



# ESPECIFICACIÓN SERVICIO REST PARA LA ENTIDAD USUARIO

Pacto de Honor

Facultad de Ingeniería y Ciencias básicas.  
Politécnico Grancolombiano.

## Contenido

Control de versiones .....	3
Objetivo .....	4
URL .....	4
Métodos .....	4
find .....	4
Satisfactorio: .....	5
No satisfactorio: .....	5
create .....	5
Satisfactorio: .....	6
No satisfactorio: .....	6
edit .....	7
Satisfactorio: .....	7
No satisfactorio: .....	8
count .....	8
listUsers .....	9
login .....	10
Satisfactorio .....	10
No satisfactorio .....	10

## Control de versiones

Versión	Fecha	Autor	Revisado por	Cambios realizados
0.1	15/04/2017	Nicolás Rubiano		Versión inicial del documento.
0.2	17/04/2017	Nicolás Rubiano		Se añaden los métodos: <ul style="list-style-type: none"><li>- listUsers</li><li>- login</li></ul> No se permite el consumo directo del servicio.

## Objetivo

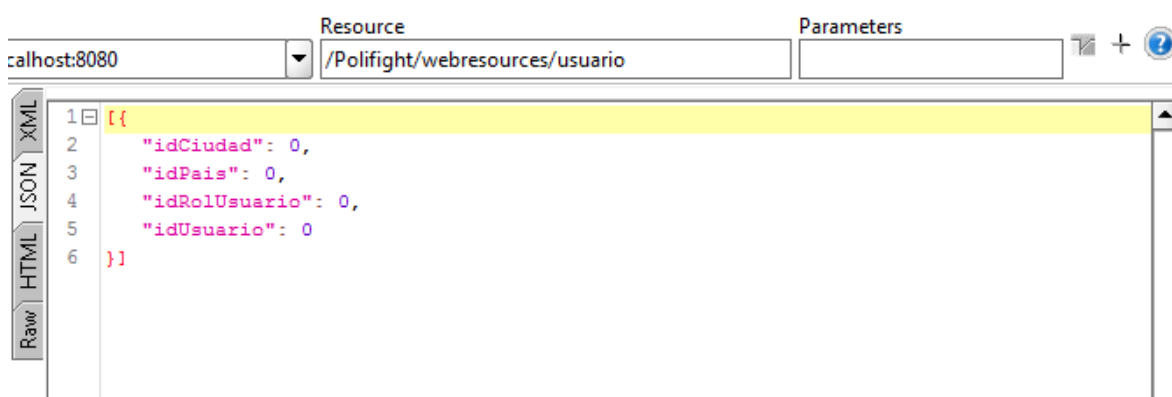
El objetivo del presente documento, es brindar herramientas de conocimiento los desarrolladores que desean consumir los servicios REST que proporcionará el área de desarrollo de Backend, específicamente sobre la entidad Usuario.

Los siguientes, son las especificaciones que se proporcionarán para tener acceso a todos los datos referentes sobre la entidad:

## URL

http://IP:Puerto/Polifight/webresources/usuario

Este servicio no permite el consumo directo, al intentar realizar dicho consumo, se arrojará una entidad de tipo Usuario vacío:



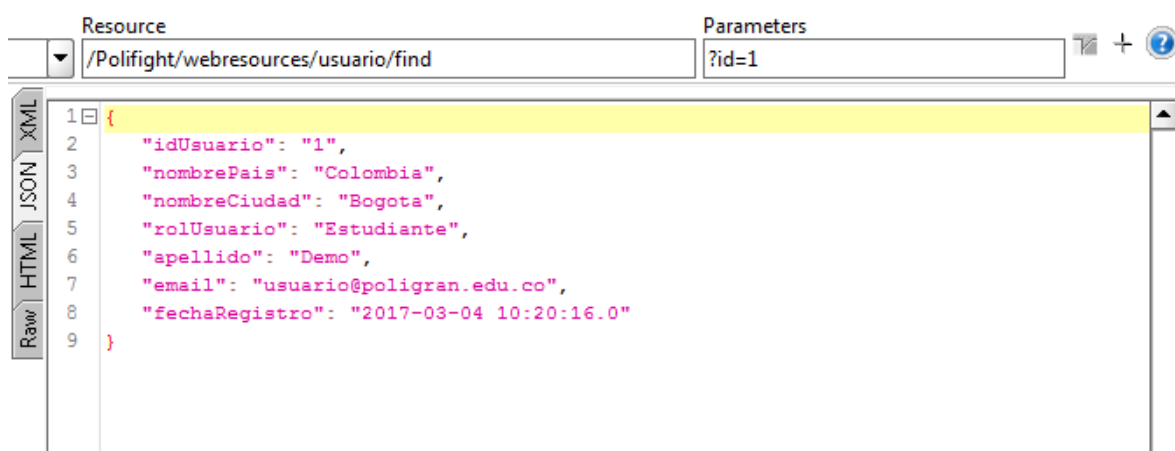
Para listar todos los datos que se encuentran en la base de datos, es necesario realizar el consumo utilizando el método [list](#)

