

Tarea online

Título de la tarea: Operaciones CRUD sobre una base de datos H2

Unidad: 10

Ciclo formativo y módulo: DAM/DAW, Programación.

Curso académico: 2025/26

¿Qué contenidos o resultados de aprendizaje trabajaremos?

RA 8: Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

RA 9: Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.

Aviso

Esta tarea solo deberá realizarla el alumnado que no esté realizando la FEOE durante el presente curso.

1.- Descripción de la tarea

Caso práctico



Por Zacharie Grossen - Trabajo propio. K2 (CC BY-SA)

Exploradores del Mundo S.L. es una empresa española de turismo de aventura especializada en expediciones a alta montaña. Opera en colaboración con federaciones de alpinismo de varios países y gestiona anualmente una gran cantidad de solicitudes de ascensión a picos de las principales cordilleras del mundo (Himalayas, Andes, Alpes y Karakórum).

Actualmente, toda la información sobre alpinistas, picos y expediciones se gestiona mediante hojas de cálculo, lo que genera duplicidades, pérdidas de datos y dificultades para generar informes. La dirección técnica ha decidido migrar este sistema a una base de datos relacional y ha contratado a **David** como consultor de desarrollo.

David deberá diseñar e implementar una base de datos relacional que permita a Exploradores del Mundo S.L. gestionar de forma eficiente el registro de alpinistas, los picos disponibles y las ascensiones realizadas.

¿Qué te pedimos que hagas?

Las actividades a realizar se centrarán en la creación de las tablas de una base de datos, así como la realización de operaciones sobre ella. Como sistema gestor de base de datos hemos escogido **H2** (<https://www.h2database.com>), por la facilidad de integración en Java, ya que no necesita una instalación aparte, sino que basta con incluir en nuestro proyecto la librería (archivo **JAR**) que se puede descargar del sitio web.

En esta tarea, tu misión es completar los ficheros `crearBD.sql` y `cargaBD.sql`, así como los métodos de la clase `Aplicacion` para que la funcionalidad del programa sea completa.

Como verás, al ejecutar el proyecto plantilla que te proporcionamos te aparecerá lo siguiente en la consola.

```
¡¡Atención!!
```

```
Mientras tu aplicación se esté ejecutando  
puedes acceder a la consola de la base de datos
```

a través del navegador web.

Página local: <http://172.30.0.53:8082>

Datos de acceso

Controlador: org.h2.Driver

URL JDBC: jdbc:h2:./proyectobase.h2db

Usuario: (no indicar nada)

Password: (no indicar nada)

| MENÚ DE LA APLICACIÓN |

-
- 1 - Mostrar picos más altos del mundo
 - 2 - Borrar alpinista
 - 3 - Modificar nacionalidad alpinista
 - 4 - Mostrar ascensiones de cada pico
 - 0 - Salir
-

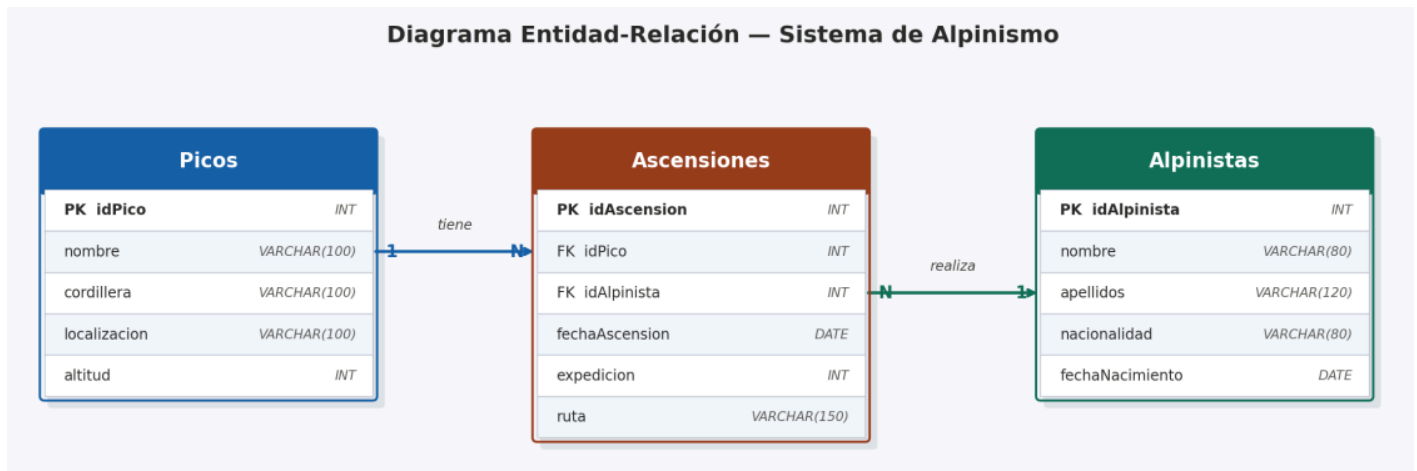
Escriba opción:

