

Módulo 9: Avance Proyecto

Jaramillo Hernández José Guadalupe

Repositorio GitHub público:

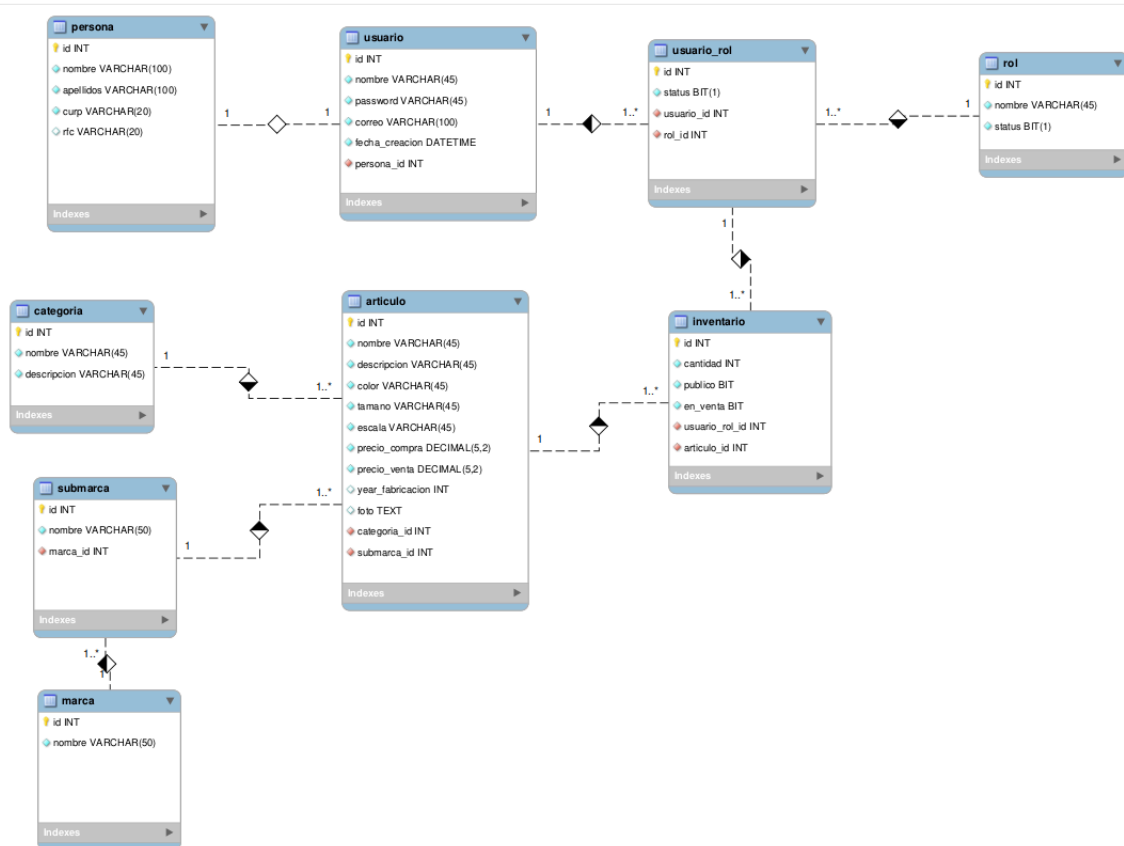
SSH:

[git@github.com:JJARAMILLOHDZ/Diplomado.git](https://github.com/JJARAMILLOHDZ/Diplomado.git)

HTTPS:

<https://github.com/JJARAMILLOHDZ/Diplomado.git>

Diagrama de la base de datos:



