

# Usuario HATEOAS

## OBTENER USUARIO POR ID

Este endpoint nos permite obtener los datos de un usuario mediante el uso de su identificador único.

### Petición:

- **Método:** POST
- **URL:** `http://localhost:8081/api/usuarios/{id}`
  - Se reemplaza {id} con el ID del usuario a buscar. En este ejemplo usaremos el `1`.

### Cuerpo de la petición Postman

No aplica.

### Respuesta

Al hacer una búsqueda exitosa se entregará una respuesta con código `200` y un objeto JSON con los siguientes campos:

- `id` (integer): Identificador único del usuario.
- `nombre` (string): Nombre del usuario.
- `correo` (string): Correo electrónico del usuario.
- `rol` (string): Rol del usuario.
- `_links` (objeto): Links a operaciones CRUD relacionadas con el usuario.
  - `self`: Link al recurso actual.
  - `lista de usuarios`: Link a la lista de productos.
  - `actualizar usuario`: Link para actualizar al producto.
  - `guardar usuario`: Link para guardar un producto.
  - `eliminar usuario`: Link para eliminar el producto.

## EJEMPLO RESPUESTA: JSON

```
{
  "id": 1,
  "nombre": "PrimerUsuario HATEOAS",
  "correo": "usuario@gmail.com",
  "rol": "Usuario",
  "_links": {
    "self": {
      "href": "http://localhost:8081/api/usuarios/1"
    },
    "lista de usuarios": {
      "href": "http://localhost:8081/api/usuarios"
    },
    "actualizar usuario": {
      "href": "http://localhost:8081/api/usuarios/1"
    },
    "guardar usuario": {
      "href": "http://localhost:8081/api/usuarios"
    },
    "eliminar usuario": {
      "href": "http://localhost:8081/api/usuarios/1"
    }
  }
}
```

## OBTENER LISTA DE USUARIOS

Este endpoint nos entrega una lista de todos los usuarios almacenados en la base de datos.

### Petición

- **Método:** GET
- **URL:** <http://localhost:8081/api/usuarios>

## Cuerpo de la petición Postman

No aplica.

## Respuesta

Al hacer una petición exitosa se entregará una respuesta con código `200` y un objeto JSON con los siguientes campos:

- `_embedded` Contiene los recursos integrados.
- `usuarioList` Una matriz con los objetos clase usuario, cada uno con los siguientes datos:
  - `id` (integer): Identificador único del usuario.
  - `nombre` (string): Nombre del usuario.
  - `correo` (string): Correo electrónico del usuario.
  - `rol` (string): Rol del usuario.
  - `_links` (objeto): Links a operaciones CRUD relacionadas con el usuario.
    - `self` : Link al recurso actual.
    - `lista de usuarios` : Link a la lista de productos.
    - `actualizar usuario` : Link para actualizar al producto.
    - `guardar usuario` : Link para guardar un producto.
    - `eliminar usuario` : Link para eliminar el producto.

```
{
  "_embedded": {
    "usuarioList": [
      {
        "id": 4,
        "nombre": "Fulanito",
        "correo": "fulanito@gmail.com",
        "rol": "Usuario",
        "_links": {
          "self": {
```

```
        "href": "http://localhost:8081/api/usuarios/1"
      },
      "lista de usuarios": {
        "href": "http://localhost:8081/api/usuarios"
      },
      "actualizar usuario": {
        "href": "http://localhost:8081/api/usuarios/1"
      },
      "guardar usuario": {
        "href": "http://localhost:8081/api/usuarios"
      },
      "eliminar usuario": {
        "href": "http://localhost:8081/api/usuarios/1"
      }
    }
  }
}
```

## AGREGAR USUARIO

Este endpoint te permite agregar un nuevo usuario a la base de datos.

### Petición

- **Método:** POST
- **URL:** `http://localhost:8081/api/usuarios`

### Cuerpo de la petición Postman

Se necesitarán los siguientes datos en formato JSON para una petición exitosa:

- `nombre` (string): Nombre del usuario.
- `correo` (string): Correo electrónico del usuario.
- `rol` (string): Rol del usuario.

## EJEMPLO PETICIÓN: JSON

```
{
  "nombre": "Fulanito",
  "correo": "fulanito@gmail.com",
  "rol": "Usuario"
}
```

## Respuesta

Al hacer un guardado exitoso se entregará una respuesta con código **200** y un objeto JSON con los siguientes campos:

- **id** (integer): Identificador único del usuario.
- **nombre** (string): Nombre del usuario.
- **correo** (string): Correo electrónico del usuario.
- **rol** (string): Rol del usuario.
- **\_links** (objeto): Links a operaciones CRUD relacionadas con el usuario.
  - **self** : Link al recurso actual.
  - **lista de usuarios** : Link a la lista de productos.
  - **actualizar usuario** : Link para actualizar al producto.
  - **guardar usuario** : Link para guardar un producto.
  - **eliminar usuario** : Link para eliminar el producto.

