



Avance de Proyecto Módulo 10

Introducción

El presente documento describe el funcionamiento y configuraciones necesarias de la API RESTful desarrollada para la gestión de usuarios en un sistema web. El proyecto implementa validaciones y manejo de excepciones según los requerimientos establecidos, y se proporciona una guía para su correcta ejecución y consumo.

Capturas de Pantalla del Funcionamiento

La aplicación en funcionamiento.

A screenshot of an IDE (IntelliJ IDEA) showing the project structure, pom.xml file, and application logs. The project structure on the left includes files like formNuevoUsuario.html, galeria.html, inicio.html, layout.html, principal.html, usuarioDetalle.html, and usuarios.html. The pom.xml file in the center lists dependencies for Spring Data, Spring Boot, and Spring Web. The application logs at the bottom show the startup sequence, including Tomcat initialization, Spring Data repository scanning, and the application starting on port 8080.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.mariadb.jdbc</groupId>
  <artifactId>mariadb-java-client</artifactId>
  <scope>runtime</scope>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.doc</groupId>
    <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
    <version>2.6.0</version>
  </dependency>
</project>
```

```
2024-11-21T23:17:21.604-06:00 INFO 6348 --- [My Art List] [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 213 ms. Found 6 JPA repository interfaces.
2024-11-21T23:17:22.522-06:00 INFO 6348 --- [My Art List] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-11-21T23:17:22.540-06:00 INFO 6348 --- [My Art List] [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-11-21T23:17:22.540-06:00 INFO 6348 --- [My Art List] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.30]
2024-11-21T23:17:22.625-06:00 INFO 6348 --- [My Art List] [ restartedMain] o.a.c.c.f.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-11-21T23:17:22.625-06:00 INFO 6348 --- [My Art List] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2856 ms
2024-11-21T23:17:22.760-06:00 INFO 6348 --- [My Art List] [ restartedMain] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Starting...
2024-11-21T23:17:22.889-06:00 INFO 6348 --- [My Art List] [ restartedMain] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection org.mariadb.jdbc.Connection@6eac8bd2
2024-11-21T23:17:23.003-06:00 INFO 6348 --- [My Art List] [ restartedMain] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Start completed.
2024-11-21T23:17:23.076-06:00 INFO 6348 --- [My Art List] [ restartedMain] org.hibernate.Version : HHH000204: Processing PersistenceUnitInfo [name: default]
2024-11-21T23:17:23.119-06:00 INFO 6348 --- [My Art List] [ restartedMain] o.h.c.internal.RegionFactoryInitiator : HHH000412: Hibernate ORM core version 6.5.3.Final
2024-11-21T23:17:23.496-06:00 INFO 6348 --- [My Art List] [ restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: Ignoring JPA class transformer
2024-11-21T23:17:24.897-06:00 INFO 6348 --- [My Art List] [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2024-11-21T23:17:25.153-06:00 INFO 6348 --- [My Art List] [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2024-11-21T23:17:26.192-06:00 WARN 6348 --- [My Art List] [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure open-in-view to avoid potential performance issues.
2024-11-21T23:17:27.067-06:00 INFO 6348 --- [My Art List] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2024-11-21T23:17:27.113-06:00 INFO 6348 --- [My Art List] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
2024-11-21T23:17:27.127-06:00 INFO 6348 --- [My Art List] [ restartedMain] dgic.core.SpringWebApplication : Started SpringWebApplication in 7.907 seconds (process running for 8.567)
```

Ejemplos de solicitudes y respuestas a la API.

Tabla Usuarios – Get All

M10 Proyecto

GET Get All data

GET Get All data Copy

GET Get One

GET Get Usuario By Genero

GET Get Usuario By Genero Copy

POST Post data

PUT Update data

DEL Delete data

PATCH Patch

M10E01 Alumnos

M10E01 AlumnosDTO

M10E01 EstadosDTO

modulo10_2024

modulo10_participacion1

GET

http://localhost:8080/api/usuarios/

Params

Authorization

Headers (8)

Body

Scripts

Settings

Query Params

Key

Value

Body

Cookies (1)

Headers (5)

Test Results (1/1)

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

[

{

"id": 1,

"correo": "juan.perez@example.com",

"nombre": "Juan",

"contrasena": "hashed_password_1",

"direccion": "Calle Falsa 123, Ciudad",

"telefono": "555-1234",

"genero": "Femenino",

"carreraProfesional": "Diseño Gráfico"

}

,

{

"id": 2,

"correo": "maria.lopez@example.com",

"nombre": "Maria",

"contrasena": "hashed_password_2",

"direccion": "Avenida Siempreviva 742, Ciudad",

"telefono": "555-5678",

"genero": "Femenino",

"carreraProfesional": "Arquitectura"

}

,

{

"id": 3,

"correo": "carlos.garcia@example.com",

"nombre": "Carlos",

"contrasena": "hashed_password_3",

"direccion": "Boulevard de los Sueños Rotos, Ciudad",

"telefono": "555-9012",

"genero": "Masculino",

"carreraProfesional": "Ingenieria"

}

]

GetBy ID

GET Get All data

GET Get All data Copy

GET Get One

GET Get Usuario By Genero

GET Get Usuario By Genero Copy

POST Post data

PUT Update data

DEL Delete data

PATCH Patch

M10E01 Alumnos

M10E01 AlumnosDTO

M10E01 EstadosDTO

modulo10_2024

modulo10_participacion1

GET

http://localhost:8080/api/usuarios/1

Params

Authorization

Headers (8)

Body

Scripts

Set

Query Params

Key

Body

Cookies (1)

Headers (5)

Test Results (1/1)

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

{

"id": 1,

"correo": "juan.perez@example.com",

"nombre": "Juan",

"contrasena": "hashed_password_1",

"direccion": "Calle Falsa 123, Ciudad",

"telefono": "555-1234",

"genero": "Impresionismo",

"carreraProfesional": "Diseño Gráfico"

}

The screenshot displays a REST client interface with a sidebar on the left listing API endpoints under the 'M10 Proyecto' folder. The selected endpoint is 'GET Get Usuario By Genero'. The main panel shows a GET request to 'http://localhost:8080/api/usuarios/genero/Femenino'. The response is displayed in the 'Body' tab, showing a JSON array of three user objects. The 'Pretty' view is selected, and the response is formatted for readability.

API Endpoints:

- GET Get All data
- GET Get All data Copy
- GET Get One
- GET Get Usuario By Genero
- GET Get Usuario By Genero Copy
- POST Post data
- PUT Update data
- DEL Delete data
- PATCH Patch

Request Details:

- Method: GET
- URL: http://localhost:8080/api/usuarios/genero/Femenino
- Params: none
- Authorization: none
- Headers: none
- Body: raw

Response (JSON):

