

Universidad Autónoma Juan Misael Saracho
Facultad de Ciencias Integradas del Gran Chaco
Carrera Ingeniería Informática



DOCUMENTACION API REST FULL

MATERIA: TECNOLOGIA DE PROGRAMACION EN RED

SIGLA: INF423

ESTUDIANTES:

- José Eduardo Reyes Flores
- Cesar Ernesto Mosquez Cabrera

DOCENTE: Elias Cassal Baldiviezo

Fecha de presentación: 22/10/2024

Contenido

Documentación API RESTful - Usuarios, Libros, Géneros	4
Descripción.....	4
Recursos Disponibles	4
Endpoints Usuarios	4
1. Obtener todos los usuarios	4
2. Obtener un usuario por ID	4
3. Crear un nuevo usuario	5
4. Actualizar un usuario existente.....	5
5. Eliminar un usuario	6
Endpoints Libros	7
1. Obtener todos los libros	7
2. Obtener un libro por ID.....	7
3. Crear un nuevo libro	8
4. Actualizar un libro existente	9
5. Eliminar un libro	10
Endpoints Géneros	10
1. Obtener todos los géneros.....	10
2. Obtener un género por ID	10
3. Crear un nuevo género	11
4. Actualizar un género existente	12
5. Eliminar un género.....	12
Endpoints Clientes	13
1. Obtener todos los clientes	13
2. Obtener un cliente por ID.....	13
3. Crear un nuevo cliente	14
4. Actualizar un cliente existente	15
5. Eliminar un cliente	15
Endpoints Préstamos	16
1. Obtener todos los préstamos	16
2. Obtener un préstamo por ID.....	16
3. Crear un nuevo préstamo	17
4. Actualizar un préstamo existente.....	17
5. Eliminar un préstamo	18
Endpoints Autores.....	19
1. Obtener todos los autores	19

2. Obtener un autor por ID	19
3. Crear un nuevo autor.....	20
4. Actualizar un autor existente	20
5. Eliminar un autor.....	21

Documentación API RESTful - Usuarios, Libros, Géneros

Descripción

Esta API permite gestionar usuarios, libros y géneros mediante operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en una base de datos. Los datos se almacenan en MySQL, y las operaciones se realizan utilizando PHP.

Base URL: `http://localhost/{nombre de la carpeta}/biblioteca`

Recursos Disponibles

- `usuario`: Recurso que representa a los usuarios.
- `libro`: Recurso que representa los libros.
- `genero`: Recurso que representa los géneros literarios.
- `cliente`: Recurso que representa a los clientes en el sistema.
- `prestamo`: Recurso que representa los préstamos en el sistema.
- `autor`: Recurso que representa a los autores en el sistema.

Endpoints Usuarios

1. Obtener todos los usuarios

- **Descripción:** Obtiene una lista de todos los usuarios registrados en el sistema.
- **Método:** GET
- **Endpoint:** `/usuarios`
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
[
  {
    "id_usuario": 1,
    "login_usuario": "juan",
    "perfil": "admin"
  },
  {
    "id_usuario": 2,
    "login_usuario": "maria",
    "perfil": "usuario"
  }
]
```

2. Obtener un usuario por ID

- **Descripción:** Obtiene la información de un usuario específico por su ID.
- **Método:** GET
- **Endpoint:** `/usuarios/{id}`

- **Parámetros de ruta:**
 - `id`: ID del usuario que se desea obtener.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "id_usuario": 1,
  "login_usuario": "juan",
  "perfil": "admin"
}
```

- **Respuesta de error:**
 - Si el usuario no se encuentra:

```
{
  "status": 404,
  "detalle": "Usuario no encontrado"
}
```

3. Crear un nuevo usuario

- **Descripción:** Crea un nuevo usuario en la base de datos.
- **Método:** POST
- **Endpoint:** /usuarios
- **Cuerpo de la solicitud:**

```
{
  "login_usuario": "juan",
  "password": "123456",
  "perfil": "admin"
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Registro exitoso",
  "usuario": {
    "login_usuario": "juan",
    "perfil": "admin"
  }
}
```

- **Respuesta de error:**
 - Si los datos están incompletos o el usuario ya existe:

```
{
  "status": 400,
  "detalle": "Faltan datos obligatorios"
}
```

