica> findAll() {
25          Query<Practica> query = HibernateUtil.getCurrentSession()
26              .createQuery(string: "select p from Practica p", type: Practica.class);
27
28          return (List<Practica>) query.getResultList();
29      }
30
31      @Override
32      public Practica findOneById(int id) throws NoResultException {
33          Query<Practica> query = HibernateUtil.getCurrentSession()
34              .createQuery(string: "select p from Practica p where id = :idPra", type: Practica.class);
35          query.setParameter(string: "idPra", o: id);
36
37          try {
38              Practica p = query.getSingleResult();
39              return p;
40          } catch (NoResultException ex) {
41
42              System.out.println("No existe una práctica con id: " + id + ", " + ex.getMessage());
43              return null;
44          }
45      }
46
47      public void update(Practica p) {
48          HibernateUtil.getCurrentSession().beginTransaction();
49          HibernateUtil.getCurrentSession().clear();
50          HibernateUtil.getCurrentSession().update(o: p);
51          HibernateUtil.getCurrentSession().getTransaction().commit();
52      }
53
54  }
55
56  }

```

## Clase FechaUtil.java

```

1 package com.mycompany.garcia_cutillas_franciscojose_actut04.util;
2
3 import java.sql.Date;
4 import java.time.LocalDate;
5 import java.time.format.DateTimeFormatter;
6
7 /**
8  *
9  * @author Fran
10  */
11 public class FechaUtil {
12
13     public static Date convierteFecha(String fechaSpain) {
14
15         DateTimeFormatter formatoSpain = DateTimeFormatter.ofPattern("dd/MM/yyyy");
16
17         LocalDate fechaLocalDate = LocalDate.parse(text: fechaSpain, formatter: formatoSpain);
18
19         return Date.valueOf(date: fechaLocalDate);
20
21     }
22
23 }
24

```

## Clase HibernateUtil.java

```

1 package com.mycompany.garcia_cutillas_franciscojose_actut04.util;
2
3 import com.mycompany.garcia_cutillas_franciscojose_actut04.entities.Alumno;
4 import com.mycompany.garcia_cutillas_franciscojose_actut04.entities.Empresa;
5 import com.mycompany.garcia_cutillas_franciscojose_actut04.entities.Practica;
6 import org.hibernate.Session;
7 import org.hibernate.SessionFactory;
8 import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
9 import org.hibernate.cfg.Configuration;
10 import org.hibernate.service.ServiceRegistry;
11
12 /**
13  *
14  * @author Fran
15  */
16 public class HibernateUtil {
17
18     private static SessionFactory sessionFactory;
19     private static Session session;
20
21     //Crea la factoria de sesiones
22     public static void buildSessionFactory() {
23
24         Configuration configuration = new Configuration();
25         configuration.configure(resource: "hibernate.cfg.xml");
26
27         configuration.addAnnotatedClass(annotatedClass: Alumno.class);
28         configuration.addAnnotatedClass(annotatedClass: Empresa.class);
29         configuration.addAnnotatedClass(annotatedClass: Practica.class);
30
31         ServiceRegistry serviceRegistry = new StandardServiceRegistryBuilder()
32             .configure(resourceName: "hibernate.cfg.xml").build();
33         sessionFactory = configuration.buildSessionFactory(serviceRegistry);
34
35     }
36
37     //Abre una nueva sesión
38     public static void openSession() {
39
40         session = sessionFactory.openSession();
41
42     }
43
44     //Devuelve la sesión actual
45     public static Session getCurrentSession() {
46
47         if ((session == null) || (!session.isOpen())) {
48
49             openSession();
50
51         }
52
53         return session;
54
55     }
56

```

```

57 //Cierra Hibernate
58 public static void closeSessionFactory() {
59
60     if (session != null) {
61         session.close();
62     }
63
64     if (sessionFactory != null) {
65         sessionFactory.close();
66     }
67
68 }
69
70 }
71
72 }
73
74 }
75

```

### Clase SiNoUtil.java

```

1 package com.mycompany.garcia_cutillas_franciscojose_actut04.util;
2
3 /**
4  *
5  * @author Fran
6  */
7 public class SiNoUtil {
8
9     public static boolean comprobarSiNo(String entrada) {
10
11         boolean correcto = false;
12
13         if (entrada.equalsIgnoreCase("S") || entrada.equalsIgnoreCase("N")) {
14
15             correcto = true;
16             return correcto;
17
18         }
19
20         return correcto;
21
22     }
23
24 }
25

```

### Fichero de configuración hibernate.cfg.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
3      "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
4 <hibernate-configuration>
5     <session-factory>
6         <property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
7         <property name="hibernate.connection.url">jdbc:mysql://localhost/fct</property>
8         <property name="hibernate.connection.username">root</property>
9         <property name="hibernate.connection.password">root</property>
10        <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>
11        <property name="hibernate.hbm2ddl.auto">none</property>
12        <property name="hibernate.show_sql">true</property>
13    </session-factory>
14 </hibernate-configuration>
15
16

```

## Fichero pom.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.myccompany</groupId>
5   <artifactId>garcia_cutillas_franciscojose_actut04</artifactId>
6   <version>1.0-SNAPSHOT</version>
7   <packaging>jar</packaging>
8   <properties>
9     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
10    <maven.compiler.source>19</maven.compiler.source>
11    <maven.compiler.target>19</maven.compiler.target>
12    <exec.mainClass>com.myccompany.garcia_cutillas_franciscojose_actut04.Garcia_Cutillas_FranciscoJose_Actut04</exec.mainClass>
13  </properties>
14  <dependencies>
15    <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
16    <dependency>
17      <groupId>org.hibernate</groupId>
18      <artifactId>hibernate-core</artifactId>
19      <version>5.6.9.Final</version>
20    </dependency>
21    <dependency>
22      <groupId>mysql</groupId>
23      <artifactId>mysql-connector-java</artifactId>
24      <version>8.0.19</version>
25    </dependency>
26  </dependencies>
27 </project>

```