## Métodos

### find

EndPoint: /Polifight/webresources/usuario/find

Este método se deberá consumir por método GET. Se recibirá el id (Parámetro "id") del rol y retornará todos los atributos asociados a dicho rol:



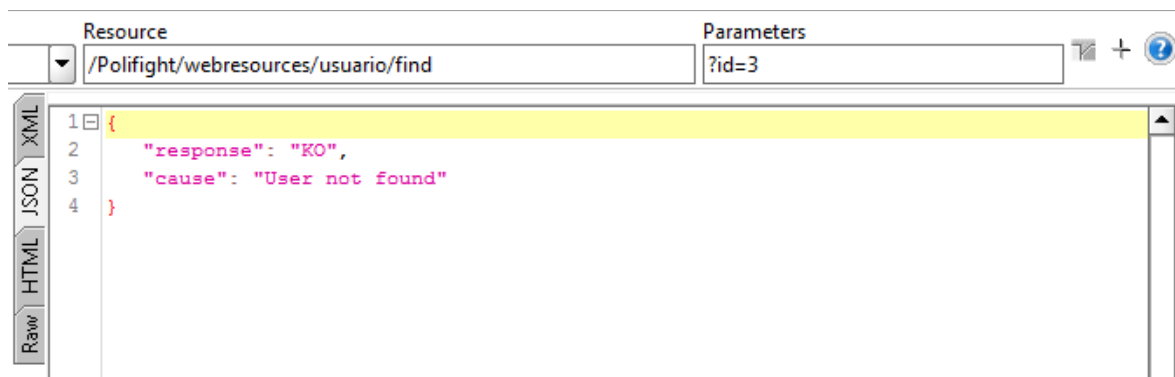
### Satisfactorio:

En el caso satisfactorio, se enviarán los atributos de la entidad en formato JSON.

### No satisfactorio:

Si el servicio tiene algún problema con la solicitud, se responderán dos campos:

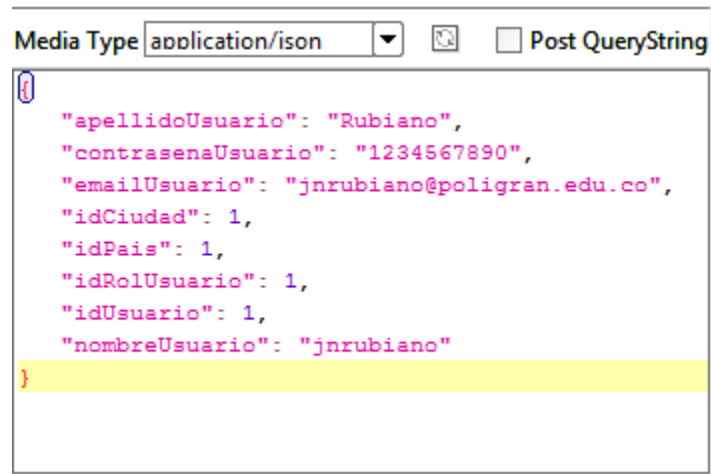
- "response": En caso de algún inconveniente se responderá "KO".
- "cause": Causa del error, para éste método se identifica sólo 1 error, Id del rol no encontrado en la DB



### create

EndPoint: /Polifight/webresources/usuario/create

Se debe enviar un JSON por el método POST con los atributos de la entidad Rol:

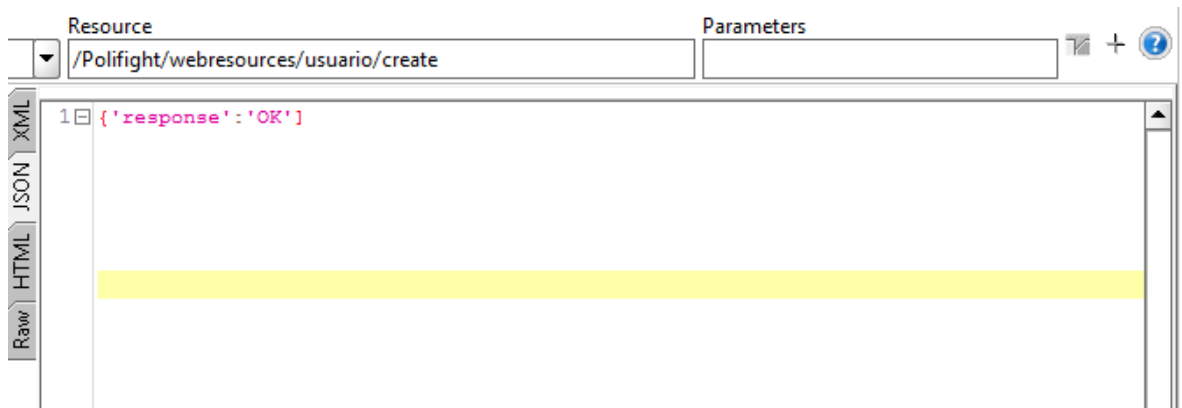


Es importante recalcar que el/los nombres de los atributos que se envían deben tener los nombres exactos que muestran en la imagen anterior.

Para ésta entidad no es necesario enviar el dato de la fecha de registro ya que este dato es capturado por el Backend.

#### Satisfactorio:

El servicio responde OK en el campo “response” en caso de que la creación se haya realizado con éxito:



id_usuario	id_pais	id_ciudad	id_rol_usuario	nombre_usuario	apellido_usuario	email_usuario	contrasena_usuario	fecha_registro
1	1	1	1	Estudiante	Demo	usuario@poligran.edu.co	demo123	2017-03-04 10:20:16
2	1	1	1	jnrubiano	Rubiano	jnrubiano@poligran.edu.co	1234567890	2017-04-15 18:28:47
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

#### No satisfactorio:

Si el servicio no puede procesar la solicitud se responderá un KO en el campo “response”. En este caso se incluirá el campo “cause” donde se indicará la causa del error

```

{

```

```

    "response": "KO",
    "cause": "Time Out"
}

```

## edit

EndPoint: /Polifight/webresources/usuario/edit

Método que servirá para modificar una entidad Rol. Se debe consumir por POST enviando un formato JSON con el parámetro "id". Éste es obligatorio, y seguido del/los atributo(s) que se deseen modificar.

Para efectos de ejemplo, modificaremos el nombre del datos insertado anteriormente en el ejemplo del método [create](#).

Media Type:  ☐ Post QueryString

```

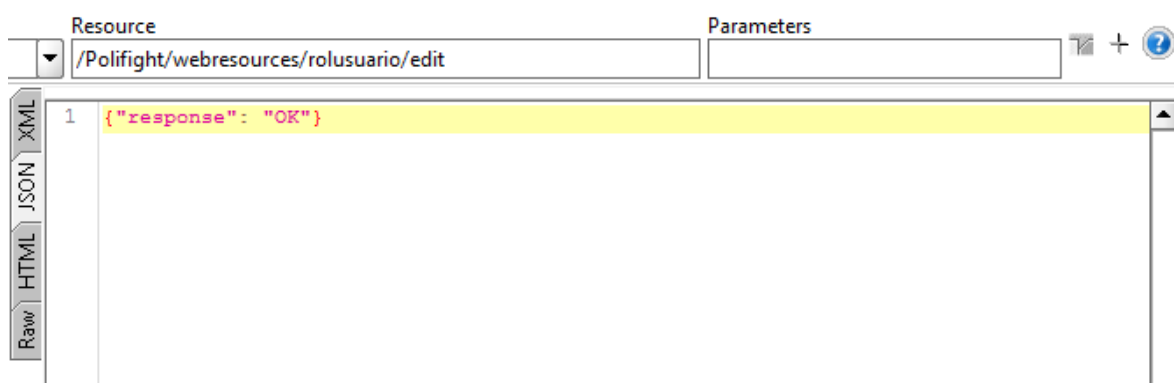
{
  "idUsuario": "2",
  "idCiudad": "52838"
}