El programa intentará crear primero las tablas y luego insertar los datos. Por ello, tendrás que rellenar los ficheros .sql que tienes en la plantilla del proyecto con sentencias SQL necesarias para crear las distintas tablas e insertar la información de los picos, alpinistas y ascensiones.

Posteriormente vamos a ir haciendo ejercicios y aumentando su complejidad: realizando consultas, modificando y borrando registros.

1.1.- Ejercicio 1: Crear las tablas de la base de datos

Lo primero que deberías hacer es crear la estructura de la base de datos. Para ello, rellena el fichero `creaBD.sql` que tienes en el paquete `resources` del proyecto que te proporcionamos como plantilla con las sentencias adecuadas para crear las tablas necesarias. En el siguiente diagrama entidad-relación puedes ver las tablas a crear.



Luis M. Salmerón ([CC BY-NC](https://creativecommons.org/licenses/by-nc/4.0/))

Como puedes ver en el diagrama anterior, necesitas tres tablas:

- ✓ **Picos**, donde se almacenarán los nombres, cordilleras, localizaciones y altitudes de los picos más altos del mundo. Observa que `idPico` es la clave primaria y será generado de forma automática.
- ✓ **Alpinistas**, donde se almacenarán el nombre y apellidos de cada alpinista, así como su nacionalidad y fecha de nacimiento. Observa que `idAlpinista` es la clave primaria y será generado de forma automática.
- ✓ **Ascensiones**, donde se pretende almacenar la relación de cuales películas están disponibles en cada una de las plataformas.
 - ◆ Aquí la clave primaria es `idAscension` y será generado de forma automática.
 - ◆ Ten en cuenta que `idPico` será clave foránea referenciando a `idPico` de la tabla **Picos**.
 - ◆ Asimismo, `idAlpinista` será clave foránea referenciando a `idAlpinista` de la tabla **Alpinistas**.
 - ◆ Las restricciones de actualización y borrado se tendrán que definir en cascada. Para ello, al definir la tabla **Ascensiones**, deberás emplear restricción de clave foránea (foreign key) haciendo uso de `ON UPDATE CASCADE ON DELETE CASCADE`.

1.2.- Ejercicio 2: Rellenar las tablas de la base de datos

En el fichero `cargaBD.sql` debes rellenar las sentencias SQL necesarias para insertar los datos en las tablas, de modo que tras su ejecución el contenido de las mismas sea el que se puede ver en las imágenes:

Tabla Picos

IDPICO	NOMBRE	CORDILLERA	LOCALIZACION	ALTITUD
1	Everest	Himalaya	Nepal / China	8849
2	K2	Karakórum	Pakistán / China	8611
3	Kangchenjunga	Himalaya	Nepal / India	8586
4	Lhotse	Himalaya	Nepal / China	8516
5	Makalu	Himalaya	Nepal / China	8485

Luis M. Salmerón (generada desde H2) ([CC BY-NC](#))