```
[
  {
    "id": 1,
    "correo": "juan.perez@example.com",
    "nombre": "Juan",
    "contrasena": "hashed_password_1",
    "direccion": "Calle Falsa 123, Ciudad",
    "telefono": "555-1234",
    "genero": "Femenino",
    "carreraProfesional": "Diseño Gráfico"
  },
  {
    "id": 2,
    "correo": "maria.lopez@example.com",
    "nombre": "María",
    "contrasena": "hashed_password_2",
    "direccion": "Avenida Siempreviva 742, Ciudad",
    "telefono": "555-5678",
    "genero": "Femenino",
    "carreraProfesional": "Arquitectura"
  },
  {
    "id": 4,
    "correo": "dasw@dadsa.com",
    "nombre": "Alans",
    "contrasena": "fregerg",
    "direccion": "asdavbrrrs",
    "telefono": "5580022382",
    "genero": "Femenino",
    "carreraProfesional": "Arquitectura"
  }
]
```

GetByCarreraProfesional(Otra tabla diferente)

GET

Get All data

GET

Get All data Copy

GET

Get One

GET

Get Usuario By Genero

GET

Get Usuario By Genero Copy

POST

Post data

PUT

Update data

DEL

Delete data

PATCH

Patch

>

M10E01 Alumnos

>

M10E01 AlumnosDTO

>

M10E01 EstadosDTO

>

modulo10_2024

>

modulo10_participacion1

GET

▼

http://localhost:8080/api/usuarios/carreraP/Arquitectura

Params

Authorization

Headers (8)

Body

Scripts ●

Settings

Body

Cookies (1)

Headers (5)

Test Results (1/1)

Pretty

Raw

Preview

Visualize

JSON ▼

↻

1

[

2

{

3

"id": 2,

4

"correo": "maria.lopez@example.com",

5

"nombre": "María",

6

"contrasena": "hashed_password_2",

7

"direccion": "Avenida Siempreviva 742, Ciudad",

8

"telefono": "555-5678",

9

"genero": "Femenino",

10

"carreraProfesional": "Arquitectura"

11

},

12

{

13

"id": 4,

14

"correo": "dasw@dadsa.com",

15

"nombre": "Alans",

16

"contrasena": "fregerg",

17

"direccion": "asdavbrrrrs",

18

"telefono": "5580022382",

19

"genero": "Femenino",

20

"carreraProfesional": "Arquitectura"

21

},

22

{

23

"id": 6,

24

"correo": "Alfredo.Gonzalo@Alfredo.com",

25

"nombre": "Alfredo",

26

"contrasena": "Contrasena12",

27

"direccion": "Calle Falsa 23, Ciudad",

28

"telefono": "5555511111",

29

"genero": "Femenino",

30

"carreraProfesional": "Arquitectura"

31

}

32

]

POST

M10 Proyecto

- GET Get All data
- GET GetAllCarrerasP
- GET Get One
- GET Get Usuario By Genero
- GET Get Usuario By Genero Copy
- POST Post data
- PUT Update data
- DEL Delete data
- PATCH Patch

M10E01 Alumnos

M10E01 AlumnosDTO

M10E01 EstadosDTO

modulo10_2024

modulo10_participacion1

POSThttp://localhost:8080/api/usuarios/

ParamsAuthorizationHeaders (10)BodyScriptsSettings

noneform-datax-www-form-urlencodedorawbinaryGraphQLJSON

