

FAMILIA PROFESIONAL:

CICLOS FORMATIVOS:

MÓDULO:

Informática y Comunicaciones

Desarrollo de Aplicaciones Multiplataforma,

Desarrollo de Aplicaciones Web

Programación

UNIDAD 11: BASES DE DATOS RELACIONALES

ACTIVIDADES



AUTORES: **Fernando Rodríguez Alonso**
Sonia Pasamar Franco

Este documento está bajo licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional License.

Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

Usted es libre de:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato.

El licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia.

Bajo los siguientes términos:

- **Atribución** — Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.
- **NoComercial** — Usted no puede hacer uso del material con propósitos comerciales.
- **SinDerivadas** — Si remezcla, transforma o crea a partir del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

BASE DE DATOS personal

Crea en **SQLite** un esquema de base de datos denominado **personal** y compuesto por dos tablas:

1) departamento:

- código: número entero, no nulo, clave primaria.
- nombre: cadena de texto, no nulo.
- ubicación: cadena de texto, no nulo.

2) empleado:

- código: número entero, no nulo, clave primaria.
- nombre: cadena de texto, no nulo.
- fecha_alta: cadena de texto, no nulo.
- salario: número real, no nulo.
- código_departamento: número entero, no nulo, clave ajena de departamento(código).

ACTIVIDAD 1x01

Codifica una clase **Actividad_1x01** que incluya un programa principal **main**. Este programa gestionará la tabla **departamento** de la base de datos **personal** mediante el siguiente menú de opciones:

0) Salir del programa.

1) Insertar un departamento en la base de datos.

Leerá por teclado el nombre y la ubicación del departamento a insertar.

Se conectará a la base de datos **personal**, creada en **SQLite**, ejecutará una sentencia SQL de inserción para añadir un nuevo departamento (con código generado de forma autoincremental) en la base de datos y cerrará la conexión.

Si la inserción se ha realizado con éxito, visualizará en consola el mensaje:

Se ha insertado un departamento en la base de datos.

2) Consultar todos los departamentos de la base de datos.

Se conectará a la base de datos **personal**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar todos los departamentos de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún departamento, visualizará en consola el mensaje:

No se ha encontrado ningún departamento en la base de datos.

Si el conjunto de resultados contiene uno o más departamentos:

- Visualizará en consola un listado de todos los departamentos almacenados en la base de datos, indicando para cada departamento, un resumen con los nombres y los valores de todos los atributos de dicho departamento.
- Visualizará en consola el número de departamentos consultados de la base de datos.

Por ejemplo:

Departamento [Código = 1, Nombre = Compras, Ubicación = P1-SA]

Departamento [Código = 2, Nombre = Ventas, Ubicación = P1-SB]

Departamento [Código = 3, Nombre = Contabilidad, Ubicación = P2-SA]

Departamento [Código = 4, Nombre = Recursos Humanos, Ubicación = P2-SB]

Se han consultado 4 departamentos de la base de datos.

3) Consultar un departamento, por código, de la base de datos.

Leerá por teclado el código del departamento a consultar.

Se conectará a la base de datos **personal**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar un departamento, por código, de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún departamento, visualizará en consola el mensaje:

No existe ningún departamento con ese código en la base de datos.

Si el conjunto de resultados contiene un departamento, visualizará en consola un resumen con los nombres y los valores de todos los atributos de dicho departamento.

El siguiente ejemplo muestra el departamento cuyo código es 2:

Departamento [Código = 2, Nombre = Ventas, Ubicación = P1-SB]

4) Actualizar un departamento, por código, de la base de datos.

Leerá por teclado el código, el nuevo nombre y la nueva ubicación del departamento a actualizar.

Se conectará a la base de datos **personal**, creada en **SQLite**, ejecutará una sentencia SQL de actualización para modificar el nombre y la ubicación de un departamento, por código, de la base de datos y cerrará la conexión.

Si la actualización se ha realizado con éxito, visualizará en consola el mensaje:

Se ha actualizado un departamento de la base de datos.

