

Avance Proyecto Módulo 10

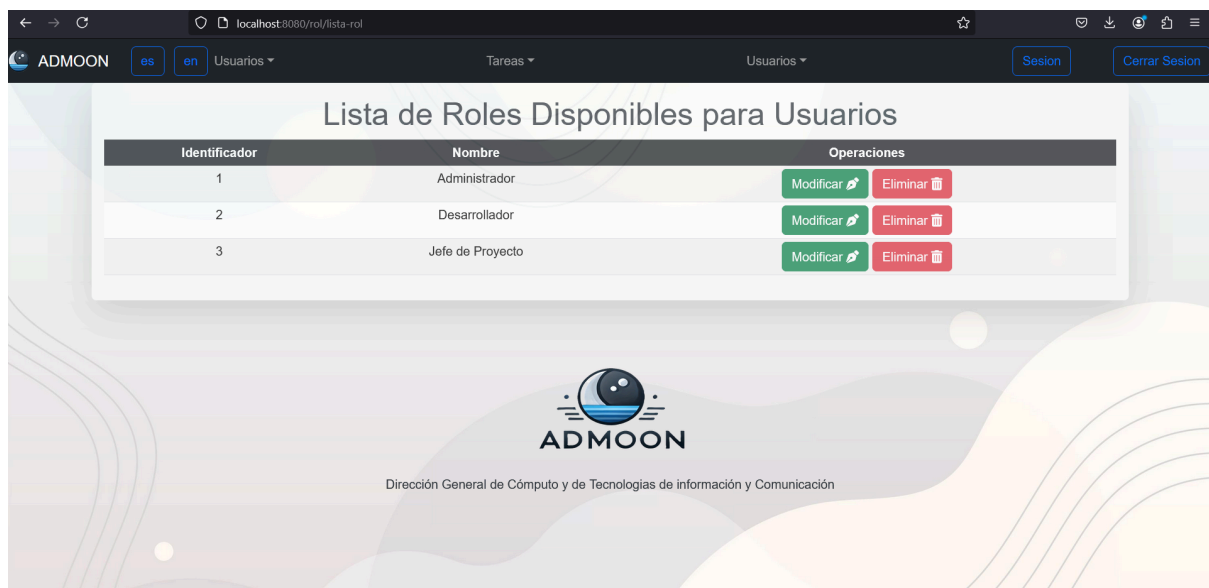
El presente proyecto tiene como objetivo el desarrollo de una API REST utilizando Java y Spring Boot, enfocada en la gestión de datos para un sistema específico. Esta API permitirá interactuar con diversas tablas del sistema, implementando operaciones CRUD (Crear, Leer, Actualizar, Eliminar) que facilitarán el manejo de información de manera eficiente y segura. En particular, se seleccionaron cuatro tablas del proyecto, de las cuales una presenta una relación con otra tabla, lo que simula un entorno de base de datos más complejo.

El sistema debe ser capaz de gestionar datos relacionados, como el ejemplo de una tabla de "autores" relacionada con una tabla de "libros", lo que constituye la *API principal* del sistema. Además, se desarrollará una interfaz de consumo que permitirá interactuar con la API a través de tecnologías modernas como Rest Template, WebClient, o frameworks de frontend, dependiendo de las necesidades del proyecto.

Este documento detallará el proceso de implementación de la API REST, describiendo las configuraciones necesarias, las operaciones CRUD implementadas, y el consumo de la API desde una interfaz externa, así como el manejo de excepciones y validaciones de campos dentro de la *API principal*. El cumplimiento de estos requisitos garantizará que el sistema sea funcional y eficiente para su propósito.

i. Capturas de pantalla del funcionamiento.