```

	id_usuario	id_pais	id_ciudad	id_rol_usuario	nombre_usuario	apellido_usuario	email_usuario	contrasena_usuario	fecha_registro
▶	1	1	1	1	Estudiante	Demo	usuario@poligran.edu.co	demo123	2017-03-04 10:20:16
	2	1	52838	1	jnrubiano	Rubiano	jnrubiano@poligran.edu.co	1234567890	2017-04-15 18:28:47
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

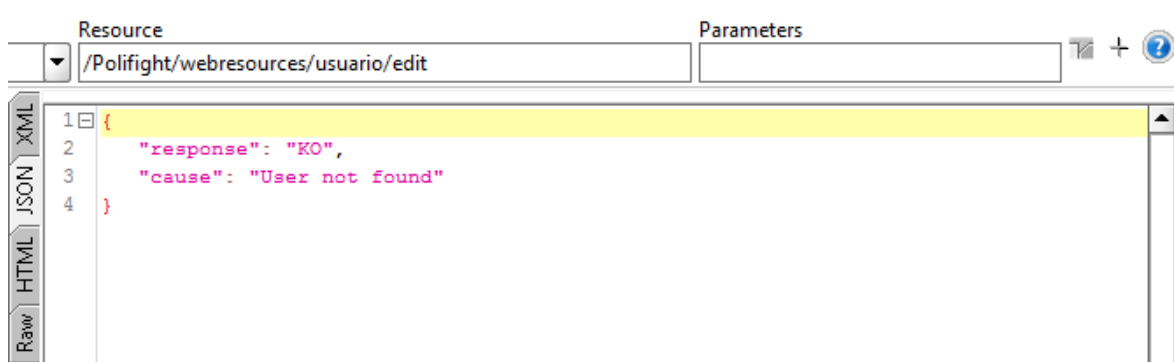
## Satisfactorio:

Si la modificación al atributo fue satisfactorio, se responderá con el campo "response": "OK":



### No satisfactorio:

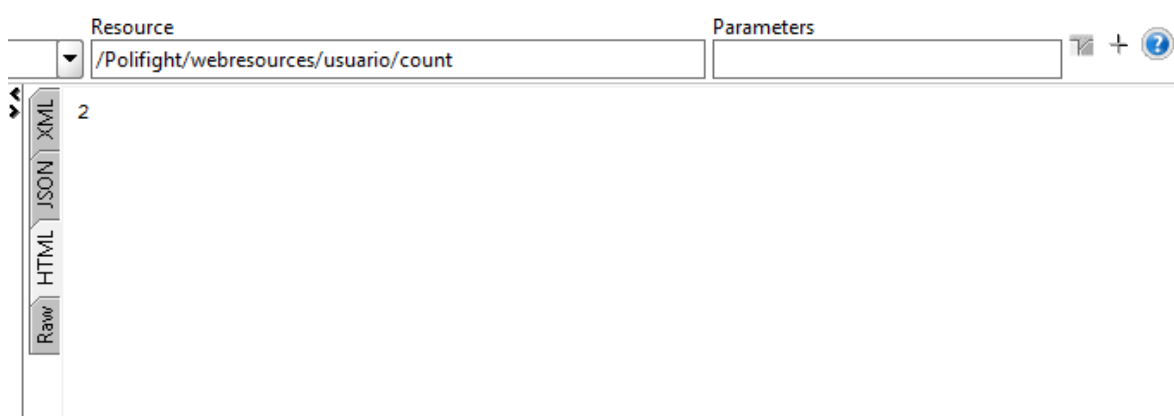
Si el servicio no puede procesar la solicitud se responderá un KO en el campo “response”. En este caso se incluirá el campo “cause” donde se indicará la causa del error



