


Ejercicios Bases de Datos Relacionales

[Descargar estos ejercicios](#)

Índice

- [Ejercicio 1](#)
-  [Ejercicio 2](#)

Ejercicio 1

Siguiendo los pasos establecidos en el tema, vamos a crear una BD con **SQLite** para una tabla sencilla que contenga los datos de una Persona. El tipo **Persona** solo tendrá las propiedades públicas Id (int), Nombre (string) y Edad (int) y el método ToString. Para realizar una codificación más clara, crearemos la clase PersonaDao con las funcionalidades CRUD que vamos a realizar sobre la BD.

En este primer ejercicio solo tendremos las funciones de inserción (Insert) de una persona en la BD y la función que devolverá la lista de todas las personas (Get), por lo que nos crearemos en la clase **PersonaDao**, además del constructor al que le llega la cadena de conexión, los métodos con las siguientes signatures:

```
public void Insert(Persona persona)
public List<Persona> Get()
```

Además de la codificación de estas clases, deberemos crear en el programa principal un método privado y estático **CreaTabla** al que le llega la cadena de conexión y crea la tabla para el tipo Persona, usando los convenios establecidos en el tema, el Id de la persona será autoincrementable. En la Main deberás crear el código necesario para añadir como mínimo dos personas a la BD y realizar una consulta que nos devolverá el contenido y se visualizará mediante un ForEach sobre la consulta.

✓ Ejercicio 2

A partir de la siguiente imagen perteneciente a un formulario que se pasa a unos asistentes a unas Jornadas informáticas, se pretende recopilar la información de estas en una BD de SQLite, para poder realizar consultas sobre ella. La información deberás extraerla de la siguiente tabla:

	31 Marzo	1 Abril	2 Abril	7 Abril	9 Abril
9.00 – 9.30	BIENVENIDA. ACREDITACIÓN.				
9.30 – 10.30		Ponente:	Ponente:	Ponente:	Ponente:
		Temática:	Temática:	Temática:	Temática:
		Valoración:	Valoración:	Valoración:	Valoración:
10.30 – 11.30	Ponente:	Ponente:	Ponente:	Ponente:	Ponente:
	Temática:	Temática:	Temática:	Temática:	Temática:
	Valoración:	Valoración:	Valoración:	Valoración:	Valoración:

La clase **Ponencia**, que será de la que se creará la tabla, contendrá las propiedades FechaPonencia (DateTime habrá que convertir a string para guardar el dato en la BD), HoraInicio (float), HoraFin (float), Ponente (string), Tematica (string) y Valoracion (int que contendrá el número de personas que han valorado positivamente la ponencia).

Se deberá usar una clase **PonenciaDao** que contenga todas las funcionalidades CRUD, como se ha explicado en el tema. Las funciones que se implementarán en este caso son:

- Create. A la que le llegará una ponencia y la insertará en la tabla.
- Get. Que devolverá una lista con todas las ponencias.
- Get. Al que le llegará un id y devolverá la ponencia con ese id.
- Get. Al que le llegará un string con el nombre de la ponencia, y devolverá la ponencia que corresponde a ese ponente.
- Update. Al que le llega una ponencia y modifica con su contenido la que haya en la tabla con el mismo id.
- Update. Al que le llega un id e incrementa en uno la valoración que corresponde a la ponencia de ese id.
- Delete. Al que le llega un id y elimina la ponencia que corresponde al id.
- Dispose. Que cierra la conexión.

El programa principal tendrá los métodos necesarios para conseguir un código claro. Habrá un menú con las siguientes opciones:

Seleccione una opción:

1. Insertar una nueva ponencia
2. Ver todas las ponencias
3. Ver una ponencia por ID
4. Actualizar una ponencia
5. Añadir Like a una ponencia
6. Eliminar una ponencia
7. Ver ponencias por día
8. Ver ponencias por hora de inicio
9. Ver las dos ponencias con mas valoraciones
10. Ver ponencias que no han gustado
11. Mostrar ponente y temática de todas las ponencias
12. Salir



Tips: Añadir un Like a una ponencia, es aumentar en uno la valoración de esa ponenci. Desde la opción 7 hasta la 12, se deberán realizar mediante consultas Linq de programación funcional.