Si la actualización no se ha realizado porque no se ha encontrado un departamento con el código dado, visualizará en consola el mensaje:

No existe ningún departamento con ese código en la base de datos.

5) Eliminar un departamento, por código, de la base de datos.

Leerá por teclado el código del departamento a eliminar.

Se conectará a la base de datos **personal**, creada en **SQLite**, ejecutará una sentencia SQL de eliminación para borrar un departamento, por código, de la base de datos y cerrará la conexión.

Si la eliminación se ha realizado con éxito, visualizará en consola el mensaje:

Se ha eliminado un departamento de la base de datos.

Si la eliminación no se ha realizado porque no se ha encontrado un departamento con el código dado, visualizará en consola el mensaje:

No existe ningún departamento con ese código en la base de datos.

Si la eliminación no se ha realizado porque el departamento encontrado está referenciado en uno o más empleados, visualizará en consola el mensaje:

El departamento está referenciado en un empleado de la base de datos.

Este programa principal deberá validar que la opción de menú elegida sea válida (comprendida entre 0 y 5). Si no lo es, visualizará en consola el mensaje:

La opción de menú debe estar comprendida entre 0 y 5.

ACTIVIDAD 1x02

Codifica una clase **Actividad_1x02** que incluya un programa principal **main**. Este programa gestionará la tabla **empleado** de la base de datos **personal** mediante el siguiente menú de opciones:

0) Salir del programa.

1) Insertar un empleado en la base de datos.

Leerá por teclado el nombre, la fecha de alta, el salario y el código de departamento del empleado a insertar.

Se conectará a la base de datos **personal**, creada en **SQLite**, ejecutará una sentencia SQL de inserción para añadir un nuevo empleado (con código generado de forma autoincremental) en la base de datos y cerrará la conexión.

Si la inserción se ha realizado con éxito, visualizará en consola el mensaje:

Se ha insertado un empleado en la base de datos.

2) Consultar todos los empleados de la base de datos.

Se conectará a la base de datos **personal**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar todos los empleados de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún empleado, visualizará en consola el mensaje:

No se ha encontrado ningún empleado en la base de datos.

Si el conjunto de resultados contiene uno o más empleados:

- Visualizará en consola un listado de todos los empleados almacenados en la base de datos, indicando para cada empleado, un resumen con los nombres y los valores de todos los atributos de dicho empleado.
- Visualizará en consola el número de empleados consultados de la base de datos.

Por ejemplo:

```
Empleado [Código = 1, Nombre = Juan Pardo, FechaAlta = 03/05/1986,
          Salario = 2657,44, CódigoDepartamento = 1]
Empleado [Código = 2, Nombre = Isabel Blanco, FechaAlta = 25/09/1990,
          Salario = 2133,17, CódigoDepartamento = 2]
Empleado [Código = 3, Nombre = Marta Gil, FechaAlta = 17/04/2002,
          Salario = 1987,21, CódigoDepartamento = 3]
Empleado [Código = 4, Nombre = Pablo Martínez, FechaAlta = 01/11/2004,
          Salario = 1655,98, CódigoDepartamento = 4]
Se han consultado 4 empleados de la base de datos.
```

3) Consultar un empleado, por código, de la base de datos.

Leerá por teclado el código del empleado a consultar.

Se conectará a la base de datos **personal**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar un empleado, por código, de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún empleado, visualizará en consola el mensaje:

No existe ningún empleado con ese código en la base de datos.

Si el conjunto de resultados contiene un empleado, visualizará en consola un resumen con los nombres y los valores de todos los atributos de dicho empleado.

El siguiente ejemplo muestra el empleado cuyo código es 4:

```
Empleado [Código = 4, Nombre = Pablo Martínez, FechaAlta = 01/11/2004,
          Salario = 1655,98, CódigoDepartamento = 4]
```

4) Actualizar un empleado, por código, de la base de datos.

Leerá por teclado el código, el nuevo nombre, la nueva fecha de alta, el nuevo salario y el nuevo código de departamento del empleado a actualizar.