## EJEMPLO RESPUESTA: JSON

```
{
  "id": 3,
  "nombre": "Fulanito",
  "correo": "fulanito@gmail.com",
  "rol": "Usuario",
  "_links": {
    "self": {
```

```
    "href": "http://localhost:8081/api/usuarios/3"
  },
  "lista de usuarios": {
    "href": "http://localhost:8081/api/usuarios"
  },
  "actualizar usuario": {
    "href": "http://localhost:8081/api/usuarios/3"
  },
  "guardar usuario": {
    "href": "http://localhost:8081/api/usuarios"
  },
  "eliminar usuario": {
    "href": "http://localhost:8081/api/usuarios/3"
  }
}
}
```

## ACTUALIZAR USUARIO

Este endpoint te permite actualizar la información de un usuario utilizando su ID para ello.

### Petición

- **Método:** PATCH
- **URL:** `http://localhost:8081/api/usuarios/{id}`
  - Se reemplaza {id} con el ID del usuario a modificar. En este ejemplo usaremos el `3`.

### Cuerpo de la petición Postman

Se necesitarán los siguientes datos en formato JSON para una petición exitosa:

- `nombre` (string): Nombre del usuario.

- `correo` (string): Correo electrónico del usuario.
- `rol` (string): Rol del usuario.

## EJEMPLO DE PETICIÓN: JSON

```
{
  "nombre": "Fulanito 2.0",
  "correo": "2.0@gmail.com",
  "rol": "Administrador"
}
```

## Respuesta

Al hacer un actualizarse el usuario exitosamente se entregará una respuesta con código `200` y un objeto JSON con los siguientes campos:

- `id` (integer): Identificador único del usuario.
- `nombre` (string): Nombre del usuario.
- `correo` (string): Correo electrónico del usuario.
- `rol` (string): Rol del usuario.
- `_links` (objeto): Links a operaciones CRUD relacionadas con el usuario.
  - `self` : Link al recurso actual.
  - `lista de usuarios` : Link a la lista de productos.
  - `actualizar usuario` : Link para actualizar al producto.
  - `guardar usuario` : Link para guardar un producto.
  - `eliminar usuario` : Link para eliminar el producto.

```
{
  "id": 3,
  "nombre": "Fulanito 2.0",
  "correo": "2.0@gmail.com",
  "rol": "Administrador",
  "_links": {
```

```
"self": {
  "href": "http://localhost:8081/api/usuarios/3"
},
"lista de usuarios": {
  "href": "http://localhost:8081/api/usuarios"
},
"actualizar usuario": {
  "href": "http://localhost:8081/api/usuarios/3"
},
"guardar usuario": {
  "href": "http://localhost:8081/api/usuarios"
},
"eliminar usuario": {
  "href": "http://localhost:8081/api/usuarios/3"
}
}
```

## ELIMINAR USUARIO

Este endpoint te permite eliminar a un usuario de la base de datos mediante el uso de su identificador único ID.

### Petición

- **Método:** DELETE
- **URL:** `http://localhost:8081/api/usuarios/{id}`
  - Se reemplaza {id} con el ID del usuario a modificar. En este ejemplo usaremos el `3`.

### Cuerpo de la petición Postman

No aplica.

### Respuesta



Al hacer un `PUT` para actualizar el usuario exitosamente se entregará una respuesta con código `200` y un objeto JSON con los siguientes campos:

- `_links` (objeto): Links a operaciones CRUD relacionadas con el usuario.
  - `self`: Link al recurso actual.
  - `lista de usuarios`: Link a la lista de productos.
  - `actualizar usuario`: Link para actualizar al producto.
  - `guardar usuario`: Link para guardar un producto.
  - `eliminar usuario`: Link para eliminar el producto.

## EJEMPLO DE RESPUESTA: JSON

```
{
  "_links": {
    "self": {
      "href": "http://localhost:8081/api/usuarios/1"
    },
    "lista de usuarios": {
      "href": "http://localhost:8081/api/usuarios"
    },
    "actualizar usuario": {
      "href": "http://localhost:8081/api/usuarios/0"
    },
    "guardar usuario": {
      "href": "http://localhost:8081/api/usuarios"
    },
    "buscar usuario": {
      "href": "http://localhost:8081/api/usuarios/0"
    }
  }
}
```