Tabla Alpinista

IDALPINISTA	NOMBRE	APELLIDOS	NACIONALIDAD	FECHANACIMIENTO
1	Edmund	Hillary	Neozelandés	1919-07-20
2	Tenzing	Norgay	Nepalés	1914-05-29
3	Ernst	Schmied	Suizo	1905-09-17
4	Jürg	Marmet	Suizo	1918-04-06
5	Achille	Compagnoni	Italiano	1914-09-26
6	Lino	Lacedelli	Italiano	1925-05-05
7	George	Bell	Estadounidense	1919-06-22
8	Charles	Houston	Estadounidense	1913-08-24
9	George	Band	Británico	1929-03-07
10	Joe	Brown	Británico	1930-10-26
11	Fritz	Luchsinger	Suizo	1921-01-07
12	Ernst	Reiss	Suizo	1922-04-02
13	Jean	Couzy	Francés	1923-09-07
14	Lionel	Terray	Francés	1921-07-25

Luis M. Salmerón (generada desde H2) ([CC BY-NC](#))

Tabla Ascensiones

IDASCENSION	IDPICO	IDALPINISTA	FECHAASCENSION	EXPEDICION	RUTA
1	1	1	1953-05-29	1	Cresta Sur
2	1	2	1953-05-29	1	Cresta Sur
3	1	3	1956-05-23	2	Cresta Sur
4	1	4	1956-05-23	2	Cresta Sur
5	2	5	1954-07-31	1	Espolón Abruzzi
6	2	6	1954-07-31	1	Espolón Abruzzi
7	2	7	1978-08-06	2	Espolón Abruzzi
8	2	8	1978-08-06	2	Espolón Abruzzi
9	3	9	1955-05-25	1	Cara Suroeste
10	3	10	1955-05-25	1	Cara Suroeste
11	4	11	1956-05-18	1	Cara Oeste
12	4	12	1956-05-18	1	Cara Oeste
13	5	13	1955-05-15	1	Pilar Noroeste
14	5	14	1955-05-15	1	Pilar Noroeste

Luis M. Salmerón (generada desde H2) ([CC BY-NC](#))

Recuerda que puedes ver cómodamente el contenido de las tablas desde la consola web: <http://localhost:8082>

En el fichero cargaDB.sql os dejamos los datos para que vosotros los introduzcáis en la base de datos.

1.3.- Ejercicio 3: Implementar el método mostrarPicos

Al ejecutar la opción 1 "Mostrar picos", nos debería aparecer por consola un listado de los picos almacenados en la base de datos:

```
-----  
| MENÚ DE LA APLICACIÓN |  
-----  
1 - Mostrar picos más altos del mundo  
2 - Borrar alpinista  
3 - Modificar nacionalidad alpinista  
4 - Mostrar ascensiones de cada pico  
0 - Salir  
-----
```

Escriba opción:

1

----- Listado de Picos más altos del mundo -----

Id	Nombre	Cordillera	Localización	Altitud (m)
1	Everest	Himalaya	Nepal / China	8849
2	K2	Karakórum	Pakistán / China	8611
3	Kangchenjunga	Himalaya	Nepal / India	8586
4	Lhotse	Himalaya	Nepal / China	8516
5	Makalu	Himalaya	Nepal / China	8485

Para ello, debes implementar el método `mostrarPicos` de la clase `Aplicacion` con las sentencias adecuadas para realizar la consulta en la base de datos y volcar el resultado en la consola.

Ten en cuenta que las sentencias preparadas (`PreparedStatement`) **mejoran el rendimiento** de la aplicación al disminuir el tiempo de análisis, **minimizan el ancho de banda consumido en el servidor** (solo se necesita enviar los parámetros cada vez, no la consulta entera) y **evitan problemas de seguridad** ante potenciales inyecciones SQL.