```
1 {
2   "id": 5,
3   "correo": "Kally.perez@example.com",
4   "nombre": "Kally",
5   "contrasena": "hashed_password_33",
6   "direccion": "Calle Falsa 23, Ciudad",
7   "telefono": "5555511111",
8   "genero": "Femenino",
9   "carreraProfesional": "Diseño Gráfico"
10 }
```

BodyCookies (1)Headers (6)Test Results (1/1)201 Created168

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 5,
3   "correo": "Kally.perez@example.com",
4   "nombre": "Kally",
5   "contrasena": "hashed_password_33",
6   "direccion": "Calle Falsa 23, Ciudad",
7   "telefono": "5555511111",
8   "genero": "Femenino",
9   "carreraProfesional": "Diseño Gráfico"
10 }
```

PUT

M10 Proyecto

- GET Get All data
- GET GetAllCarrerasP
- GET Get One
- GET Get Usuario By Genero
- GET Get Usuario By Genero Copy
- POST Post data
- PUT Update data
- DEL Delete data
- PATCH Patch

M10E01 Alumnos

M10E01 AlumnosDTO

M10E01 EstadosDTO

modulo10_2024

modulo10_participacion1

PUThttp://localhost:8080/api/usuarios/7

ParamsAuthorizationHeaders (10)BodyScripts

noneform-datax-www-form-urlencodedoraw

```
1 {
2   "id": 7,
3   "correo": "Ana.perez@example.com",
4   "nombre": "Ana",
5   "contrasena": "hashed_password_33",
6   "direccion": "Calle Falsa 23, Ciudad",
7   "telefono": "5555511111",
8   "genero": "Femenino",
9   "carreraProfesional": "Diseño Gráfico"
10 }
```

BodyCookies (1)Headers (5)Test Results (1/1)