4. Actualizar un usuario existente

- **Descripción:** Actualiza los datos de un usuario específico.
- **Método:** PUT
- **Endpoint:** /usuarios/{id}
- **Parámetros de ruta:**
 - id: ID del usuario que se desea actualizar.
- **Cuerpo de la solicitud:**

```
{
  "login_usuario": "juan_actualizado",
  "password": "nueva_password",
  "perfil": "usuario",
  "estado": "activo"
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Actualización con éxito",
  "datos": {
    "login_usuario": "juan_actualizado",
    "perfil": "usuario",
    "estado": "activo"
  }
}
```

- **Respuesta de error:**
 - Si el ID es incorrecto o faltan datos:

```
{
  "status": 400,
  "detalle": "ID requerido para actualizar"
}
```

5. Eliminar un usuario

- **Descripción:** Elimina un usuario específico de la base de datos.
- **Método:** DELETE
- **Endpoint:** /usuarios/{id}
- **Parámetros de ruta:**
 - id: ID del usuario que se desea eliminar.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Eliminación con éxito"
}
```

- **Respuesta de error:**
 - Si el ID no es válido o no se proporciona:

```
{
```

```
    "status": 400,  
    "detalle": "ID requerido para eliminar"  
}
```

Endpoints Libros

1. Obtener todos los libros

- **Descripción:** Obtiene una lista de todos los libros registrados en el sistema.
- **Método:** GET
- **Endpoint:** /libros
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
[  
  {  
    "id_libro": 1,  
    "titulo": "El Principito",  
    "autor": "Antoine de Saint-Exupéry",  
    "genero": "Ficción",  
    "sinopsis": "Un cuento poético...",  
    "fecha_publicacion": "1943-04-06",  
    "editorial": "Reynal & Hitchcock",  
    "numero_paginas": 96,  
    "stock": 100,  
    "imagen_libro": "principito.jpg"  
  },  
  {  
    "id_libro": 2,  
    "titulo": "1984",  
    "autor": "George Orwell",  
    "genero": "Ciencia Ficción",  
    "sinopsis": "Una distopía...",  
    "fecha_publicacion": "1949-06-08",  
    "editorial": "Secker & Warburg",  
    "numero_paginas": 328,  
    "stock": 50,  
    "imagen_libro": "1984.jpg"  
  }  
]
```

2. Obtener un libro por ID

- **Descripción:** Obtiene la información de un libro específico por su ID.
- **Método:** GET
- **Endpoint:** /libros/{id}
- **Parámetros de ruta:**
 - id: ID del libro que se desea obtener.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{  
  "status": 200,  
  "detalle": "libro encontrado",  
}
```

```

"libro": {
  "id_libro": 1,
  "titulo": "El Principito",
  "autor": "Antoine de Saint-Exupéry",
  "genero": "Ficción",
  "sinopsis": "Un cuento poético...",
  "fecha_publicacion": "1943-04-06",
  "editorial": "Reynal & Hitchcock",
  "numero_paginas": 96,
  "stock": 100,
  "imagen_libro": "principito.jpg"
}
}

```

- **Respuesta de error:**
 - Si el libro no se encuentra:

```

{
  "status": 404,
  "detalle": "Libro no encontrado"
}

```

3. Crear un nuevo libro

- **Descripción:** Crea un nuevo libro en la base de datos.
- **Método:** POST
- **Endpoint:** /libros
- **Cuerpo de la solicitud:**

```

{
  "titulo": "El Principito",
  "autor": "Antoine de Saint-Exupéry",
  "genero": "Ficción",
  "sinopsis": "Un cuento poético...",
  "fecha_publicacion": "1943-04-06",
  "editorial": "Reynal & Hitchcock",
  "numero_paginas": 96,
  "stock": 100,
  "imagen_libro": "principito.jpg"
}

```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```

{
  "status": 200,
  "detalle": "Registro exitoso",
  "libro": {
    "titulo": "El Principito",
    "autor": "Antoine de Saint-Exupéry",
    "genero": "Ficción",
    "sinopsis": "Un cuento poético...",
    "fecha_publicacion": "1943-04-06",
    "editorial": "Reynal & Hitchcock",
    "numero_paginas": 96,
    "stock": 100,
    "imagen_libro": "principito.jpg"
  }
}

