

## Actividad UT2\_01: JDBC y SQLite. Base de datos de alumnos

Crear un proyecto en NetBeans, de nombre UT2\_ACT01\_SQLITE, y dentro un paquete app con una clase principal. Crear una base de datos SQLite de nombre alumnos.db en la siguiente ruta dentro del directorio del proyecto: ./src/db/alumnos.db.

Implementar una aplicación Java para acceder y gestionar los datos de la base de datos alumnos.db. Debe incluir los siguientes métodos:

- **private static Connection abreConexion (String url):** recibe una cadena de conexión a una base de datos y abre la conexión a la base de datos especificada.
- **private static void crearTabla (Connection con, String tname):** recibe una conexión y el nombre de una tabla y crea en la base de datos una tabla con ese nombre y los siguientes campos:
  - **id:** identificador del alumno, de tipo entero, clave principal.
  - **nombre:** nombre del alumno, de tipo TEXT, no puede ser NULL.
  - **ciclo:** ciclo formativo en el que está matriculado, de tipo TEXT, no puede ser NULL.
- **private static void insertarAlumno(Connection con, int id, String nombre, String ciclo, String tablename):** recibe una conexión, los datos de un alumno y el nombre de una tabla. Inserta un alumno con los datos especificados en la tabla de nombre *tablename*. Utiliza el método `executeUpdate()` de la interfaz `PreparedStatement`, y sus métodos `setXXX`, para parametrizar las sentencias SQL correspondientes.
- **private static ResultSet consultarTodosLosAlumnos(Connection con, String tablename):** recibe una conexión y el nombre de una tabla y devuelve un `ResultSet` con todos los alumnos de la tabla. Utiliza el método `executeQuery()` de la interfaz `Statement`.
- **private static void mostrarRegistrosAlumnos(ResultSet resul):** recibe un `ResultSet` con registros de la tabla alumno y muestra todos sus datos por consola con formato tabulado. Si el `ResultSet` está vacío se indica con un mensaje.
- **private static int ultimoidAlumno(Connection con, String tablename):** recibe una conexión y el nombre de una tabla y devuelve el id del último alumno de la tabla (es el último id utilizado). Utiliza el método `executeQuery` de la interfaz `Statement`.
- **private static void eliminarAlumnoPorId(Connection conexion, String tablename):** recibe una conexión y el nombre de la tabla y borra un alumno con un id determinado, que se pedirá por teclado. Utiliza el método `executeUpdate()` de la interfaz `PreparedStatement`, y sus métodos `setXXX` para parametrizar las sentencias SQL correspondientes. Si no existe el id de alumno se indicará con un mensaje.
- **private static void modificarAlumnoPorId(Connection conexion, String tablename):** recibe una conexión y el nombre de la tabla y modifica los datos de un alumno con un id determinado, que se pedirá por teclado. Se piden los nuevos datos de nombre y ciclo del alumno. Utiliza `PreparedStatement` y sus métodos `setXXX` para parametrizar las sentencias SQL correspondientes. Si no existe el id de alumno se indicará con un mensaje.

En todos los métodos se debe mostrar por consola un mensaje de éxito o error.

En el método main, escribir el código necesario para, utilizando estos métodos, hacer las siguientes operaciones:

1. Abrir la conexión a la base de datos
  2. Crear la tabla de alumnos
  3. Realizar operaciones CRUD sobre la tabla de alumnos:
    - Guardar los datos de varios alumnos en la tabla, preguntando al usuario cuántos alumnos quiere guardar, o bien repitiendo la operación hasta que el usuario elija terminar. Los id de alumno tomarán valores consecutivos según se van insertando alumnos, empezando por el 1. A cada alumno nuevo se le asigna automáticamente el valor de id inmediatamente posterior al de la última fila en la tabla.
    - Mostrar los datos de todos los alumnos de la tabla.
    - Eliminar alumnos.
    - Actualizar datos de alumnos.
- **La tabla alumnos tiene los siguientes campos:**
    - **id:** identificador del alumno, de tipo entero, clave principal.
    - **nombre:** nombre del alumno, de tipo TEXT, no puede ser NULL.
    - **ciclo:** ciclo formativo en el que está matriculado, de tipo TEXT, no puede ser NULL.