Se conectará a la base de datos **personal**, creada en **SQLite**, ejecutará una sentencia SQL de actualización para modificar el nombre, la fecha de alta, el salario y el código de departamento de un empleado, por código, de la base de datos y cerrará la conexión.

Si la actualización se ha realizado con éxito, visualizará en consola el mensaje:

Se ha actualizado un empleado de la base de datos.

Si la actualización no se ha realizado porque no se ha encontrado un empleado con el código dado, visualizará en consola el mensaje:

No existe ningún empleado con ese código en la base de datos.

5) Eliminar un empleado, por código, de la base de datos.

Leerá por teclado el código del empleado a eliminar.

Se conectará a la base de datos **personal**, creada en **SQLite**, ejecutará una sentencia SQL de eliminación para borrar un empleado, por código, de la base de datos y cerrará la conexión.

Si la eliminación se ha realizado con éxito, visualizará en consola el mensaje:

Se ha eliminado un empleado de la base de datos.

Si la eliminación no se ha realizado porque no se ha encontrado un empleado con el código dado, visualizará en consola el mensaje:

No existe ningún empleado con ese código en la base de datos.

Este programa principal deberá validar que la opción de menú elegida sea válida (comprendida entre 0 y 5). Si no lo es, visualizará en consola el mensaje:

La opción de menú debe estar comprendida entre 0 y 5.

BASE DE DATOS biblioteca

Crea en **SQLite** un esquema de base de datos denominado **biblioteca** y compuesto por tres tablas:

1) libro:

- código: número entero, no nulo, clave primaria.
- isbn: cadena de texto, no nulo, único.
- título: cadena de texto, no nulo.
- escritor: cadena de texto, no nulo.
- año_publicación: número entero, no nulo.
- puntuación: número real, no nulo.

2) socio:

- código: número entero, no nulo, clave primaria.
- dni: cadena de texto, no nulo, único.
- nombre: cadena de texto, no nulo.
- domicilio: cadena de texto, no nulo.
- teléfono: cadena de texto, no nulo.
- correo: cadena de texto, no nulo.

3) préstamo:

- código_libro: número entero, no nulo, clave ajena de libro(código).
- código_socio: número entero, no nulo, clave ajena de socio(código).
- fecha_inicio: cadena de texto, no nulo.
- fecha_fin: cadena de texto, no nulo.
- fecha_devolución: cadena de texto.
- (código_libro, código_socio, fecha_inicio) clave primaria.

Este esquema de base de datos tiene las siguientes restricciones:

- Un libro está escrito únicamente por un escritor.
- Un libro solo se puede sacar prestado:
 - si el libro no se ha prestado a ningún socio.
 - si el libro se ha prestado a otro socio y este socio lo ha devuelto.
- Un socio solo puede sacar prestado un libro:
 - si el socio no tiene ningún libro prestado.
 - si el socio no tiene ningún otro libro prestado pendiente de devolver.
- Un socio solo puede sacar prestado un libro de la biblioteca en un día.

ACTIVIDAD 2x01

Codifica una clase **Actividad_2x01** que incluya un programa principal **main**. Este programa gestionará la tabla **libro** de la base de datos **biblioteca** mediante el siguiente menú de opciones:

0) **Salir del programa.**

1) **Insertar un libro en la base de datos.**

Leerá por teclado el ISBN, el título, el escritor, el año de publicación y la puntuación del libro a insertar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de inserción para añadir un nuevo libro (con código generado de forma autoincremental) en la base de datos y cerrará la conexión.

Si la inserción se ha realizado con éxito, visualizará en consola el mensaje:

Se ha insertado un libro en la base de datos.

2) **Eliminar un libro, por código, de la base de datos.**

Leerá por teclado el código del libro a eliminar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de eliminación para borrar un libro, por código, de la base de datos y cerrará la conexión.

Si la eliminación se ha realizado con éxito, visualizará en consola el mensaje:

Se ha eliminado un libro de la base de datos.

Si la eliminación no se ha realizado porque no se ha encontrado un libro con el código dado, visualizará en consola el mensaje:

No existe ningún libro con ese código en la base de datos.