```



```
}  
}
```

- **Respuesta de error:**
 - Si faltan datos obligatorios o el libro ya existe:

```
{  
  "status": 400,  
  "detalle": "Faltan datos obligatorios"  
}
```

4. Actualizar un libro existente

- **Descripción:** Actualiza los datos de un libro específico.
- **Método:** PUT
- **Endpoint:** /libros/{id}
- **Parámetros de ruta:**
 - id: ID del libro que se desea actualizar.
- **Cuerpo de la solicitud:**

```
{  
  "titulo": "El Principito (Edición Revisada)",  
  "autor": "Antoine de Saint-Exupéry",  
  "genero": "Ficción",  
  "sinopsis": "Un cuento poético revisado...",  
  "fecha_publicacion": "1943-04-06",  
  "editorial": "Reynal & Hitchcock",  
  "numero_paginas": 100,  
  "stock": 150,  
  "imagen_libro": "principito_actualizado.jpg"  
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{  
  "status": 200,  
  "detalle": "Actualización completa",  
  "libro": {  
    "titulo": "El Principito (Edición Revisada)",  
    "autor": "Antoine de Saint-Exupéry",  
    "genero": "Ficción",  
    "sinopsis": "Un cuento poético revisado...",  
    "fecha_publicacion": "1943-04-06",  
    "editorial": "Reynal & Hitchcock",  
    "numero_paginas": 100,  
    "stock": 150,  
    "imagen_libro": "principito_actualizado.jpg"  
  }  
}
```

- **Respuesta de error:**
 - Si el ID es incorrecto o faltan datos:

```
{  
  "status": 400,
```

```
    "detalle": "ID requerido para actualizar"
  }
```

5. Eliminar un libro

- **Descripción:** Elimina un libro específico de la base de datos.
- **Método:** DELETE
- **Endpoint:** /libros/{id}
- **Parámetros de ruta:**
 - id: ID del libro que se desea eliminar.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Eliminación completa"
}
```

- **Respuesta de error:**
 - Si el ID no es válido o no se proporciona:

```
{
  "status": 400,
  "detalle": "ID requerido para eliminar"
}
```

Endpoints Géneros

1. Obtener todos los géneros

- **Descripción:** Obtiene una lista de todos los géneros registrados en el sistema.
- **Método:** GET
- **Endpoint:** /generos
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
[
  {
    "id_genero": 1,
    "nombre": "Ficción",
    "descripcion": "Género narrativo basado en hechos imaginarios."
  },
  {
    "id_genero": 2,
    "nombre": "Ciencia Ficción",
    "descripcion": "Género que mezcla la ciencia con la imaginación."
  }
]
```

2. Obtener un género por ID

- **Descripción:** Obtiene la información de un género específico por su ID.
- **Método:** GET
- **Endpoint:** /generos/{id}
- **Parámetros de ruta:**
 - id: ID del género que se desea obtener.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Genero encontrado",
  "genero": {
    "id_genero": 1,
    "nombre": "Ficción",
    "descripcion": "Género narrativo basado en hechos imaginarios."
  }
}
```

- **Respuesta de error:**
 - Si el género no se encuentra:

```
{
  "status": 404,
  "detalle": "Genero no encontrado"
}
```

3. Crear un nuevo género

- **Descripción:** Crea un nuevo género en la base de datos.
- **Método:** POST
- **Endpoint:** /generos
- **Cuerpo de la solicitud:**

```
{
  "nombre": "Fantasía",
  "descripcion": "Género narrativo que utiliza elementos sobrenaturales."
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Registro exitoso",
  "genero": {
    "nombre": "Fantasía",
    "descripcion": "Género narrativo que utiliza elementos sobrenaturales."
  }
}
```

- **Respuesta de error:**
 - Si faltan datos obligatorios o el género ya existe:

```
{
  "status": 400,
  "detalle": "Faltan datos obligatorios"
}
```

4. Actualizar un género existente

- **Descripción:** Actualiza los datos de un género específico.
- **Método:** PUT
- **Endpoint:** /generos/{id}
- **Parámetros de ruta:**
 - id: ID del género que se desea actualizar.
- **Cuerpo de la solicitud:**