### count

EndPoint: `/Polifight/webresources/rolusuario/count`

Este método retorna la cantidad de datos en formato HTML que se encuentran en la DB. Se debe consumir por método GET directamente sin parámetros:

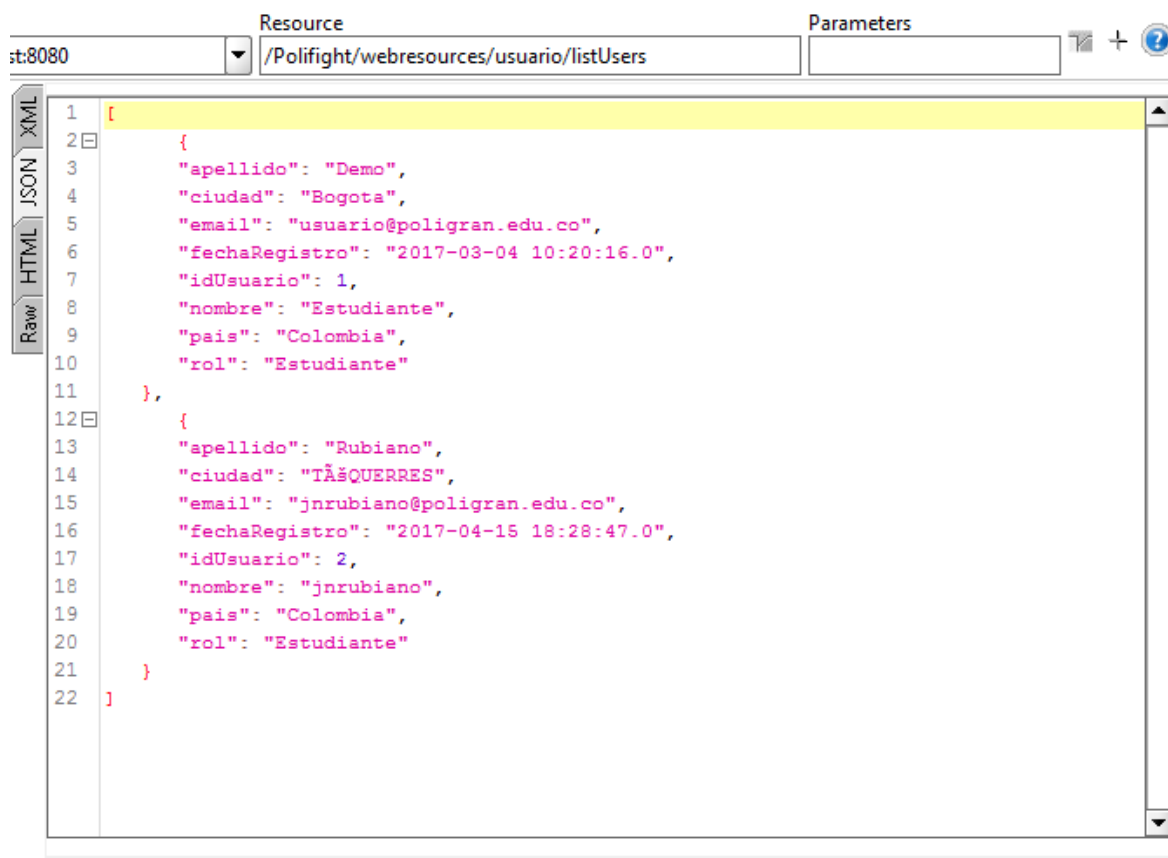




## listUsers

EndPoint: /Polifight/webresources/usuario/listUsers

Se debe realizar el consumo por método GET. Se retorna una lista en formato JSON con todos los datos contenidos en la base de datos:



Si no se encuentran datos para los usuarios, se mostrará un mensaje JSON con los elementos 'response': 'OK' y 'cause' donde se indicará que no se encuentran datos en la base de datos.