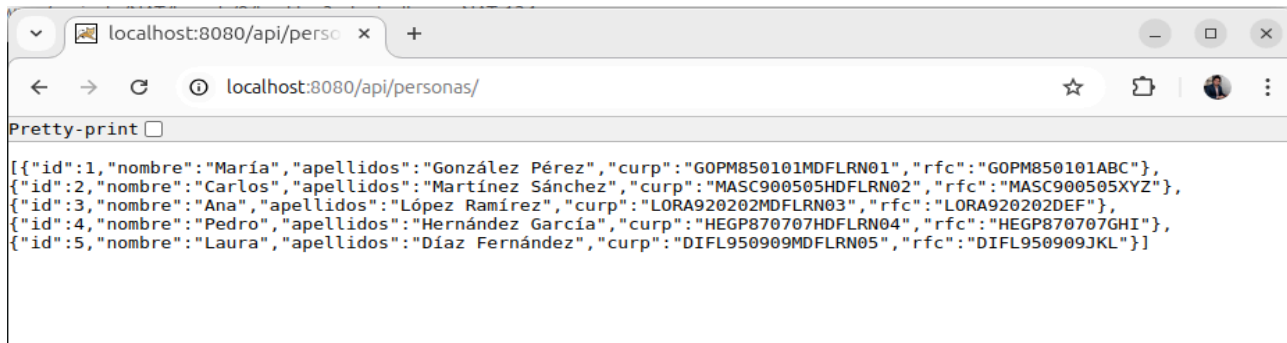
Existen 2 archivos sql en la carpeta resources

Para crear la base de datos: schema.sql

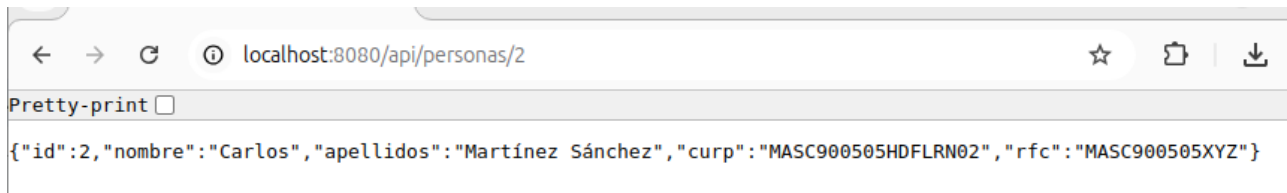
Para ingresar datos a la base de datos: data.sql

Tabla Personas

1.- Método GET (all) <http://localhost:8080/api/personas/>



2. Método GET(id) <http://localhost:8080/api/personas/{id}>



3. Método POST <http://localhost:8080/api/personas/>

POST

▼

http://localhost:8080/api/personas/

ParamsAuthorizationHeaders (8)Body ●ScriptsTests

☐ none☐ form-data☐ x-www-form-urlencoded☒ raw☐ binary

```
1  {
2    "nombre": "Roberto",
3    "apellidos": "Sánchez Mendoza",
4    "curp": "SAMR800101HDFLPN09",
5    "rfc": "SAMR800101XYZ"
6  }
```

BodyCookiesHeaders (5)Test Results🔄

{ } JSON ▼

▶ Preview🔗 Visualize ▼

```
1  {
2    "id": 6,
3    "nombre": "Roberto",
4    "apellidos": "Sánchez Mendoza",
5    "curp": "SAMR800101HDFLPN09",
6    "rfc": "SAMR800101XYZ"
7  }
```

4. Método PUT <http://localhost:8080/api/personas/{id}>

PUT ▼ http://localhost:8080/api/personas/6

Params Authorization Headers (8) **Body ●** Scripts Tests

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary

```
1 {
2   "nombre": "PRUEBA",
3   "apellidos": "FINAL"
4 }
```

Body Cookies Headers (5) Test Results ↺

{ } JSON ▼ ▶ Preview 🔍 Visualize ▼

```
1 {
2   "id": 6,
3   "nombre": "PRUEBA",
4   "apellidos": "FINAL",
5   "curp": "SAMR800101HDFLPN09",
6   "rfc": "SAMR800101XYZ"
7 }
```