```
{
  "nombre": "Fantasía Épica",
  "descripcion": "Subgénero de la fantasía que incluye héroes y guerras."
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Actualización completa",
  "genero": {
    "nombre": "Fantasía Épica",
    "descripcion": "Subgénero de la fantasía que incluye héroes y guerras."
  }
}
```

- **Respuesta de error:**
 - Si el ID es incorrecto o faltan datos:

```
{
  "status": 400,
  "detalle": "ID requerido para actualizar"
}
```

5. Eliminar un género

- **Descripción:** Elimina un género específico de la base de datos.
- **Método:** DELETE
- **Endpoint:** /generos/{id}
- **Parámetros de ruta:**
 - id: ID del género que se desea eliminar.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Eliminación completa"
}
```

```
}
```

- **Respuesta de error:**
 - Si el ID no es válido o no se proporciona:

```
{
  "status": 400,
  "detalle": "ID requerido para eliminar"
}
```

Endpoints Clientes

1. Obtener todos los clientes

- **Descripción:** Obtiene una lista de todos los clientes registrados en el sistema.
- **Método:** GET
- **Endpoint:** /clientes
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
[
  {
    "id_cliente": 1,
    "nombre": "Juan",
    "apellido": "Pérez",
    "email": "juan.perez@example.com",
    "telefono": "555-555-555",
    "direccion": "Av. Siempre Viva 123",
    "imagen_cliente": "juan_perez.jpg"
  },
  {
    "id_cliente": 2,
    "nombre": "Ana",
    "apellido": "Gómez",
    "email": "ana.gomez@example.com",
    "telefono": "444-444-444",
    "direccion": "Calle Falsa 456",
    "imagen_cliente": "ana_gomez.jpg"
  }
]
```

2. Obtener un cliente por ID

- **Descripción:** Obtiene la información de un cliente específico por su ID.
- **Método:** GET
- **Endpoint:** /clientes/{id}
- **Parámetros de ruta:**
 - id: ID del cliente que se desea obtener.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Cliente encontrado",
}
```

```

    "cliente": {
      "id_cliente": 1,
      "nombre": "Juan",
      "apellido": "Pérez",
      "email": "juan.perez@example.com",
      "telefono": "555-555-555",
      "direccion": "Av. Siempre Viva 123",
      "imagen_cliente": "juan_perez.jpg"
    }
  }
}

```

- **Respuesta de error:**
 - Si el cliente no se encuentra:

```

{
  "status": 404,
  "detalle": "Cliente no existe"
}

```

3. Crear un nuevo cliente

- **Descripción:** Crea un nuevo cliente en la base de datos.
- **Método:** POST
- **Endpoint:** /clientes
- **Cuerpo de la solicitud:**

```

{
  "nombre": "Carlos",
  "apellido": "Sánchez",
  "email": "carlos.sanchez@example.com",
  "telefono": "666-666-666",
  "direccion": "Calle Verde 789",
  "imagen_cliente": "carlos_sanchez.jpg"
}

```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```

{
  "status": 201,
  "detalle": "Registro exitoso de cliente",
  "cliente": {
    "nombre": "Carlos",
    "apellido": "Sánchez",
    "email": "carlos.sanchez@example.com",
    "telefono": "666-666-666",
    "direccion": "Calle Verde 789",
    "imagen_cliente": "carlos_sanchez.jpg"
  }
}

```

- **Respuesta de error:**
 - Si los datos están incompletos o el cliente ya existe:

```

{
  "status": 400,
  "detalle": "Faltan datos obligatorios"
}

```

```
}
```

4. Actualizar un cliente existente

- **Descripción:** Actualiza los datos de un cliente específico.
- **Método:** PUT
- **Endpoint:** /clientes/{id}
- **Parámetros de ruta:**
 - id: ID del cliente que se desea actualizar.
- **Cuerpo de la solicitud:**

```
{
  "nombre": "Carlos Actualizado",
  "apellido": "Sánchez",
  "email": "carlos.sanchez@example.com",
  "telefono": "666-666-666",
  "direccion": "Calle Azul 101",
  "imagen_cliente": "carlos_sanchez_actualizado.jpg"
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Datos actualizados exitosamente",
  "cliente": {
    "nombre": "Carlos Actualizado",
    "apellido": "Sánchez",
    "email": "carlos.sanchez@example.com",
    "telefono": "666-666-666",
    "direccion": "Calle Azul 101",
    "imagen_cliente": "carlos_sanchez_actualizado.jpg"
  }
}
```

- **Respuesta de error:**
 - Si el ID es incorrecto o faltan datos:

