

CONTENIDOS

- Base de Datos SQLite
- Sentencias SQL
- ADO Net
- Repositorios

INTRODUCCIÓN

Copie el siguiente enlace en su navegador: <https://classroom.github.com/a/8EfETzsE> esto creará el repositorio para poder subir el **Trabajo Práctico Nro. 9**, realice los pasos ya aprendidos para *clonar* el repositorio en su máquina y poder comenzar a trabajar de forma *local*.

Considerando la base de datos Sqlite del tablero Kanban del Trabajo anterior crear una API RESTful que permita a los usuarios administrar tableros, tareas y usuarios. La aplicación debe utilizar una base de datos SQLite para almacenar la información.

1) Agregando Dependencia para utilizar SQLite con ADO.Net

Primero se debe instalar las dependencias de SQLite ejecutando el siguiente comando en la consola de administración de paquetes:

```
> dotnet add package System.Data.SQLite
```

2) Creando las clases modelo:

Crear una carpeta llamada Models donde ubicará los modelos de su aplicación. A continuación se detallan los modelos que deben crear:

Clase Usuarios:

- int Id
- NombreDeUsuario

Clase Tablero

- int Id
- int IdUsuarioPropietario
- string Nombre
- string Descripcion

Clase Tareas

- int Id
- string Nombre
- string Descripcion
- string Color
- EstadoTarea* Estado
- int? IdUsuarioAsignado

*. **EstadoTarea** es un Enum con los valores Ideas, ToDo, Doing, Review, Done

3) Creando repositorios:

Para una mejor organización y separación de responsabilidades en su API en .NET, es recomendable utilizar el Patrón de Repositorio para gestionar las operaciones de acceso a la base de datos. Cree una carpeta llamada Repositorios, donde ubicará los repositorios a construir.

A continuación se detallan los repositorios y los métodos que deben crear:

Repositorio de Usuarios:

Crear un repositorio llamado UsuarioRepository para gestionar todas las operaciones relacionadas con usuarios. Este repositorio debe incluir métodos para:

- Crear un nuevo usuario. (recibe un objeto Usuario)
- Modificar un usuario existente. (recibe un Id y un objeto Usuario)
- Listar todos los usuarios registrados. (devuelve un List de Usuarios)
- Obtener detalles de un usuario por su ID. (recibe un Id y devuelve un Usuario)
- Eliminar un usuario por ID

Repositorio de Tableros:

Crear un repositorio llamado TableroRepository para gestionar todas las operaciones relacionadas con tableros. Este repositorio debe incluir métodos para:

- Crear un nuevo tablero (devuelve un objeto Tablero)
- Modificar un tablero existente (recibe un id y un objeto Tablero)
- Obtener detalles de un tablero por su ID. (recibe un id y devuelve un Tablero)
- Listar todos los tableros existentes (devuelve un list de tableros)
- Listar todos los tableros de un usuario específico. (recibe un IdUsuario, devuelve un list de tableros)
- Eliminar un tablero por ID

Repositorio de Tareas:

Crear un repositorio llamado TareaRepository para gestionar todas las operaciones relacionadas con tareas. Este repositorio debe incluir métodos para:

- Crear una nueva tarea en un tablero. (recibe un idTablero, devuelve un objeto Tarea)
- Modificar una tarea existente. (recibe un id y un objeto Tarea)
- Obtener detalles de una tarea por su ID. (devuelve un objeto Tarea)
- Listar todas las tareas asignadas a un usuario específico. (recibe un idUsuario, devuelve un list de tareas)
- Listar todas las tareas de un tablero específico. (recibe un idTablero, devuelve un list de tareas)
- Eliminar una tarea (recibe un IdTarea)
- Asignar Usuario a Tarea (recibe idUsuario y un idTarea)

4) Creando EndPoints:

Taller de Lenguajes II – 2023

Programador Universitario / Licenciatura en informática / Ingeniería en Informática

TP Nro 9

A continuación, vamos a detallar los controladores y los distintos endpoints que se deberán agregar al sistema:

Crear un Controlador de Usuarios (UsuarioController) que incluya los endpoints para:

- **POST** /api/usuario: Permite crear un nuevo usuario.
- **GET** /api/usuario: Permite listar los usuarios existentes.
- **GET** /api/usuario/{Id}: Permite buscar un usuarios por id.
- **PUT** /api/tarea/{Id}/Nombre : Permite modificar un nombre de un Usuario.

Crear un Controlador de Tableros (TableroController) que incluya los endpoints para:

- **POST** /api/Tablero: Permite crear un tableros.
- **GET** /api/tableros: Permite listar los tableros existentes.

Crear un Controlador de Tareas (TareasController) que incluya los endpoints para:

- **POST** /api/tarea: Permite crear una Tarea.
- **PUT** /api/Tarea/{id}/Nombre/{Nombre}: Permite modificar una Tarea.
- **PUT** /api/Tarea/{id}/Estado/{estado}: Permite modificar el estado de una Tarea.
- **DELETE** /api/Tarea/{id}: Elimina una tarea por su ID.
- **GET** /api/Tarea/{Estado}: Cantidad de tareas en un estado
- **GET** /api/Tarea/Usuario/{Id}: Listar tareas asignada a un usuario
- **GET** /api/Tarea/Tablero/{Id}: Listar tareas asignada de un tablero