5. Método DELETE <http://localhost:8080/api/v1/alumnos/{id}>

DELETE ▼ http://localhost:8080/api/personas/6

Params Authorization Headers (6) Body Scripts Tests ⋮

Body Cookies Headers (5) Test Results ↺

{ } JSON ▼ ▶ Preview 🔍 Visualize ▼

```
1 {
2   "id": 6,
3   "nombre": "PRUEBA",
4   "apellidos": "FINAL",
5   "curp": "SAMR800101HDFLPN09",
6   "rfc": "SAMR800101XYZ"
7 }
```

6.- Manejo de excepciones

POST ▼ http://localhost:8080/api/personas/

Params Authorization Headers (8) **Body** ● Scripts Tests Se

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary

```
1 {
2   "apellidos": "Sánchez Mendoza",
3   "curp": "SAMR800101HDFLPN09",
4   "rfc": "SAMR800101XYZ"
5 }
```

Body Cookies Headers (4) Test Results 🕒 400 Bad Request

{} JSON ▼ ▶ Preview 🔍 Visualize ▼

```
1 {
2   "nombre": "El nombre no puede estar vacío"
3 }
```

Tabla usuario(Relacionada a Persona)

1.- Método GET (all) <http://localhost:8080/api/usuarios/>

← → ↻ ⓘ localhost:8080/api/usuarios/ ☆ 📄

Pretty-print ☐

```
[{"id":1,"nombre":"mgonzalez","password":"password123","correo":"maria.gonzalez@email.com","persona":
{"id":1,"nombre":"María","apellidos":"González Pérez","curp":"G0PM850101MDFLRN01","rfc":"G0PM850101ABC"}},
{"id":2,"nombre":"cmartinez","password":"securepass","correo":"carlos.martinez@email.com","persona":
{"id":2,"nombre":"Carlos","apellidos":"Martínez Sánchez","curp":"MASC900505HDFLRN02","rfc":"MASC900505XYZ"}},
{"id":3,"nombre":"alopez","password":"ana1234","correo":"ana.lopez@email.com","persona":
{"id":3,"nombre":"Ana","apellidos":"López Ramírez","curp":"LORA920202MDFLRN03","rfc":"LORA920202DEF"}},
{"id":4,"nombre":"pherandez","password":"pedro2023","correo":"pedro.hernandez@email.com","persona":
{"id":4,"nombre":"Pedro","apellidos":"Hernández García","curp":"HEGP870707HDFLRN04","rfc":"HEGP870707GHI"}},
{"id":5,"nombre":"ldiaz","password":"lauraPass","correo":"laura.diaz@email.com","persona":
{"id":5,"nombre":"Laura","apellidos":"Díaz Fernández","curp":"DIFL950909MDFLRN05","rfc":"DIFL950909JKL"}}]
```

2. Método GET(id) <http://localhost:8080/api/usuarios/{id}>

← → ↻ ⓘ localhost:8080/api/usuarios/4 ☆ 📄

Pretty-print ☐

```
{"id":4,"nombre":"pherandez","password":"pedro2023","correo":"pedro.hernandez@email.com","persona":
{"id":4,"nombre":"Pedro","apellidos":"Hernández García","curp":"HEGP870707HDFLRN04","rfc":"HEGP870707GHI"}}
```

3. Método POST <http://localhost:8080/api/usuarios/>

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

Params Authorization Headers (8) **Body** Scripts

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw

```
1 {
2     "nombre": "JJARAMILLO",
3     "password": "Abcd123",
4     "correo": "prueba@prueba.com",
5     "persona": {
6         "id": 7
7     }
8 }
```

Body Cookies Headers (5) Test Results

{ } JSON Preview Visualize

```
1 {
2     "id": 6,
3     "nombre": "JJARAMILLO",
4     "password": "Abcd123",
5     "correo": "prueba@prueba.com",
6     "persona": {
7         "id": 7,
8         "nombre": null,
9         "apellidos": null,
10        "curp": null,
11        "rfc": null
12    }
13 }
```

4. Método PUT <http://localhost:8080/api/usuarios/{id}>

PUT <http://localhost:8080/api/usuarios/6>

Params Authorization Headers (8) **Body** Scripts

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw


```
1 {
2     "password": "PRUEBA",
3     "correo": "hola@prueba.com"
4 }
5 }
```