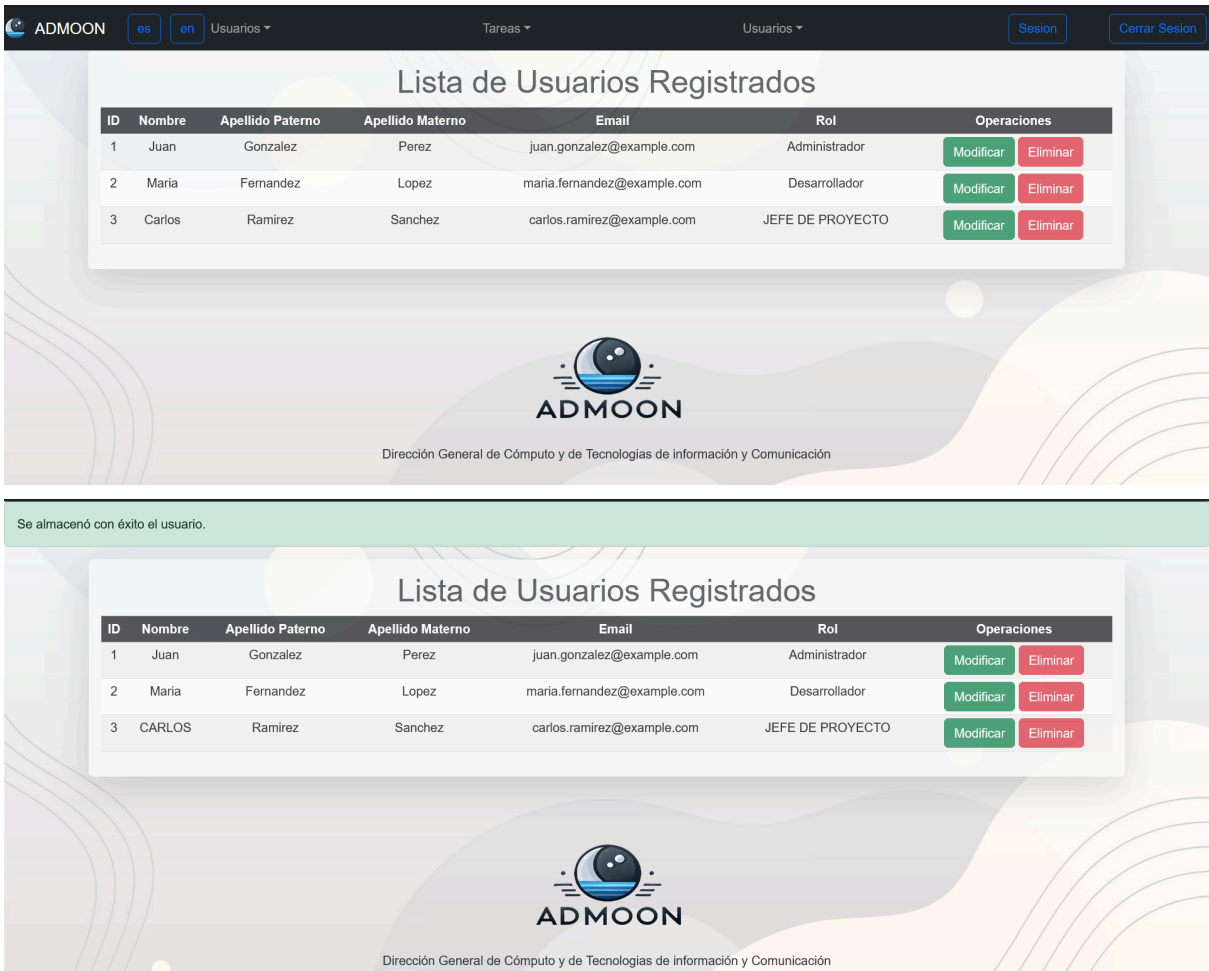
Rol.





En el caso de eliminar Rol se realizó la programación correspondiente pero no se puede realizar ya que los roles están ligados a otras clases.

Usuario.



Al igual que con roles no se puede eliminar los usuarios ya que están ligados a una tarea por ejemplo por lo que primero es necesario borrar la tarea y después el usuario pero la programación funciona correctamente.

EstadosTarea

ADMOON

es en Usuarios

Tareas

Usuarios


Sesion

Cerrar Sesion

El estado se guardó correctamente.

Lista de Estados de Tareas

ID	Nombre del Estado	Acciones
1	Pendiente	<div>EditarEliminar</div>
2	En Proceso	<div>EditarEliminar</div>
3	Completada	<div>EditarEliminar</div>

ADMOON

Dirección General de Cómputo y de Tecnologías de información y Comunicación

ADMOON

es en Usuarios

Tareas

Usuarios

Sesion

Cerrar Sesion

Lista de Estados de Tareas

ID	Nombre del Estado	Acciones
1	Pendiente	<div>EditarEliminar</div>
2	En Proceso	<div>EditarEliminar</div>
3	Completada	<div>EditarEliminar</div>

ADMOON

Dirección General de Cómputo y de Tecnologías de información y Comunicación

ADMOON

es en Usuarios

Tareas

Usuarios

Sesion

Cerrar Sesion

El estado se guardó correctamente.