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 7,
3   "correo": "Ana.perez@example.com",
4   "nombre": "Ana",
5   "contrasena": "hashed_password_33",
6   "direccion": "Calle Falsa 23, Ciudad",
7   "telefono": "5555511111",
8   "genero": "Femenino",
9   "carreraProfesional": "Diseño Gráfico"
10 }
```

Delete

M10 Proyecto

- GET Get All data
- GET GetAllCarrerasP
- GET Get One
- GET Get Usuario By Genero
- GET Get Usuario By Genero Copy
- POST Post data
- PUT Update data
- DEL Delete data
- PATCH Patch

M10E01 Alumnos

M10E01 AlumnosDTO

M10E01 EstadosDTO

modulo10_2024

modulo10_participacion1

DELETEhttp://localhost:8080/api/usuarios/7

ParamsAuthorizationHeaders (8)BodyScriptsSettings

Query Params

	Key	Value	Description
	Key	Value	Description

BodyCookies (1)Headers (4)Test Results (1/1)

200 OK1

PrettyRawPreviewVisualizeText

1

Patch

M10 Proyecto

- GET Get All data
- GET GetAllCarrerasP
- GET Get One
- GET Get Usuario By Genero
- GET Get Usuario By Genero Copy
- POST Post data
- PUT Update data
- DEL Delete data
- PATCH Patch

M10E01 Alumnos

M10E01 AlumnosDTO

M10E01 EstadosDTO

modulo10_2024

modulo10_participacion1

PATCHhttp://localhost:8080/api/usuarios/6

ParamsAuthorizationHeaders (10)BodyScriptsSettings

noneform-datax-www-form-urlencodedorawbinary

1{

2"correo": "fredo.Gonzalo@Alfredo.com",

3"nombre": "Aldo",

4"contrasena": "Contrasena12",

5"carreraProfesional": "Arquitectura"

6}

BodyCookies (1)Headers (5)Test Results

PrettyRawPreviewVisualizeJSON

1{

2"id": 6,

3"correo": "fredo.Gonzalo@Alfredo.com",

4"nombre": "Aldo",

5"contrasena": "Contrasena12",

6"direccion": "Calle Falsa 23, Ciudad",

7"telefono": "5555511111",

8"genero": "Femenino",

9"carreraProfesional": "Arquitectura"

10}

Tabla Pintura - Post

Collections

environments

History

+

...

M10 Proyecto

GET Get All data

GET GetAllCarrerasP

GET Get One

GET Get Usuario By Genero

GET Get Usuario By Genero Copy

POST Post data

POST PostPintura

PUT Update data

DEL Delete data

PATCH Patch

PATCH Patch Artista

M10E01 Alumnos

M10E01 AlumnosDTO

M10E01 EstadosDTO

modulo10_2024

modulo10_participacion1

M10 Proyecto / PostPintura

POST http://localhost:8080/api/pinturas/

Params Authorization Headers (10) Body Scripts Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1 {

2 "id": 10,

3 "tituloObra": "Hombre blanco",

4 "anioCreacion": "1989-06-01",

5 "tecnica": "Óleo sobre lienzo",

6 "detalles": "Una de las obras más famosas de Botero...",

7 "idArtista": 2,

8 "nombreArtista": "Botero"

9 }

Body

Cookies (1)

Headers (6)

Test Results (1/1)

201 Created 45 ms

Pretty

Raw

Preview

Visualize

JSON

1 [Body]

2 "id": 10,

3 "tituloObra": "Hombre blanco",

4 "anioCreacion": "1989-06-01",

5 "tecnica": "Óleo sobre lienzo",

6 "detalles": "Una de las obras más famosas de Botero...",

7 "idArtista": 2,

8 "nombreArtista": "Leonardo da Vinci"

9 [Body]

Tabla Artista - Put

...

✓ M10 Proyecto

GET Get All data

GET GetAllCarrerasP

GET Get One

GET Get Usuario By Genero

GET Get Usuario By Genero Copy

POST Post data

POST PostPintura

PUT Update data

DEL Delete data

PATCH Patch

PUT Put Artista

> M10E01 Alumnos

> M10E01 AlumnosDTO

> M10E01 EstadosDTO

> modulo10_2024

> modulo10_participacion1

HTTP

M10 Proyecto / Put Artista

PUT

▼

http://localhost:8080/api/artistas/7

Params

Authorization

Headers (10)

Body ●

Scripts

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

1 {

2 |

3 | "id": 7,

4 | "nombre": "Botero",

5 | "nacionalidad": "Colombiano"

5 }

Body

Cookies (1)

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON ▼

⋮

1 {

2 |

3 | "id": 7,

4 | "nombre": "Botero",

4 | "nacionalidad": "Colombiano"

5 |

5 }

Manejo de excepciones y respuestas de error

En la capa de servicio, se lanzan excepciones personalizadas si, no se encuentra un usuario al intentar actualizarlo.