Si la eliminación no se ha realizado porque el libro encontrado está referenciado en uno o más préstamos, visualizará en consola el mensaje:

El libro está referenciado en un préstamo de la base de datos.

3) **Consultar todos los libros de la base de datos.**

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar todos los libros de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún libro, visualizará en consola el mensaje:

No se ha encontrado ningún libro en la base de datos.

Si el conjunto de resultados contiene uno o más libros:

- Visualizará en consola un listado de todos los libros almacenados en la base de datos, indicando para cada libro, un resumen con los nombres y los valores de todos los atributos de dicho libro.
- Visualizará en consola el número de libros consultados de la base de datos.

Por ejemplo:

```

Libro [Código = 1, ISBN = 2220123456789,
      Título = The Hobbit, Escritor = J.R.R. Tolkien,
      AñoPublicación = 1937, Puntuación = 6,5]
Libro [Código = 2, ISBN = 4446789012345,
      Título = The Lord of the Rings, Escritor = J.R.R. Tolkien,
      AñoPublicación = 1955, Puntuación = 9,4]
Libro [Código = 3, ISBN = 6663456789012,
      Título = Dune, Escritor = Frank Herbert,
      AñoPublicación = 1965, Puntuación = 8,2]
Libro [Código = 4, ISBN = 8889012345678,
      Título = The Gods Themselves, Escritor = Isaac Asimov,
      AñoPublicación = 1972, Puntuación = 7,3]
Se han consultado 4 libros de la base de datos.

```

4) Consultar varios libros, por escritor, de la base de datos, ordenados por puntuación decendente.

Leerá por teclado el escritor de los libros a consultar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar los libros de un escritor, ordenados por puntuación decendente, de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún libro, visualizará en consola el mensaje:

No existe ningún libro con ese escritor en la base de datos.

Si el conjunto de resultados contiene uno o más libros:

- Visualizará en consola un listado de los libros del escritor dado almacenados en la base de datos, ordenados por puntuación decendente, indicando para cada libro, un resumen con los nombres y los valores de todos los atributos de dicho libro.
- Visualizará en consola el número de libros consultados de la base de datos.

El siguiente ejemplo muestra los libros cuyo escritor es "J.R.R. Tolkien":

```

Libro [Código = 2, ISBN = 4446789012345,
      Título = The Lord of the Rings, Escritor = J.R.R. Tolkien,
      AñoPublicación = 1955, Puntuación = 9,4]
Libro [Código = 1, ISBN = 2220123456789,
      Título = The Hobbit, Escritor = J.R.R. Tolkien,
      AñoPublicación = 1937, Puntuación = 6,5]
Se han consultado 2 libros de la base de datos.

```

5) Consultar los libros no prestados de la base de datos.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar los libros que no se han prestado a ningún socio, de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún libro, visualizará en consola el mensaje:

No existe ningún libro no prestado en la base de datos.

Si el conjunto de resultados contiene uno o más libros:

- Visualizará en consola un listado de los libros no prestados almacenados en la base de datos, indicando para cada libro, un resumen con los nombres y los valores de todos los atributos de dicho libro.
- Visualizará en consola el número de libros consultados de la base de datos.

6) Consultar los libros devueltos, en una fecha, de la base de datos.

Leerá por teclado la fecha de devolución de los préstamos a consultar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar los libros que se han devuelto en una fecha, de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún libro, visualizará en consola el mensaje:

No existe ningún libro devuelto en esa fecha en la base de datos.

Si el conjunto de resultados contiene uno o más libros:

- Visualizará en consola un listado de los libros devueltos en la fecha dada almacenados en la base de datos, indicando para cada libro, un resumen con los nombres y los valores de todos los atributos de dicho libro.
- Visualizará en consola el número de libros consultados de la base de datos.

Este programa principal deberá validar que la opción de menú elegida sea válida (comprendida entre 0 y 6). Si no lo es, visualizará en consola el mensaje:

La opción de menú debe estar comprendida entre 0 y 6.