Lista de Estados de Tareas

ID	Nombre del Estado	Acciones
1	Pendiente	<div>EditarEliminar</div>
2	En Proceso	<div>EditarEliminar</div>
3	Completada	<div>EditarEliminar</div>

localhost:8080

¿Está seguro de que desea eliminar el estado?

AceptarCancelar

ADMOON

Dirección General de Cómputo y de Tecnologías de información y Comunicación

ADMOON

es

en

Usuarios

Tareas

Usuarios


Sesion

Cerrar Sesion

El estado fue eliminado correctamente.

Lista de Estados de Tareas

ID	Nombre del Estado	Acciones
1	Pendiente	<div>Editar</div> <div>Eliminar</div>
2	En Proceso	<div>Editar</div> <div>Eliminar</div>

ADMOON

Dirección General de Cómputo y de Tecnologías de información y Comunicación

Tarea.

ADMOON

es

en

Usuarios

Tareas

Usuarios


Sesion

Cerrar Sesion

Lista de Tareas

ID	Nombre	Descripción	Fecha de Inicio	Fecha de Fin	Usuario Asignado	Estado	Acciones
1	Diseño de la interfaz	Diseño inicial de la interfaz gráfica	04/01/2023	19/01/2023	Juan	Pendiente	<div>Editar</div> <div>Eliminar</div>
2	Desarrollo del backend	Implementación de la lógica del servidor	20/01/2023	31/05/2023	Maria	En Proceso	<div>Editar</div> <div>Eliminar</div>
3	Pruebas unitarias	Realización de pruebas de las funciones	01/06/2023	19/06/2023	Juan	Pendiente	<div>Editar</div> <div>Eliminar</div>

Agregar Nueva Tarea

ADMOON

Dirección General de Cómputo y de Tecnologías de información y Comunicación

ADMOON

es

en

Usuarios

Tareas

Usuarios

Sesion

Cerrar Sesion

Lista de Tareas


ID	Nombre	Descripción	Fecha de Inicio	Fecha de Fin	Usuario Asignado	Estado	Acciones
1	Diseño de la interfaz	Diseño inicial de la interfaz gráfica	04/01/2023	19/01/2023	Juan	Pendiente	<div>Editar</div> <div>Eliminar</div>
2	Desarrollo del backend	Implementación de la lógica del servidor	20/01/2023	31/05/2023	Maria	En Proceso	<div>Editar</div> <div>Eliminar</div>
3	Pruebas unitarias	Realización de pruebas de			Juan	Pendiente	<div>Editar</div> <div>Eliminar</div>

localhost:8080

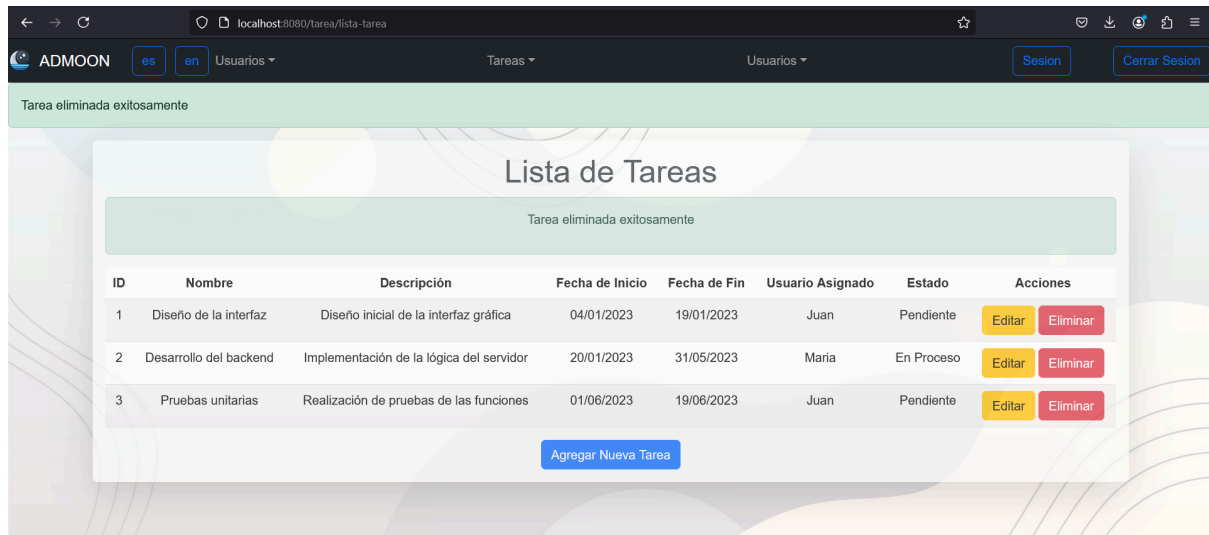
¿Estás seguro de que deseas eliminar esta tarea?