```
}

@ExceptionHandler(UsuarioNoEncontradoException.class) no usages
public ResponseEntity<DetalleError> manejoUsuarioNoEncontrado(UsuarioNoEncontradoException ex) {
    DetalleError detalleError = new DetalleError();
    detalleError.setMensaje(ex.getMessage());
    detalleError.setStatusCode(HttpStatus.NOT_FOUND.toString());
    detalleError.setTimeStamp(LocalDateTime.now());
    detalleError.setDetalle("El usuario solicitado no existe");

    return ResponseEntity.status(HttpStatus.NOT_FOUND).body(detalleError);
}
}
```

```
package dgtic.core.exception;

public class UsuarioNoEncontradoException extends RuntimeException { 4 usages

    public UsuarioNoEncontradoException(String message) { 1 usage
        super(message);
    }
}
```

```
@Override 2 usages
public UsuarioDto updateUsuario(UsuarioDto usuarioDto) throws ParseException {
    Optional<Usuario> usuarioOpt = usuarioRepository.findById(String.valueOf(usuarioDto.getId()));
    if (usuarioOpt.isPresent()) {
        Usuario usuarioActualizado = usuarioRepository.save(this.convertToEntity(usuarioDto));
        return this.convertToDto(usuarioActualizado);
    } else {
        throw new UsuarioNoEncontradoException("Usuario con ID " + usuarioDto.getId() + " no encontrado");
    }
}
```

