

## CONTENIDOS

- Logueo
- Try-catch
- Autenticación

## INTRODUCCIÓN

Seguiremos trabajando con el repo del Trabajo Práctico Nro.10:

<https://classroom.github.com/a/Hj0gl71t>

En esta nueva iteración de trabajo, sobre el proyecto que venimos desarrollando vamos a agregar controles de logueo y control de excepciones, con esto seguimos haciendo un desarrollo más robusto. Además se agregará la cadena de conexión al archivo appsettings.json para que se inyecten en los repositorios.

### 1. Logueo:

1. Incluir en el endpoint de control de acceso:
  - a. Cuando el acceso fue exitoso, muestre por consola logueo de tipo info la siguiente leyenda “el usuario + UsuarioLogueado “ + ingreso correctamente
  - b. Cuando el acceso fue rechazado, muestre por consola logueo de tipo warning la siguiente leyenda “Intento de acceso invalido - Usuario:” +UsuarioLogueado+” Clava ingresada: ” + Clave
2. En los bloques Try-catch incluya logueo del tipo error, muestre por consola la serializando el error utilizando el comando toString(). Código de ejemplo en página 2.

### 2. Control de excepciones:

- a. Agregue un bloque Try-Catch en cada endpoint del sistema para garantizar que cualquier error que pudiera ocurrir durante la ejecución de estos pueda ser capturada. En caso de ocurrir una excepción, loguee el error, y envíe al usuario una página de error. Código de ejemplo en página 2.
- b. Para cada método de los repositorios del sistema, implemente un sistema para enviar excepciones en casos donde la consulta no resulta exitosa.

Ejemplo: Si se envía un id de tablero que no existe en la base de datos, debería enviar una excepción informando este error. Código de ejemplo en página 3.

Ejemplo: Si se envía un id de tablero que no existe en la base de datos, debería enviar una excepción informando este error. Código de ejemplo en página 3.

## Taller de Lenguajes II – 2023

Programador Universitario / Licenciatura en informática / Ingeniería en Informática  
TP Nro 13

### 3. Cadena de conexión:

a. Agregue en los archivos appsettings.json. la siguiente línea:

```
{  
  [...] // otras configuraciones  
  "ConnectionStrings": {  
    "SqliteConexion": "Data Source=Data/Sqlite.db;"  
  }  
}
```

b. Inyecte la conexión en los repositorios recuperando la información del archivo Json desde el objeto builder(). Para eso agregue el siguiente código al archivo program.cs:

```
var builder = WebApplication.CreateBuilder(args);  
  
// Add services to the container.  
  
builder.Services.AddControllersWithViews();  
  
var CadenaDeConexion =  
builder.Configuration.GetConnectionString("SqliteConexion")!.ToString();  
builder.Services.AddSingleton<string>(CadenaDeConexion);  
  
// Aquí se realiza la inyección de los repositorios  
// [...]  
//  
var app = builder.Build();
```

### Para Proyecto Final únicamente.

Implemente la funcionalidad de asignar usuarios a tareas. El funcionamiento debería ser el siguiente:

- c. El usuario logueado debe poder asignar un usuario a las tareas de las que es propietario.
- d. El usuario logueado debería poder ver en la lista de tableros, además de los tableros que le pertenecen, todos los tableros donde tenga tareas asignadas. Los permisos del usuario logueado para tableros que no le pertenecen son:
  - i. Tableros: Solo lectura
  - ii. Tareas no asignadas: Solo lectura.
  - iii. Tareas asignadas: Lectura y modificación, pero únicamente para modificar el estado de la tarea.

## Taller de Lenguajes II – 2023

Programador Universitario / Licenciatura en informática / Ingeniería en Informática

TP Nro 13

Ejemplo 1 - Manejo de excepciones en endpoints:

```
[HttpPost]
public IActionResult CrearTarea(int idTablero, CrearTareaViewModel
nuevaTareaVM) {
    try{
        var tareaNueva = new Tarea(nuevaTareaVM);
        tareasRepo.CrearTarea(idTablero, tareaNueva);
        return RedirectToAction("ListarTareas", new{idTablero = idTablero});
    }
    catch (Exception ex)
    {
        logger.LogError(ex.ToString());
        return BadRequest();
    }
}
```

Ejemplo 2 - Disparar una excepción en caso de que una consulta resulte vacía.

```
public Tablero GetTablero(int idTablero){
    Tablero tablero =null;
    using(var connection = new SQLiteConnection(connectionString))
    {
        connection.Open();
        string queryString = @"SELECT * FROM Tablero WHERE id = @idTablero";
        var command = new SQLiteCommand(queryString, connection);
        command.Parameters.Add(new SQLiteParameter("@idTablero", idTablero));
        using(var reader = command.ExecuteReader())
        {
            if(reader.Read()){
                tablero.Id = Convert.ToInt32(reader["id"]);
                tablero.IdUsuarioPropietario =
Convert.ToInt32(reader["id_usuario_propietario"]);
                tablero.Nombre = reader["nombre"].ToString();
                tablero.Descripcion = reader["descripcion"].ToString();
            }
        }
        connection.Close();
    }
    if (tablero==null)
        throw new Exception("Tablero no creado.");
    return tablero;
}
```