Body Cookies Headers (5) Test Results


{ } JSON Preview Visualize





```
1 {
2     "id": 6,
3     "nombre": "JJARAMILLO",
4     "password": "PRUEBA",
5     "correo": "hola@prueba.com",
6     "persona": {
7         "id": 7,
8         "nombre": "Jose",
9         "apellidos": "Jaramillo",
10        "curp": "SAMR800101HDFLPN09",
11        "rfc": "SAMR800101XYZ"
12    }
13 }
```

5. Método DELETE <http://localhost:8080/api/v1/usuarios/{id}>

DELETE  <http://localhost:8080/api/usuarios/6>


Params **Authorization** **Headers (6)** **Body** **Script**

Body **Cookies** **Headers (5)** **Test Results** 

{ } JSON   Preview  Visualize 

```
1  {
2    "id": 6,
3    "nombre": "JJARAMILLO",
4    "password": "PRUEBA",
5    "correo": "hola@prueba.com",
6    "persona": {
7      "id": 7,
8      "nombre": "Jose",
9      "apellidos": "Jaramillo",
10     "curp": "SAMR800101HDFLPN09",
11     "rfc": "SAMR800101XYZ"
12   }
13 }
```


6.- Manejo de excepciones





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

Params **Authorization** **Headers (8)** **Body** **Scripts** **Tests** **Settings**

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ custom

```
1  {
2    "password": "123",
3    "correo": "prueba@prueba.com",
4    "persona": {
5      "id": 7
6    }
7  }
```

Body **Cookies** **Headers (4)** **Test Results** 

{ } JSON   Preview  Visualize 

```
1  {
2    "password": "el tamaño debe estar entre 5 y 30",
3    "nombre": "El nombre de usuario no puede estar vacío"
4  }
```

400 Bad Request

Tabla rol

1.- Método GET (all) <http://localhost:8080/api/roles/>

```
localhost:8080/api/roles/

Pretty-print ☐

[{"id":1,"nombre":"ADMINISTRADOR"}, {"id":2,"nombre":"COLECCIONISTA"}, {"id":3,"nombre":"VISITA"}]
```

2. Método GET(id) <http://localhost:8080/api/roles/{id}>

```
localhost:8080/api/roles/2

Pretty-print ☐

{"id":2,"nombre":"COLECCIONISTA"}
```

3. Método POST <http://localhost:8080/api/roles/>

```
POST http://localhost:8080/api/roles/

Params  Authorization  Headers (8)  Body ●
none  form-data  x-www-form-urlencoded

1 {
2   "nombre": "PRUEBA",
3   "status": true
4 }
```

```
Body  Cookies  Headers (5)  Test Results  ↻

{} JSON ▾  ▶ Preview  🔗 Visualize ▾

1 {
2   "id": 4,
3   "nombre": "PRUEBA"
4 }
```


4. Método PUT <http://localhost:8080/api/roles/{id}>

PUT ⌵ <http://localhost:8080/api/roles/4>

Params Authorization Headers (8) **Body ●**

☐ none ☐ form-data ☐ x-www-form-urlencoded

```
1 {  
2   "nombre": "PRUEBA2",  
3   "status": false  
4 }
```

Body Cookies Headers (5) Test Results ↺

{} JSON ⌵ ▶ Preview 🔗 Visualize ⌵

```
1 {  
2   "id": 4,  
3   "nombre": "PRUEBA2"  
4 }
```

5. Método DELETE <http://localhost:8080/api/v1/roles/{id}>

DELETE ⌵ <http://localhost:8080/api/roles/4>

Params Authorization Headers (6) Body ⌵

Body Cookies Headers (5) Test Results ↺

{} JSON ⌵ ▶ Preview 🔗 Visualize ⌵

```
1 {  
2   "id": 4,  
3   "nombre": "PRUEBA2"  
4 }
```