ACTIVIDAD 2x02

Codifica una clase **Actividad_2x02** que incluya un programa principal **main**. Este programa gestionará la tabla **socio** de la base de datos **biblioteca** mediante el siguiente menú de opciones:

0) Salir del programa.**1) Insertar un socio en la base de datos.**

Leerá por teclado el DNI, el nombre, el domicilio, el teléfono y el correo del socio a insertar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de inserción para añadir un nuevo socio (con código generado de forma autoincremental) en la base de datos y cerrará la conexión.

Si la inserción se ha realizado con éxito, visualizará en consola el mensaje:

Se ha insertado un socio en la base de datos.

2) Eliminar un socio, por código, de la base de datos.

Leerá por teclado el código del socio a eliminar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de eliminación para borrar un socio, por código, de la base de datos y cerrará la conexión.

Si la eliminación se ha realizado con éxito, visualizará en consola el mensaje:

Se ha eliminado un socio de la base de datos.

Si la eliminación no se ha realizado porque no se ha encontrado un socio con el código dado, visualizará en consola el mensaje:

No existe ningún socio con ese código en la base de datos.

Si la eliminación no se ha realizado porque el socio encontrado está referenciado en uno o más préstamos, visualizará en consola el mensaje:

El socio está referenciado en un préstamo de la base de datos.

3) Consultar todos los socios de la base de datos.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar los socios que , de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún socio, visualizará en consola el mensaje:

No se ha encontrado ningún socio en la base de datos.

Si el conjunto de resultados contiene uno o más socios:

- Visualizará en consola un listado de todos los socios almacenados en la base de datos, indicando para cada socio, un resumen con los nombres y los valores de todos los atributos de dicho socio.
- Visualizará en consola el número de socios consultados de la base de datos.

Por ejemplo:

```
Socio [Código = 1, DNI = 11112222A, Nombre = Carlos Serrano,
      Domicilio = Avda. Rojo, Nº5, Calatayud, Zaragoza,
      Teléfono = 555444333, Correo = carlos@gmail.com]
Socio [Código = 2, DNI = 33334444B, Nombre = Isabel Latorre,
      Domicilio = C/ Verde, Nº2, Calatayud, Zaragoza,
      Teléfono = 666777888, Correo = isabel@hotmail.com]
Socio [Código = 3, DNI = 55556666C, Nombre = María Martínez,
      Domicilio = Avda. Azul, Nº3, Monzón, Huesca,
      Teléfono = 777999222, Correo = maria@yahoo.com]
Socio [Código = 4, DNI = 77778888D, Nombre = Jorge García,
      Domicilio = C/ Amarillo, Nº7, Alcañiz, Teruel,
      Teléfono = 888111000, Correo = jorge@gmail.com]
Se han consultado 4 socios de la base de datos.
```

4) Consultar varios socios, por localidad, de la base de datos, ordenados por nombre ascendente.

Leerá por teclado la localidad de los socios a consultar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar los socios de una localidad, ordenados por nombre ascendente, de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún socio, visualizará en consola el mensaje:

No existe ningún socio con esa localidad en la base de datos.

Si el conjunto de resultados contiene uno o más socios:

- Visualizará en consola un listado de los socios de la localidad dada almacenados en la base de datos, ordenados por nombre ascendente, indicando para cada socio, un resumen con los nombres y los valores de todos los atributos de dicho socio.
- Visualizará en consola el número de socios consultados de la base de datos.

El siguiente ejemplo muestra los socios cuya localidad es "Calatayud":

```
Socio [Código = 1, DNI = 11112222A, Nombre = Carlos Serrano,
      Domicilio = Avda. Rojo, Nº5, Calatayud, Zaragoza,
      Teléfono = 555444333, Correo = carlos@gmail.com]
Socio [Código = 2, DNI = 33334444B, Nombre = Isabel Latorre,
      Domicilio = C/ Verde, Nº2, Calatayud, Zaragoza,
      Teléfono = 666777888, Correo = isabel@hotmail.com]
Se han consultado 2 socios de la base de datos.
```

5) Consultar los socios sin préstamos de la base de datos.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar los socios que no han sacado prestado ningún libro, de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún socio, visualizará en consola el mensaje:

No existe ningún socio sin préstamos en la base de datos.