```
List of constraint violations:[
  ConstraintViolationImpl{interpolatedMessage='El género es obligatorio', propertyPath=genero, rootBeanClass=class dgtic.core.model.entities.Usuario, messageTemplate='El género es obligatorio'}
]
```

Visualizacion con Thymeleaf.

My Art List

InicioUsuariosGalería

Bienvenido a My Art List

Aquí se puede gestionar usuarios y explorar una galería de pinturas famosas.

© 2024 My Art List. Todos los derechos reservados.

My Art List

InicioUsuariosGalería

Id	Nombre	Correo	Direccion	Telefono	Genero	Carrera Profesional	Acciones
1	Juan	juan.perez@example.com	Calle Falsa 123, Ciudad	555-1234	Femenino	Diseño Gráfico	<div>Ver</div> <div>Editar</div>
2	Maria	maria.lopez@example.com	Avenida Siempreviva 742, Ciudad	555-5678	Femenino	Arquitectura	<div>Ver</div> <div>Editar</div>
3	Carlos	carlos.garcia@example.com	Boulevard de los Sueños Rotos, Ciudad	555-9012	Masculino	Ingeniería	<div>Ver</div> <div>Editar</div>
4	Alans	dasw@dadsa.com	asdavbbrms	5580022382	Femenino	Arquitectura	<div>Ver</div> <div>Editar</div>
5	Kally	Kally.perez@example.com	Calle Falsa 23, Ciudad	5555511111	Femenino	Diseño Gráfico	<div>Ver</div> <div>Editar</div>
6	Aldo	fredo.gonzalo@alfredo.com	Calle Falsa 23, Ciudad	5555511111	Femenino	Arquitectura	<div>Ver</div> <div>Editar</div>
9	Alan	alan@hotmail.com	Cerrada	5580066123	Femenino	Diseño Gráfico	<div>Ver</div> <div>Editar</div>
10	Mauris	Alans@correo.com	Cerrada	5580066123	Femenino	Diseño Gráfico	<div>Ver</div> <div>Editar</div>

Crear Nuevo Usuario

© 2024 My Art List. Todos los derechos reservados.

My Art List

InicioUsuariosGalería

Juan

juan.perez@example.com

Detalles:

Id:1

Direccion:Calle Falsa 123, Ciudad

Genero:Femenino

Carrera Profesional:Diseño Gráfico

Todos los usuarios

© 2024 My Art List. Todos los derechos reservados.

My Art List

InicioUsuariosGalería

Crear Nuevo Usuario

Nombre

Contraseña

Telefono

Carrera Profesional

Correo

Direccion

Genero

Crear Usuario

Cancelar

© 2024 My Art List. Todos los derechos reservados.

My Art List

InicioUsuariosGalería

Id

Nombre

direccion

Carrera Profesional

contrasena

correo

genero

telefono

Guardar Cambios

1

Juan

Calle Falsa 123, Ciudad

Diseño Gráfico

hashed_password_1

juan.perez@example.com

Femenino

555-1234

© 2024 My Art List. Todos los derechos reservados.



Peter_Paul_Rubens_-_The_miracles_of_St._Francis_Xavier

Una pintura de Peter_Paul_Rubens.



immaculate conception

Obra de Peter_Paul_Rubens.

Indicaciones Especiales

Cómo Acceder a la Sección del Consumo del API RESTful

URL Base de la API:

La API RESTful está disponible en <http://localhost:8080/api/usuarios>.

Endpoints Disponibles:

URL Base: `/api/usuarios/`

- **GET /api/usuarios/**
 - **Descripción:** Obtiene una lista de todos los usuarios registrados en el sistema.
- **GET /api/usuarios/{id}**
 - **Descripción:** Obtiene los detalles de un usuario específico basado en su ID.
- **POST /api/usuarios/**
 - **Descripción:** Crea un nuevo usuario con la información proporcionada en el cuerpo de la solicitud.
- **PUT /api/usuarios/{id}**
 - **Descripción:** Actualiza completamente la información de un usuario existente identificado por su ID.
- **PATCH /api/usuarios/{id}**
 - **Descripción:** Actualiza parcialmente uno o más campos de un usuario existente.
- **DELETE /api/usuarios/{id}**
 - **Descripción:** Elimina el usuario identificado por su ID del sistema.
- **GET /api/usuarios/genero/{genero}**
 - **Descripción:** Obtiene una lista de usuarios filtrados por género (por ejemplo, Masculino, Femenino).
- **GET /api/usuarios/carrera/{carreraProfesional}**
 - **Descripción:** Obtiene una lista de usuarios que pertenecen a una carrera profesional específica.

Endpoints para Carreras Profesionales

URL Base: `/api/carrerasprofesionales/`

- **GET /api/carrerasprofesionales/**
 - **Descripción:** Obtiene una lista de todas las carreras profesionales disponibles.
- **GET /api/carrerasprofesionales/{id}**
 - **Descripción:** Obtiene los detalles de una carrera profesional específica basada en su ID.

- **POST /api/carrerasprofesionales/**
 - **Descripción:** Crea una nueva carrera profesional con la información proporcionada.
- **PUT /api/carrerasprofesionales/{id}**
 - **Descripción:** Actualiza completamente la información de una carrera profesional existente.
- **PATCH /api/carrerasprofesionales/{id}**
 - **Descripción:** Actualiza parcialmente uno o más campos de una carrera profesional.
- **DELETE /api/carrerasprofesionales/{id}**
 - **Descripción:** Elimina la carrera profesional identificada por su ID.

Endpoints para Pinturas

URL Base: /api/pinturas/

- **GET /api/pinturas/**
 - **Descripción:** Obtiene una lista de todas las pinturas registradas.
- **GET /api/pinturas/{id}**
 - **Descripción:** Obtiene los detalles de una pintura específica basada en su ID.
- **POST /api/pinturas/**
 - **Descripción:** Crea una nueva pintura con la información proporcionada en el cuerpo de la solicitud.
- **PUT /api/pinturas/{id}**
 - **Descripción:** Actualiza completamente la información de una pintura existente.
- **PATCH /api/pinturas/{id}**
 - **Descripción:** Actualiza parcialmente los datos de una pintura.
- **DELETE /api/pinturas/{id}**
 - **Descripción:** Elimina la pintura identificada por su ID.
- **GET /api/pinturas/artista/{idArtista}**
 - **Descripción:** Obtiene una lista de pinturas realizadas por un artista específico.

Endpoints para Artistas

URL Base: /api/artistas/

- **GET /api/artistas/**
 - **Descripción:** Obtiene una lista de todos los artistas registrados.
- **GET /api/artistas/{id}**
 - **Descripción:** Obtiene los detalles de un artista específico basado en su ID.

- **POST /api/artistas/**
 - **Descripción:** Crea un nuevo artista con la información proporcionada.
- **PUT /api/artistas/{id}**
 - **Descripción:** Actualiza completamente la información de un artista existente.
- **PATCH /api/artistas/{id}**
 - **Descripción:** Actualiza parcialmente uno o más campos de un artista.
- **DELETE /api/artistas/{id}**
 - **Descripción:** Elimina el artista identificado por su ID.
- **GET /api/artistas/nacionalidad/{nacionalidad}**
 - **Descripción:** Obtiene una lista de artistas filtrados por nacionalidad.

Ejemplo de Solicitud para Crear un Usuario:

POST http://localhost:8080/api/usuarios/

Content-Type: application/json"

Body:{

```
"correo": "nuevo_usuario@example.com",  
"nombre": "Nuevo Usuario",  
"contrasena": "password123",  
"direccion": "Calle Falsa 123",  
"telefono": "5551234567",  
"genero": "Masculino",  
"carreraProfesional": "Ingeniería de Software"  
}'
```