## login

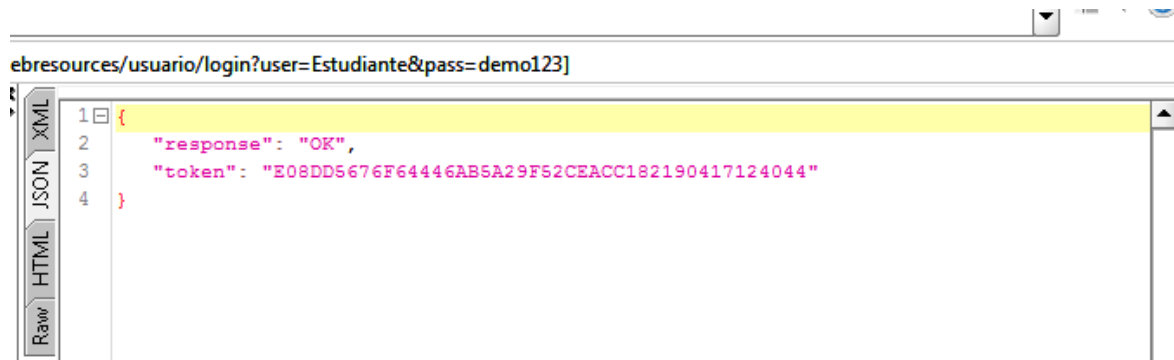
EndPoint: <http://IP:Puerto/Polifight/webresources/usuario/login>

Servicio que recibe un usuario y una contraseña y retorna, en caso exitoso, un campo “response” con un “OK” y un campo “token” donde se especifica el String de tipo Token que será enviado en el consumo de los demás servicios que lo soliciten. Este servicio debe ser invocado por método POST.

Las variables que se deben enviar son:

- user: String que indica el usuario
- pass: String que indica la contraseña asociada al usuario. Este String debe ser enviado cifrado.

## Satisfactorio

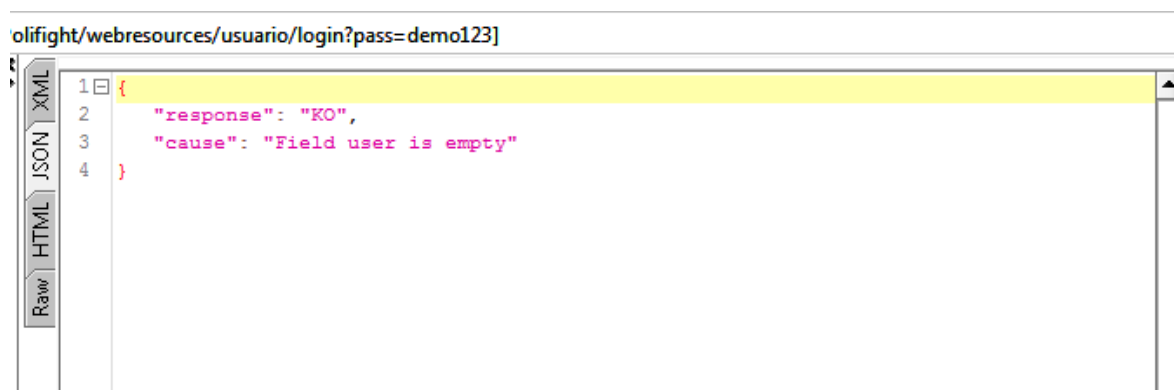


## No satisfactorio

Se identifican 4 tipo de casos donde puede suceder un error:

### *Campo usuario vacio*

Si el campo “user” se envía vacío, el servicio responderá el siguiente mensaje:



### *Campo contraseña vacio*

Si el campo “pass” se envía vacío, el servicio responderá el siguiente mensa:

```
[/Polifight/webresources/usuario/login?user=Estudiante]

1 {
2   "response": "KO",
3   "cause": "Field pass is empty"
4 }
```

### *Sin parámetros de entrada*

Si el servicio es invocado y no se envían los dos parámetros indicados anteriormente, el servicio responderá el siguiente mensaje:

```
[/Polifight/webresources/usuario/login]

1 {
2   "response": "KO",
3   "cause": "Field pass and user is empty"
4 }
```

### *El usuario no está registrado*

Si se envía el usuario y la contraseña y éstos no son encontrados en la DB, el servicio responderá el siguiente mensaje:

```
Polifight/webresources/usuario/login?user=jnrubiano&pass=12345]

1 {
2   "response": "OK",
3   "cause": "Unregistered user"
4 }
```

Es importante recalcar, que actualmente el servicio no indica si existe el usuario pero la contraseña está inválida, en este caso se responderá el mensaje anterior.