Si el conjunto de resultados contiene uno o más socios:

- Visualizará en consola un listado de los socios sin préstamos almacenados en la base de datos, indicando para cada socio, un resumen con los nombres y los valores de todos los atributos de dicho socio.
- Visualizará en consola el número de socios consultados de la base de datos.

6) Consultar los socios con préstamos en una fecha de la base de datos.

Leerá por teclado la fecha de inicio de los préstamos a consultar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar los socios que han realizado préstamos en una fecha, de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún socio, visualizará en consola el mensaje:

No existe ningún socio con préstamos en esa fecha en la base de datos.

Si el conjunto de resultados contiene uno o más socios:

- Visualizará en consola un listado de los socios con préstamos en la fecha dada almacenados en la base de datos, indicando para cada socio, un resumen con los nombres y los valores de todos los atributos de dicho socio.
- Visualizará en consola el número de socios consultados de la base de datos.

Este programa principal deberá validar que la opción de menú elegida sea válida (comprendida entre 0 y 6). Si no lo es, visualizará en consola el mensaje:

La opción de menú debe estar comprendida entre 0 y 6.

ACTIVIDAD 2x03

Codifica una clase **Actividad_2x03** que incluya un programa principal **main**. Este programa gestionará la tabla **préstamo** de la base de datos **biblioteca** mediante el siguiente menú de opciones:

0) Salir del programa.**1) Insertar un préstamo en la base de datos.**

Leerá por teclado el código de libro, el código de socio, la fecha de inicio y la fecha de fin del préstamo a insertar.

Si el código de libro hace referencia a un libro que no está disponible (es decir, se ha prestado el libro a un socio y éste aún no lo ha devuelto), visualizará en consola el mensaje:

Se ha prestado ese libro a un socio y éste aún no lo ha devuelto.

Si el código de socio hace referencia a un socio que no puede realizar un préstamo (es decir, se ha prestado un libro al socio y éste aún no lo ha devuelto), visualizará en consola el mensaje:

Se ha prestado un libro a ese socio y éste aún no lo ha devuelto.

En caso contrario (el libro está disponible y el socio puede realizar un préstamo), se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de inserción para añadir un nuevo préstamo (sin indicar una fecha de devolución) en la base de datos y cerrará la conexión.

Si la inserción se ha realizado con éxito, visualizará en consola el mensaje:

Se ha insertado un préstamo en la base de datos.

2) Actualizar un préstamo, por datos identificativos, de la base de datos.

Leerá por teclado el código de libro, el código de socio, la fecha de inicio y la nueva fecha de devolución del préstamo a actualizar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de actualización para modificar la fecha de devolución de un préstamo, por datos identificativos, de la base de datos y cerrará la conexión.

Si la actualización se ha realizado con éxito, visualizará en consola el mensaje:

Se ha actualizado un préstamo de la base de datos.

Si la actualización no se ha realizado porque no se ha encontrado un préstamo con los datos identificativos dados, visualizará en consola el mensaje:

No existe ningún préstamo con esos datos identificativos en la base de datos.

3) Eliminar un préstamo, por datos identificativos, de la base de datos.

Leerá por teclado el código de libro, el código de socio y la fecha de inicio del préstamo a eliminar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de eliminación para borrar un préstamo, por datos identificativos, de la base de datos y cerrará la conexión.

Si la eliminación se ha realizado con éxito, visualizará en consola el mensaje:

Se ha eliminado un préstamo de la base de datos.

Si la eliminación no se ha realizado porque no se ha encontrado un socio con el código dado, visualizará en consola el mensaje:

No existe ningún préstamo con esos datos identificativos en la base de datos.

4) Consultar todos los préstamos de la base de datos.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar todos los préstamos de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún préstamo, visualizará en consola el mensaje:

No se ha encontrado ningún préstamo en la base de datos.