Por tanto, **en aquellas sentencias SQL de tu código que incorporen algún tipo de parámetro (por ejemplo en las actualizaciones o borrados), debes procurar utilizar sentencias preparadas (`PreparedStatement`).**

1.4.- Ejercicio 4: Implementar el método borrarAlpinista

Debes implementar el método `borrarAlpinista` para que será invocado al ejecutar la opción 2 "Borrar alpinista", como vemos a continuación en este ejemplo de ejecución. Comenzaremos listando los alpinistas almacenados en la base de datos junto a su identificador, el cual necesitaremos para poder eliminarlo. Si el alpinista está referenciado podrá ser borrado, ya que se ha definido el borrado en cascada al crear las distintas tablas:

A continuación se muestra un ejemplo de borrado, concretamente se borra el alpinista con identificador 14:

```
-----  
| MENÚ DE LA APLICACIÓN |  
-----
```

```
1 - Mostrar picos más altos del mundo  
2 - Borrar alpinista  
3 - Modificar nacionalidad alpinista  
4 - Mostrar ascensiones de cada pico  
0 - Salir  
-----
```

Escriba opción:

2

```
----- Listado de alpinistas -----
```

```
Id      Nombre      Apellidos  
-----
```

```
1      Edmund      Hillary  
2      Tenzing     Norgay  
3      Ernst       Schmied  
4      Jürg        Marmet  
5      Achille     Compagnoni  
6      Lino        Lacedelli  
7      George      Bell  
8      Charles     Houston  
9      George      Band  
10     Joe         Brown  
11     Fritz       Luchsinger  
12     Ernst       Reiss  
13     Jean        Couzy  
14     Lionel      Terray
```

Escriba el identificador del alpinista a borrar:

14

Borrado correctamente el alpinista con identificador 14.

Si se introduce un identificador que no existe en la base de datos el programa debe informar sobre ello:

```
-----  
| MENÚ DE LA APLICACIÓN |  
-----
```

- 1 - Mostrar picos más altos del mundo
 - 2 - Borrar alpinista
 - 3 - Modificar nacionalidad alpinista
 - 4 - Mostrar ascensiones de cada pico
 - 0 - Salir
- ```

```

Escriba opción:

2

```
----- Listado de alpinistas -----
```

| Id | Nombre  | Apellidos  |
|----|---------|------------|
| 1  | Edmund  | Hillary    |
| 2  | Tenzing | Norgay     |
| 3  | Ernst   | Schmied    |
| 4  | Jürg    | Marmet     |
| 5  | Achille | Compagnoni |
| 6  | Lino    | Lacedelli  |
| 7  | George  | Bell       |
| 8  | Charles | Houston    |
| 9  | George  | Band       |
| 10 | Joe     | Brown      |
| 11 | Fritz   | Luchsinger |
| 12 | Ernst   | Reiss      |
| 13 | Jean    | Couzy      |

Escriba el identificador del alpinista a borrar:

20

No se realizó ninguna modificación en la base de datos.

Ten en cuenta que las sentencias preparadas (**PreparedStatement**) **mejoran el rendimiento** de la aplicación al disminuir el tiempo de análisis, **minimizan el ancho de banda consumido** en el servidor (solo se necesita enviar los parámetros cada vez, no la consulta entera) **y evitan problemas de seguridad** ante potenciales inyecciones SQL.

Por tanto, **en aquellas sentencias SQL de tu código que incorporen algún tipo de parámetro (por ejemplo en las actualizaciones o borrados), debes procurar utilizar sentencias preparadas (PreparedStatement).**

# 1.5.- Ejercicio 5: Implementar el método modificarNacionalidadAlpinista

Debes implementar el método `modificarNacionalidadAlpinista` que será ejecutado al seleccionar la opción 3 "Modificar nacionalidad alpinista" donde podremos modificar la nacionalidad de un alpinista. Como se muestra en el siguiente ejemplo de ejecución, al seleccionar la opción nº 3, comenzaremos listando los distintos alpinistas existentes junto a su identificador, el cual necesitaremos para poder modificarlo. A continuación, pediremos la nueva nacionalidad que queremos modificar y se procederá a la modificación.

