

## Taller de Lenguajes II – 2024

### Práctica recuperación 1er parcial

#### CONTENIDOS

- Implementación de ASP WebApi
- Separación de responsabilidades
- ADO Net
- Repositorios
- Validaciones
- Inyección de dependencias

Cree un proyecto **API** web de **ASP.NET** para manejar tareas .

1) Cree un modelo para representar una tarea.

La tarea debe tener las siguientes propiedades:

- Id: un número entero que identifica de forma única la tarea.
- Título: un título para la tarea.
- Descripción: una descripción de la tarea.
- Estado: un estado para la tarea, como "Pendiente", "En progreso" o "Completada".

2) Cree un repositorio:

Para una mejor organización y separación de responsabilidades en su API en .NET, es recomendable utilizar el Patrón Repositorio para gestionar las operaciones de acceso a la base de datos.

Cree una carpeta llamada **Repository**, donde ubicará los repositorios a construir.

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

Repositorio de Tareas:

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

- Crear una nueva tarea . (recibe un objeto Tarea)
- Obtener detalles de una tarea por su ID. (recibe un Id y devuelve una Tarea)
- Modificar una tarea existente. (recibe un Id y un objeto Tarea)
- Eliminar una tarea por ID
- Listar todas las Tareas registradas. (devuelve un List de Tarea)
- Listar todas las Tareas **completadas**. (devuelve un List de Tarea)

**Observación:** retornar valores adecuados si la tarea no existe.

### 3) Creando EndPoints:

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

Crear un Controlador de Tarea(**TareaController**) que incluya los **endpoints** para:

- **POST** /api/Tarea: Permite crear una nueva tarea
- **GET** /api/Tarea/{id}: Permite obtener una tarea por id
- **PUT** /api/Tarea/{id}: Permite actualizar una tarea.
- **GET** /api/Tarea: Permite listar las tareas existentes.
- **DELETE** /api/Tarea/{id}: Permite eliminar una tarea
- **GET** /api/Tarea: permite listar todas las tareas.
- **GET** /api/Tarea/estado/{estado}: permite listar todas las tareas por un estado en particular.

### 4) Incorpore los ViewModels

- Para crear una tarea: debe tener solo los datos necesarios para crear una tarea.
- Para modificar una tarea: debe tener los datos que me permita identificar la tarea y los nuevos datos actualizados.

### 5) Añadir las siguientes validaciones

Para asegurar la integridad de los datos y evitar la inserción de registros incompletos o incorrectos.

- Titulo (string) es obligatorio
- Descripción (string) es obligatorio, con una longitud máxima de 250 caracteres.
- Estado es obligatorio

### 6) Implemente las siguientes inyección de dependencias

- Agregue el repositorio al proyecto por inyección de dependencias
- implemente la Inyección de Cadena de conexión de la base de datos