6.- Manejo de excepciones

POST ▼ | http://localhost:8080/api/roles/

Params Authorization Headers (8) **Body** ● Scripts

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw

```
1 {
2   "nombre": "",
3   "status": true
4 }
```

Body Cookies Headers (4) Test Results | ↻

{ } JSON ▼ ▶ Preview 🔍 Visualize ▼

```
1 {
2   "nombre": "El rol no puede venir vacío"
3 }
```

Tabla usuario_rol (Tabla de enlace entre otras dos tablas)

1.- Método GET (all) <http://localhost:8080/api/usuario-roles/>

← → ↻ ⓘ localhost:8080/api/usuario-roles/ ☆ 📁 👤

Pretty-print ☐

```
[{"id":1,"usuario":{"id":1,"nombre":"maria.gonzalez","password":"password123","correo":"maria.gonzalez@email.com","persona":{"id":1,"nombre":"María","apellidos":"González Pérez","curp":"G0PM850101MDFLRN01","rfc":"G0PM850101ABC"}}, "rol":{"id":1,"nombre":"ADMINISTRADOR"},"status":true}, {"id":2,"usuario":{"id":2,"nombre":"carlos.martinez@email.com","persona":{"id":2,"nombre":"Carlos","apellidos":"Martínez Sánchez","curp":"MASC900505HDFLRN02","rfc":"MASC900505XYZ"}}, "rol":{"id":1,"nombre":"ADMINISTRADOR"},"status":true}, {"id":3,"usuario":{"id":3,"nombre":"ana.lopez@email.com","persona":{"id":3,"nombre":"Ana","apellidos":"López Ramírez","curp":"LORA920202MDFLRN03","rfc":"LORA920202DEF"}}, "rol":{"id":2,"nombre":"COLECCIONISTA"},"status":true}, {"id":4,"usuario":{"id":4,"nombre":"pedro.hernandez@email.com","persona":{"id":4,"nombre":"Pedro","apellidos":"Hernández García","curp":"HEGP870707HDFLRN04","rfc":"HEGP870707GHI"}}, "rol":{"id":2,"nombre":"COLECCIONISTA"},"status":true}, {"id":5,"usuario":{"id":5,"nombre":"laura.diaz@email.com","persona":{"id":5,"nombre":"Laura","apellidos":"Díaz Fernández","curp":"DIFL950909MDFLRN05","rfc":"DIFL950909JKL"}}, "rol":{"id":3,"nombre":"VISITA"},"status":true}]
```

2. Método GET(id) <http://localhost:8080/api/usuario-roles/{id}>

← → ↻ ⓘ localhost:8080/api/usuario-roles/3 ☆ 📁 👤

Pretty-print ☐

```
[{"id":3,"usuario":{"id":3,"nombre":"alopez","password":"anal234","correo":"ana.lopez@email.com","persona":{"id":3,"nombre":"Ana","apellidos":"López Ramírez","curp":"LORA920202MDFLRN03","rfc":"LORA920202DEF"}}, "rol":{"id":2,"nombre":"COLECCIONISTA"},"status":true}]
```

3. Método POST <http://localhost:8080/api/usuario-roles/>


The screenshot shows a REST client interface with a POST request to `http://localhost:8080/api/usuario-roles/`. The request body is a JSON object with two nested objects: `usuario` (with `id: 5`) and `rol` (with `id: 1`). The response body is a JSON object with the following structure:

```
1 {
2   "id": 7,
3   "usuario": {
4     "id": 5,
5     "nombre": null,
6     "password": null,
7     "correo": null,
8     "persona": null
9   },
10  "rol": {
11    "id": 1,
12    "nombre": null
13  },
14  "status": true
15 }
```

4. Método PUT

Debido a que esta tabla es una tabla de enlace entre dos tablas y esa es su única función no concidero adecuado tener un método put disponible