```
{
  "status": 400,
  "detalle": "ID requerido para actualizar"
}
```

5. Eliminar un cliente

- **Descripción:** Elimina un cliente específico de la base de datos.
- **Método:** DELETE
- **Endpoint:** /clientes/{id}
- **Parámetros de ruta:**
 - id: ID del cliente que se desea eliminar.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Eliminación completa"
}
```

- **Respuesta de error:**
 - Si el ID no es válido o no se proporciona:

```
{
  "status": 400,
  "detalle": "ID requerido para eliminar"
}
```

Endpoints Préstamos

1. Obtener todos los préstamos

- **Descripción:** Obtiene una lista de todos los préstamos registrados en el sistema.
- **Método:** GET
- **Endpoint:** /prestamos
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
[
  {
    "id_prestamo": 1,
    "cliente": "Juan Pérez",
    "libro": "El Principito",
    "fecha_devolucion": "2024-10-30",
    "estado": "Pendiente"
  },
  {
    "id_prestamo": 2,
    "cliente": "María López",
    "libro": "1984",
    "fecha_devolucion": "2024-11-10",
    "estado": "Devuelto"
  }
]
```

2. Obtener un préstamo por ID

- **Descripción:** Obtiene la información de un préstamo específico por su ID.
- **Método:** GET
- **Endpoint:** /prestamos/{id}
- **Parámetros de ruta:**
 - id: ID del préstamo que se desea obtener.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Préstamo encontrado",
  "prestamo": {
```



```
    "id_prestamo": 1,  
    "cliente": "Juan Pérez",  
    "libro": "El Principito",  
    "fecha_devolucion": "2024-10-30",  
    "estado": "Pendiente"  
  }  
}
```

- **Respuesta de error:**
 - Si el préstamo no se encuentra:

```
{  
  "status": 404,  
  "detalle": "Préstamo no encontrado"  
}
```

3. Crear un nuevo préstamo

- **Descripción:** Crea un nuevo préstamo en la base de datos.
- **Método:** POST
- **Endpoint:** /prestamos
- **Cuerpo de la solicitud:**

```
{  
  "cliente": "Juan Pérez",  
  "libro": "El Principito",  
  "fecha_devolucion": "2024-10-30"  
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{  
  "status": 200,  
  "detalle": "Registro exitoso",  
  "prestamo": {  
    "cliente": "Juan Pérez",  
    "libro": "El Principito",  
    "fecha_devolucion": "2024-10-30"  
  }  
}
```

- **Respuesta de error:**
 - Si faltan datos obligatorios o el préstamo ya existe:

```
{  
  "status": 400,  
  "detalle": "Faltan datos obligatorios"  
}
```

4. Actualizar un préstamo existente

- **Descripción:** Actualiza los datos de un préstamo específico.
- **Método:** PUT
- **Endpoint:** /prestamos/{id}

- **Parámetros de ruta:**
 - `id`: ID del préstamo que se desea actualizar.
- **Cuerpo de la solicitud:**

```
{
  "cliente": "Juan Pérez",
  "libro": "El Principito",
  "fecha_devolucion": "2024-10-30",
  "estado": "Devuelto"
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Actualización completa",
  "prestamo": {
    "cliente": "Juan Pérez",
    "libro": "El Principito",
    "fecha_devolucion": "2024-10-30",
    "estado": "Devuelto"
  }
}
```

- **Respuesta de error:**
 - Si el ID es incorrecto o faltan datos:

```
{
  "status": 400,
  "detalle": "ID requerido para actualizar"
}
```

5. Eliminar un préstamo

- **Descripción:** Elimina un préstamo específico de la base de datos.
- **Método:** DELETE
- **Endpoint:** `/prestamos/{id}`
- **Parámetros de ruta:**
 - `id`: ID del préstamo que se desea eliminar.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Eliminación completa"
}
```

- **Respuesta de error:**
 - Si el ID no es válido o no se proporciona:

```
{
  "status": 400,
  "detalle": "ID requerido para eliminar"
}
```

Endpoints Autores

1. Obtener todos los autores

- **Descripción:** Obtiene una lista de todos los autores registrados en el sistema.
- **Método:** GET
- **Endpoint:** /autores
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
[
  {
    "id_autor": 1,
    "nombre": "Gabriel",
    "apellido": "García Márquez",
    "nacionalidad": "Colombiana",
    "fecha_nacimiento": "1927-03-06",
    "biografia": "Escritor y periodista...",
    "imagen_autor": "gabriel.jpg"
  },
  {
    "id_autor": 2,
    "nombre": "Isabel",
    "apellido": "Allende",
    "nacionalidad": "Chilena",
    "fecha_nacimiento": "1942-08-02",
    "biografia": "Autora de novelas...",
    "imagen_autor": "isabel.jpg"
  }
]
```

2. Obtener un autor por ID

- **Descripción:** Obtiene la información de un autor específico por su ID.
- **Método:** GET
- **Endpoint:** /autores/{id}
- **Parámetros de ruta:**
 - id: ID del autor que se desea obtener.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": {
    "id_autor": 1,
    "nombre": "Gabriel",
    "apellido": "García Márquez",
    "nacionalidad": "Colombiana",
    "fecha_nacimiento": "1927-03-06",
    "biografia": "Escritor y periodista...",
    "imagen_autor": "gabriel.jpg"
  }
}
```

- **Respuesta de error:**

- Si el autor no se encuentra:

```
{
  "status": 404,
  "detalle": "El autor no existe"
}
```

3. Crear un nuevo autor

- **Descripción:** Crea un nuevo autor en la base de datos.
- **Método:** POST
- **Endpoint:** /autores
- **Cuerpo de la solicitud:**

```
{
  "nombre": "Gabriel",
  "apellido": "García Márquez",
  "nacionalidad": "Colombiana",
  "fecha_nacimiento": "1927-03-06",
  "biografia": "Escritor y periodista...",
  "imagen_autor": "gabriel.jpg"
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "registro exitoso",
  "autor": {
    "nombre": "Gabriel",
    "apellido": "García Márquez",
    "nacionalidad": "Colombiana",
    "fecha_nacimiento": "1927-03-06",
    "biografia": "Escritor y periodista...",
    "imagen_autor": "gabriel.jpg"
  }
}
```

- **Respuesta de error:**
 - Si faltan datos obligatorios o el autor ya existe:

```
{
  "status": 400,
  "detalle": "Faltan datos obligatorios"
}
```

4. Actualizar un autor existente

- **Descripción:** Actualiza los datos de un autor específico.
- **Método:** PUT
- **Endpoint:** /autores/{id}
- **Parámetros de ruta:**
 - id: ID del autor que se desea actualizar.
- **Cuerpo de la solicitud:**

```
{
  "nombre": "Gabriel",
  "apellido": "García Márquez",
  "nacionalidad": "Colombiana",
  "fecha_nacimiento": "1927-03-06",
  "biografia": "Escritor y periodista destacado...",
  "imagen_autor": "gabriel_actualizado.jpg"
}
```

- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "actualizacion completa",
  "autor": {
    "nombre": "Gabriel",
    "apellido": "García Márquez",
    "nacionalidad": "Colombiana",
    "fecha_nacimiento": "1927-03-06",
    "biografia": "Escritor y periodista destacado...",
    "imagen_autor": "gabriel_actualizado.jpg"
  }
}
```

- **Respuesta de error:**
 - Si el ID es incorrecto o faltan datos:

```
{
  "status": 400,
  "detalle": "ID requerido para actualizar"
}
```

5. Eliminar un autor

- **Descripción:** Elimina un autor específico de la base de datos.
- **Método:** DELETE
- **Endpoint:** /autores/{id}
- **Parámetros de ruta:**
 - id: ID del autor que se desea eliminar.
- **Autenticación requerida:** No.
- **Respuesta exitosa:**

```
{
  "status": 200,
  "detalle": "Eliminacion completa"
}
```

- **Respuesta de error:**
 - Si el ID no es válido o no se proporciona:

```
{
  "status": 400,
  "detalle": "ID requerido para eliminar"
}
```