Aceptar

Cancelar

ADMOON

Dirección General de Cómputo y de Tecnologías de información y Comunicación



Configuraciones necesarias para la aplicación de consumo

Dependencias necesarias

Para consumir un API RESTful utilizando WebClient, es necesario incluir la dependencia de Spring WebFlux en el archivo pom.xml. Esto garantiza el acceso al cliente WebClient y otras herramientas reactivas de Spring Boot. La dependencia es la siguiente:

```
<dependency>
```

```
    <groupId>org.springframework.boot</groupId>
```

```
    <artifactId>spring-boot-starter-webflux</artifactId>
```

```
</dependency>
```

Configuración del WebClient

Para centralizar el acceso al API, configuré un bean reutilizable. Esto facilita la gestión de las solicitudes, ya que todas las operaciones tendrán como base la URL especificada. La configuración es la siguiente:

```
@Configuration
```

```
public class WebClientConfig {
```

```
    @Bean
```

```
    public WebClient webClient(WebClient.Builder builder) {
```

```
        return builder.baseUrl("http://localhost:8080/api")
```

```
        .build();  
    }  
}
```

Manejo de errores

Implementé un manejador de errores para capturar y registrar cualquier excepción que pueda ocurrir durante el consumo del API. Esto permite identificar problemas de forma más clara y estructurada:

@Component

```
public class WebClientErrorHandler {  
    public void handleError(Throwable throwable) {  
        if (throwable instanceof WebClientResponseException) {  
            WebClientResponseException ex = (WebClientResponseException) throwable;  
            System.err.println("Código de error: " + ex.getStatusCode());  
            System.err.println("Cuerpo del error: " + ex.getResponseBodyAsString());  
        } else {  
            System.err.println("Error desconocido: " + throwable.getMessage());  
        }  
    }  
}
```

Cómo acceder a la sección del consumo del RESTful

Implementación de métodos de consumo

Para consumir los endpoints del API, definí un servicio llamado RolService. Este servicio utiliza WebClient para realizar las operaciones CRUD. Por ejemplo, el método para eliminar un rol se implementó de la siguiente manera:

```
public Mono<Void> deleteRol(Long id) {  
    return webClient.delete()  
        .uri("/rol/{id}", id)
```

```
        .retrieve()

        .bodyToMono(Void.class)

        .doOnError(e -> System.err.println("Error al eliminar el Rol: " +
e.getMessage()));
    }
}
```

Controlador para probar el consumo

Para exponer los métodos del servicio y probar el consumo del API, creé un controlador que interactúa con RolService. A continuación, se muestra el código para eliminar un rol:

```
@RestController

@RequestMapping("/consumo")

public class RolController {

    private final RolService rolService;

    public RolController(RolService rolService) {

        this.rolService = rolService;
    }

    @DeleteMapping("/rol/{id}")

    public Mono<Void> eliminarRol(@PathVariable Long id) {

        return rolService.deleteRol(id);
    }

}
}
```

Pruebas locales

Antes de integrar las operaciones en la lógica principal del sistema, realicé pruebas utilizando herramientas como Postman y cURL para verificar el correcto funcionamiento de los endpoints:

Postman: Configuré una solicitud DELETE hacia `http://localhost:8080/api/rol/{id}`.

Con estas configuraciones y pruebas, validé que el consumo del API funciona correctamente y que los errores son manejados de forma adecuada