Si se introduce un identificador correcto, se debe modificar el campo nacionalidad y mostrar un mensaje indicando que se ha modificado correctamente:

```

MENÚ DE LA APLICACIÓN
1 - Mostrar picos más altos del mundo
2 - Borrar alpinista
3 - Modificar nacionalidad alpinista
4 - Mostrar ascensiones de cada pico
0 - Salir

```

Escriba opción:

3

```
----- Listado de alpinistas -----
Id Nombre Apellidos Nacionalidad

1 Edmund Hillary Neozelandés
2 Tenzing Norgay Nepalés
3 Ernst Schmied Suizo
4 Jürg Marmet Suizo
5 Achille Compagnoni Italiano
6 Lino Lacedelli Italiano
7 George Bell Estadounidense
8 Charles Houston Estadounidense
9 George Band Británico
10 Joe Brown Británico
11 Fritz Luchsinger Suizo
12 Ernst Reiss Suizo
13 Jean Couzy Francés
14 Lionel Terray Francés
```

Escriba el identificador del alpinista a modificar:

2

Introduzca la nueva nacionalidad del alpinista:

Nepalí

Modificación correctamente el alpinista con identificador 2.

En el caso de introducir un identificador incorrecto (no asociado a ningún alpinista), al no producirse modificación alguna, se informará también por consola:

```

MENÚ DE LA APLICACIÓN
1 - Mostrar picos más altos del mundo
2 - Borrar alpinista
3 - Modificar nacionalidad alpinista
4 - Mostrar ascensiones de cada pico
0 - Salir

```

Escriba opción:

3

```
----- Listado de alpinistas -----
Id Nombre Apellidos Nacionalidad

1 Edmund Hillary Neozelandés
2 Tenzing Norgay Nepalí
3 Ernst Schmied Suizo
4 Jürg Marmet Suizo
5 Achille Compagnoni Italiano
6 Lino Lacedelli Italiano
7 George Bell Estadounidense
8 Charles Houston Estadounidense
9 George Band Británico
10 Joe Brown Británico
11 Fritz Luchsinger Suizo
12 Ernst Reiss Suizo
13 Jean Couzy Francés
14 Lionel Terray Francés
```

Escriba el identificador del alpinista a modificar:

20

Introduzca la nueva nacionalidad del alpinista:

Inglés

No se realizó ninguna modificación en los datos de los alpinistas.

Ten en cuenta que las sentencias preparadas (PreparedStatement) **mejoran el rendimiento** de la aplicación al disminuir el tiempo de análisis, **minimizan el ancho de banda consumido** en el servidor (solo se necesita enviar los parámetros cada vez, no la consulta entera) y **evitan problemas de seguridad** ante potenciales inyecciones SQL.

Por tanto, en aquellas sentencias SQL de tu código que incorporen algún tipo de parámetro (por ejemplo en las actualizaciones o borrados), debes procurar utilizar sentencias preparadas (PreparedStatement).

# 1.6.- Ejercicio 6: Implementar el método mostrarAscensionesPicos

Al seleccionar la opción nº 4 "Mostrar ascensiones de cada pico" se deberá ejecutar el método `mostrarAscensionesPicos`. Como se muestra a continuación, dicho método será el encargado de imprimir por consola todas las ascensiones asociadas a cada uno de los picos. Por ello, tienes que implementar el método `mostrarAscensionesPicos` de la clase `Aplicacion` con las sentencias adecuadas para realizar la consulta en la base de datos y volcar el resultado en la consola.

```

MENÚ DE LA APLICACIÓN
```

- 1 - Mostrar picos más altos del mundo
  - 2 - Borrar alpinista
  - 3 - Modificar nacionalidad alpinista
  - 4 - Mostrar ascensiones de cada pico
  - 0 - Salir
- ```
-----
```

Escriba opción:

4

```
-----  
| ASCENSIONES A PICOS |  
-----
```

```
----- Everest -----
```

Nombre alpinista	Fecha	Ruta ascenso
Edmund Hillary	1953-05-29	Cresta Sur
Tenzing Norgay	1953-05-29	Cresta Sur
Ernst Schmied	1956-05-23	Cresta Sur
Jürg Marmet	1956-05-23	Cresta Sur

```
----- K2 -----
```