5. Método DELETE <http://localhost:8080/api/v1/usuario-roles/{id}>






DELETE  http://localhost:8080/api/usuario-roles/7

Params Authorization Headers (6) Body Scripts

Query Params


	Key	Value
	Key	Value


Body Cookies Headers (5) Test Results 

 JSON   Preview  Visualize 

```
1  {
2    "id": 7,
3    "usuario": {
4      "id": 5,
5      "nombre": "ldiaz",
6      "password": "lauraPass",
7      "correo": "laura.diaz@email.com",
8      "persona": {
9        "id": 5,
10       "nombre": "Laura",
11       "apellidos": "Díaz Fernández",
12       "curp": "DIFL950909MDFLRN05",
13       "rfc": "DIFL950909JKL"
14     }
15   },
16   "rol": {
17     "id": 1,
18     "nombre": "ADMINISTRADOR"
19   },
20   "status": true
21 }
```


6.- Manejo de excepciones






POST  http://localhost:8080/api/usuario-roles/

Params Authorization Headers (8) **Body**  Scripts

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw

```
1  {
2    "status": true
3  }
```

Body Cookies Headers (4) Test Results 

 JSON   Preview  Visualize 


```
1  {
2    "usuario": "Usuario necesario",
3    "rol": "Rol necesario"
4  }
```

Adicional documentación de api utilizando SWAGGER

URL para acceder

<http://localhost:8080/swagger-ui/index.html#/>

localhost:8080/swagger-ui/index.html#/

Swagger
powered by SMARTBEAR

Select a definition

coelin

OpenAPI definition

v0

OAS3

/api-docs/coelin

Servers

http://localhost:8080 - Generated server url

Personas

API para gestión de personas

GET

/api/personas/

Obtener todas las personas

POST

/api/personas/

Crear una nueva persona

GET

/api/personas/{id}

Obtener personas por id

PUT

/api/personas/{id}

Actualizar persona por id

DELETE

/api/personas/{id}

Eliminar persona por id

Roles

API para gestión de roles

GET

/api/roles/

Obtener todos los roles registrados

POST

/api/roles/

Crear un nuevo rol

GET

/api/roles/{id}

Obtener rol por id

PUT

/api/roles/{id}

Actualizar rol por id

DELETE

/api/roles/{id}

Eliminar rol por id

Usuario-Rol

API para gestión de enlaces entre las tablas de usuario y rol

GET

/api/usuario-roles/

Obtener todos los enlaces entre usuario y rol registrados

POST

/api/usuario-roles/

Crear un nuevo usuario-rol

GET

/api/usuario-roles/{id}

Obtener usuario-rol por id

DELETE

/api/usuario-roles/{id}

Eliminar usuario-rol por id

Usuarios

API para gestión de usuarios

GET

/api/usuarios/

Obtener todos los usuarios registrados

POST

/api/usuarios/

Crear un nuevo usuario

GET

/api/usuarios/{id}

Obtener usuario por id

PUT

/api/usuarios/{id}

Actualizar usuario por id

DELETE

/api/usuarios/{id}

Eliminar usuario por id

Documentación JSON:

http://localhost:8080/api-docs

← → ↺

localhost:8080/api-docs

Google Lens

Incognito (2)