Si el conjunto de resultados contiene uno o más préstamos:

- Visualizará en consola un listado de todos los préstamos almacenados en la base de datos, indicando para cada préstamo, un resumen con los nombres y los valores de todos los atributos de dicho préstamo.
- Visualizará en consola el número de préstamos consultados de la base de datos.

Por ejemplo:

```
Préstamo [CódigoLibro = 2, CódigoSocio = 1, FechaInicio = 10/01/2022,
          FechaFin = 24/01/2022, FechaDevolucion = 21/01/2022]
Préstamo [CódigoLibro = 3, CódigoSocio = 4, FechaInicio = 09/02/2022,
          FechaFin = 23/02/2022, FechaDevolucion = 25/02/2022]
Préstamo [CódigoLibro = 2, CódigoSocio = 4, FechaInicio = 29/03/2022,
          FechaFin = 12/04/2022, FechaDevolucion = null]
Préstamo [CódigoLibro = 3, CódigoSocio = 1, FechaInicio = 29/03/2022,
          FechaFin = 12/04/2022, FechaDevolucion = null]
Se han consultado 4 préstamos de la base de datos.
```

5) Consultar los préstamos no devueltos de la base de datos.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar los préstamos que no se han devuelto de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ningún préstamo, visualizará en consola el mensaje:

No existe ningún préstamo no devuelto en la base de datos.

Si el conjunto de resultados contiene uno o más préstamos:

- Visualizará en consola un listado de los préstamos no devueltos almacenados en la base de datos, indicando para cada préstamo, un resumen con los nombres y los valores de todos los atributos de dicho préstamo.
- Visualizará en consola el número de préstamos consultados de la base de datos.

6) Consultar DNI y nombre de socio, ISBN y título de libro y fecha de devolución de los préstamos realizados en una fecha de la base de datos.

Leerá por teclado la fecha de inicio de los préstamos a consultar.

Se conectará a la base de datos **biblioteca**, creada en **SQLite**, ejecutará una sentencia SQL de consulta para recuperar el DNI y el nombre de socio, el ISBN y el título de libro y la fecha de devolución de los préstamos realizados en una fecha de la base de datos y cerrará la conexión.

Si el conjunto de resultados no contiene ninguna línea de resumen, visualizará en consola el mensaje:

No existe ningún préstamo realizado en esa fecha en la base de datos.

Si el conjunto de resultados contiene una o más líneas de resumen:

- Visualizará en consola un listado de las líneas de resumen obtenidas de la base de datos, indicando para cada línea de resumen, los nombres y los valores de todos los atributos (DNI y nombre de socio, ISBN y título de libro, fecha de devolución) de dicha línea de resumen.
- Visualizará en consola el número de líneas de resumen obtenidas de la base de datos.

Este programa principal deberá validar que la opción de menú elegida sea válida (comprendida entre 0 y 6). Si no lo es, visualizará en consola el mensaje:

La opción de menú debe estar comprendida entre 0 y 6.

ACTIVIDAD 2x04 (ampliación)

Codifica una clase **Actividad_2x04** que incluya un programa principal **main**. Este programa gestionará varias **consultas multitabla (libro, socio y préstamo)** de la base de datos **biblioteca** mediante el siguiente menú de opciones:

- 0) Salir del programa.
- 1) Consultar el libro o los libros que ha/n sido prestado/s menos veces (y que como mínimo haya/n sido prestado/s una vez).
- 2) Consultar el socio o los socios que ha/n realizado más préstamos.
- 3) Consultar los libros que han sido prestados (incluyendo los libros no devueltos) una cantidad de veces inferior a la media.
- 4) Consultar los socios que han realizado una cantidad de préstamos superior a la media.
- 5) Consultar el ISBN, el título y el número de veces de los libros que han sido prestados, ordenados por el número de préstamos descendente.
- 6) Consultar el DNI, el nombre y el número de veces de los socios que han realizado préstamos, ordenados por el número de préstamos descendente.

Este programa principal deberá validar que la opción de menú elegida sea válida (comprendida entre 0 y 6). Si no lo es, visualizará en consola el mensaje:

La opción de menú debe estar comprendida entre 0 y 6.