Nombre alpinista	Fecha	Ruta ascenso
Achille Compagnoni	1954-07-31	Espolón Abruzzi
Lino Lacedelli	1954-07-31	Espolón Abruzzi
George Bell	1978-08-06	Espolón Abruzzi
Charles Houston	1978-08-06	Espolón Abruzzi

```
----- Kangchenjunga -----
```

Nombre alpinista		Fecha	Ruta ascenso
George	Band	1955-05-25	Cara Suroeste
Joe	Brown	1955-05-25	Cara Suroeste

----- Lhotse -----

Nombre alpinista		Fecha	Ruta ascenso
Fritz	Luchsinger	1956-05-18	Cara Oeste
Ernst	Reiss	1956-05-18	Cara Oeste

----- Makalu -----

Nombre alpinista		Fecha	Ruta ascenso
Jean	Couzy	1955-05-15	Pilar Noroeste
Lionel	Terray	1955-05-15	Pilar Noroeste

Como ayuda, la lógica que debes usar es la siguiente: "Para cada pico existente en la tabla Picos, debes mostrar el nombre del pico en la consola y buscar las ascensiones asociadas a ese pico en la tabla Ascensiones. Para cada alpinista, usando su identificador, buscar el nombre y apellidos mostrando también la fecha de ascenso y la ruta utilizada para ello". Realizar esto es más sencillo de lo que a priori pueda parecer dado que cuando hacemos una consulta trabajamos con los resultados usando bucles, lo que simplifica trabajar con los resultados "de uno en uno" (`while (resultados.next())` en lugar de tener que hacer consultas avanzadas con SQL.

1.7.- Ejercicio 7: Bases de datos OO



Sebastián López, elaboración propia usando ChatGPT-OpenAI ([CC BY-NC](https://creativecommons.org/licenses/by-nc/4.0/))

Contesta a las siguientes preguntas referentes a las bases de datos orientadas a objetos.

[Cuestionario bases de datos orientadas a objetos](#) (doc - 21504 B)

Para realizar este ejercicio renombra el fichero como

"Apellido1_Apellido2_Nombre_Tarea_2025_26_ParteOO".

Una vez hayas terminado de cumplimentar el cuestionario, genera un PDF a partir de él para evitar problemas de formato a la hora de corregirlo. ¡Y no olvides incluir ese documento en el paquete junto con el resto de tu proyecto! (En el zip generado exportando tu proyecto de NetBeans y que has nombrado incluyendo tu nombre y apellidos, añade el .pdf que has generado).

2.- Información de interés

Recursos necesarios y recomendaciones

Como recursos para la tarea:

- ✓ Computador personal.
- ✓ JDK y JRE de Java SE. SDK de JavaFX
- ✓ Entorno de desarrollo NetBeans.
- ✓ En la página web del servidor de base de datos H2 existe una referencia del lenguaje SQL [referencia SQL](#).
- ✓ Para la tarea, descomprime y trabaja con el siguiente proyecto:

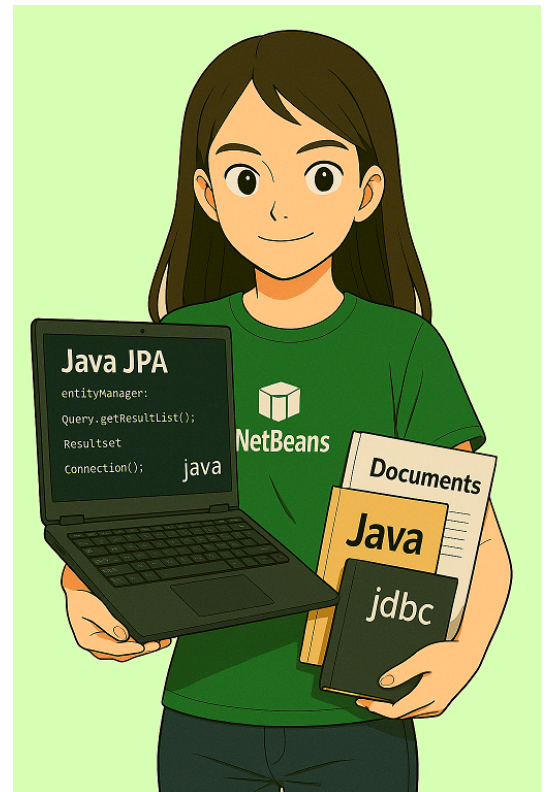
Es obligatorio utilizar este proyecto con sus clases y programas. Deberás utilizarlo para hacer las modificaciones y añadidos que se indican en el Caso Práctico. Por las dependencias de archivos contenidos en el proyecto **es recomendable que hagas un "Clean and Build" del proyecto** en tu equipo antes de ejecutarlo.

En el proyecto base encontrarás algunas variables y métodos ya implementados, quedando por hacer aquellos donde debas modificar alguna característica de los componentes de la interfaz. Podemos descargar el proyecto base aquí:

[PROG10_Tarea_2025_26_Alumnado.zip](#) (zip - 95 KB)

Como consejos:

- ✓ **Indenta** tu código de forma adecuada. Recuerda que NetBeans puede hacer esto por ti (Source > Format).
- ✓ Procura respetar las convenciones de **nombrado de identificadores** indicadas en la unidad 1. Recuerda que en NetBeans puedes renombrar rápidamente una variable, un método o una clase usando la opción Refactor.



Sebastián López, usando ChatGPT-OpenAI ([CC BY-NC](#))

Para saber más

Una vez realizada la tarea, el envío se realizará a través de la plataforma. Comprime la carpeta del proyecto NetBeans en un fichero .zip y nómbralo siguiendo las siguientes pautas:

3.- Evaluación de la tarea

Criterios de evaluación implicados

RA 8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

- ✓ a) Se han identificado las características de las bases de datos orientadas a objetos.
- ✓ b) Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.
- ✓ c) Se han instalado sistemas gestores de bases de datos orientados a objetos.
- ✓ d) Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.
- ✓ e) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.
- ✓ f) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.
- ✓ g) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.
- ✓ h) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.



[Peggy Marco](#) (Pixabay License)

RA9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.

- ✓ a) Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos.
- ✓ b) Se han programado conexiones con bases de datos.
- ✓ c) Se ha escrito código para almacenar información en bases de datos.
- ✓ d) Se ha escrito código para recuperar y mostrar información almacenada en bases de datos.
- ✓ e) Se han efectuado borrados y modificaciones sobre la información almacenada.
- ✓ f) Se han creado aplicaciones que muestren la información almacenada en bases de datos.
- ✓ g) Se han creado aplicaciones para gestionar la información presente en bases de datos.

¿Cómo valoramos y puntuamos tu tarea?

La rúbrica, así como la ponderación para cada uno de los ítems, es la siguiente:

(RA9.a, RA9.g, RA9.c) Ejercicio 1. Creación de tablas y claves

No se crean correctamente las tablas o presentan errores estructurales. No se definen claves primarias o las relaciones entre tablas son inexistentes o incorrectas.	0 puntos.
Se crean las tablas, pero con errores en la definición de claves primarias o foráneas, con tipos de datos inadecuados o restricciones no aplicadas.	1 punto.
Se crean correctamente las tablas con sus claves primarias y foráneas bien definidas. El diseño es consistente y adecuado (tipos de datos correctos, restricciones bien aplicadas).	2 puntos.
(RA9.c, RA9.h) Ejercicio 2. Inserción de datos en las tablas	
No se insertan los datos o se producen errores que impiden poblar las tablas correctamente.	0 puntos.
Se insertan algunos datos, pero con errores (duplicados, violaciones de integridad, orden incorrecto de inserción, etc.).	0,5 puntos.
Se insertan correctamente todos los datos en las tablas respetando las relaciones entre ellas.	1 punto.
(RA9.b, RA9.d, RA9.f) Ejercicio 3. Método mostrarPicos	
No se implementa correctamente la consulta o no se muestran los datos de los picos.	0 puntos.
Se realiza la consulta, pero la información mostrada es incompleta o presenta errores de formato.	0,75 puntos.
Se implementa la consulta de forma adecuada y se muestran correctamente los datos de los picos.	1,5 puntos.
(RA9.b, RA9.e) Ejercicio 4. Método modificarNacionalidadAlpinista	
No se implementa la modificación o no funciona correctamente.	0 puntos.
Se implementa parcialmente, pero con errores en la actualización o en la recogida de datos.	0,75 puntos.
Se modifica correctamente la nacionalidad, se gestionan casos como identificadores inexistentes y se informa del resultado.	1.5 puntos.
(RA9.b, RA9.e) Ejercicio 5. Método borrarAlpinista	
No se implementa correctamente el borrado o provoca errores en la ejecución.	0 puntos.
Se realiza el borrado, pero sin controlar correctamente la selección del alpinista o el resultado de la operación.	0,75 puntos.

Se borra correctamente el alpinista y se gestionan errores como restricciones de integridad referencial.	1,5 puntos.
(RA9.b, RA9.d, RA9.f) Ejercicio 6. Método mostrarAscensionesPicos	
No se implementa correctamente la consulta o no se muestran las ascensiones.	0 puntos.
Se muestran datos, pero sin relacionar correctamente picos y alpinistas o con información incompleta.	0,75 puntos.
Se muestran correctamente las ascensiones de cada pico con nombre del alpinista, fecha y ruta.	1,5 puntos.
(RA8.a, RA8.b, RA8.c, RA8.d, RA8.e, RA8.f) Ejercicio 7. Preguntas 2 a 6 – Conceptos de JPA y persistencia	
No se responde correctamente ninguna de las preguntas.	0 puntos.
Se responde correctamente 1 pregunta.	0,1 puntos.
Se responde correctamente 2 preguntas.	0,2 puntos.
Se responde correctamente 3 preguntas.	0,3 puntos.
Se responde correctamente 4 preguntas.	0,4 puntos.
Se responden correctamente todas las preguntas.	0,5 puntos.
(RA8.g, RA8.h, RA8.c, RA8.d, RA8.e, RA8.f) Ejercicio 7. Preguntas 1 y 7 – Persistencia y desfase objeto-relacional	
No comprende el concepto de persistencia de objetos ni el desfase objeto-relacional. Las respuestas son incorrectas.	0 puntos.
Comprende parcialmente los conceptos, dado que una de las respuestas es incorrecta.	0,25 puntos.
Comprende correctamente la finalidad de JPA como mecanismo de persistencia de objetos y el concepto de impedancia objeto-relacional, respondiendo adecuadamente a las preguntas.	0,5 puntos.

IMPORTANTE: Motivos de devolución de la tarea o de calificación mínima

La tarea obtendrá automáticamente una calificación de **cero** si:

- ✓ No se entrega **un único proyecto** que incluya el ejercicio que se pide en la tarea utilizando el IDE NetBeans.
- ✓ El programa presenta **errores de compilación** impidiendo su ejecución y prueba.
- ✓ **No se respeta el proyecto base** que se entrega al alumno para realizar la tarea.
- ✓ Se utilizan **contenidos** en alguno de los ejercicios del proyecto **que no se han estudiado** en la unidad actual o anteriores a la misma.
- ✓ Se presenta un proyecto total o parcialmente copiado de otro alumno.

Recordatorio: Si una tarea obtiene calificación cero por alguno de los motivos previamente mencionados, será devuelta al alumno. Este tendrá derecho a una segunda entrega únicamente en los siguientes casos:

- Que se trate de la primera vez que presenta la tarea.
- Que la fecha de entrega inicial haya sido al menos una semana antes de la fecha límite establecida para dicha tarea.