Print-pretty

```
{
  "swagger": "3.0.1",
  "info": {
    "title": "OpenAPI definition",
    "version": "v0",
    "servers": [
      {
        "url": "http://localhost:8080",
        "description": "Generated server url"
      }
    ],
    "tags": [
      {
        "name": "Roles",
        "description": "API para gesti\u00f3n de roles"
      },
      {
        "name": "Usuarios",
        "description": "API para gesti\u00f3n de usuarios"
      },
      {
        "name": "Usuario-Rol",
        "description": "API para gesti\u00f3n de enlaces entre las tablas de usuario y rol"
      },
      {
        "name": "Personas",
        "description": "API para gesti\u00f3n de personas"
      }
    ],
    "paths": {
      "/api/usuarios/{id}": {
        "get": {
          "tags": [
            "Usuarios"
          ],
          "summary": "Obtener usuario por id",
          "description": "Retorna el usuario registrado con el id solicitado",
          "operationId": "getById",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            }
          ],
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "application/json": {
                  "schema": {
                    "$ref": "#/components/schemas/Usuario"
                  }
                }
              }
            }
          },
          "put": {
            "tags": [
            "Usuarios"
          ],
          "summary": "Actualizar usuario por id",
          "description": "Actualiza el registro de un usuario",
          "operationId": "update",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            }
          ],
          "requestBody": {
            "content": {
              "application/json": {
                "schema": {
                  "type": "integer",
                  "format": "int32"
                }
              }
            }
          },
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "application/json": {
                  "schema": {
                    "$ref": "#/components/schemas/Usuario"
                  }
                }
              }
            },
            "400": {
              "description": "Error",
              "content": {
                "application/json": {
                  "schema": {
                    "type": "integer",
                    "format": "int32"
                  }
                }
              }
            }
          }
        },
        "delete": {
          "tags": [
            "Usuarios"
          ],
          "summary": "Eliminar usuario por id",
          "description": "Elimina el registro de un usuario por id",
          "operationId": "delete",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            }
          ],
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "application/json": {
                  "schema": {
                    "type": "integer",
                    "format": "int32"
                  }
                }
              }
            }
          }
        }
      },
      "/api/roles/{id}": {
        "get": {
          "tags": [
            "Roles"
          ],
          "summary": "Obtener rol por id",
          "description": "Retorna el rol registrado con el id solicitado",
          "operationId": "getById",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            }
          ],
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "application/json": {
                  "schema": {
                    "$ref": "#/components/schemas/Rol"
                  }
                }
              }
            }
          },
          "put": {
            "tags": [
            "Roles"
          ],
          "summary": "Actualizar rol por id",
          "description": "Actualiza el registro de un rol",
          "operationId": "update",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            }
          ],
          "requestBody": {
            "content": {
              "application/json": {
                "schema": {
                  "type": "integer",
                  "format": "int32"
                }
              }
            }
          },
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "application/json": {
                  "schema": {
                    "$ref": "#/components/schemas/Rol"
                  }
                }
              }
            },
            "400": {
              "description": "Error",
              "content": {
                "application/json": {
                  "schema": {
                    "type": "integer",
                    "format": "int32"
                  }
                }
              }
            }
          }
        },
        "delete": {
          "tags": [
            "Roles"
          ],
          "summary": "Eliminar rol por id",
          "description": "Elimina el registro de un rol por id",
          "operationId": "delete",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            }
          ],
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "application/json": {
                  "schema": {
                    "type": "integer",
                    "format": "int32"
                  }
                }
              }
            }
          }
        }
      },
      "/api/personas/{id}": {
        "get": {
          "tags": [
            "Personas"
          ],
          "summary": "Obtener persona por id",
          "description": "Retorna la persona registrada con el id solicitado",
          "operationId": "getById",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            }
          ],
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "application/json": {
                  "schema": {
                    "$ref": "#/components/schemas/Persona"
                  }
                }
              }
            }
          },
          "put": {
            "tags": [
            "Personas"
          ],
          "summary": "Actualizar persona por id",
          "description": "Actualiza el registro de una persona",
          "operationId": "update",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            }
          ],
          "requestBody": {
            "content": {
              "application/json": {
                "schema": {
                  "type": "integer",
                  "format": "int32"
                }
              }
            }
          },
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "application/json": {
                  "schema": {
                    "$ref": "#/components/schemas/Persona"
                  }
                }
              }
            },
            "400": {
              "description": "Error",
              "content": {
                "application/json": {
                  "schema": {
                    "type": "integer",
                    "format": "int32"
                  }
                }
              }
            }
          }
        },
        "delete": {
          "tags": [
            "Personas"
          ],
          "summary": "Eliminar persona por id",
          "description": "Elimina el registro de una persona por id",
          "operationId": "delete",
          "parameters": [
            {
              "name": "id",
              "in": "path",
              "required": true,
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            }
          ],
          "responses": {
            "200": {
              "description": "OK",
              "content": {
                "application/json": {
                  "schema": {
                    "type": "integer",
                    "format": "int32"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```