



# Práctica 1: Servicios RESTful

Sistemas Orientados a Servicios

---

García Tejada, Loreto  
Gutiérrez Martín, María

## Tabla de contenido

<b>1. Diseño del servicio RESTful .....</b>	<b>2</b>
<b>1.1. Entrypoint .....</b>	<b>2</b>
<b>1.2. Recursos .....</b>	<b>2</b>
<b>1.3. Diagrama Entidad-Relación .....</b>	<b>2</b>
<b>1.4. Resumen del diseño del servicio .....</b>	<b>3</b>
1.4.1. Usuario .....	3
1.4.2. Usuario (concreto) .....	3
1.4.3. Cuenta bancaria .....	4
1.4.4. Transferencia.....	4
1.4.5. Retirada .....	5
1.4.6. Movimientos .....	6
1.4.7. Información para el móvil de un usuario .....	6
<b>2. Capturas de la ejecución de las operaciones.....</b>	<b>7</b>
<b>2.1. Usuario.....</b>	<b>7</b>
2.1.1. Crear un perfil de usuario de banco.....	7
2.1.2. Listado de todos los usuarios (y su saldo).....	8
<b>2.2. Usuario (concreto) .....</b>	<b>9</b>
2.2.1. Ver los datos básicos de un usuario.....	9
2.2.2. Cambiar datos básicos del usuario.....	10
2.2.3. Borrar todos los datos de un usuario.....	10
<b>2.3. Cuenta bancaria.....</b>	<b>12</b>
2.3.1. Crear una cuenta bancaria para un usuario.....	12
2.3.2. Eliminar la cuenta bancaria.....	12
<b>2.4. Transferencia.....</b>	<b>14</b>
2.4.1. Realizar una transferencia entre cuentas .....	14
2.4.2. Eliminar una transferencia .....	15
2.4.3. Listado de todas las transferencias de una cuenta .....	16
<b>2.5. Retiradas .....</b>	<b>18</b>
2.5.1. Realizar una retirada de efectivo .....	18
2.5.2. Listado de las retiradas de efectivo realizadas por un usuario.....	19
<b>2.6. Movimientos .....</b>	<b>21</b>
2.6.1. Listado de todos los movimientos .....	21
<b>2.7. Información para el móvil de un usuario.....</b>	<b>22</b>
2.7.1. Mostrar los datos básicos de un cliente, sus cuentas y su saldo, los últimos 10 movimientos realizados en todas sus cuentas .....	22
<b>3. Capturas de la ejecución del cliente .....</b>	<b>23</b>

## 1. Diseño del servicio RESTful

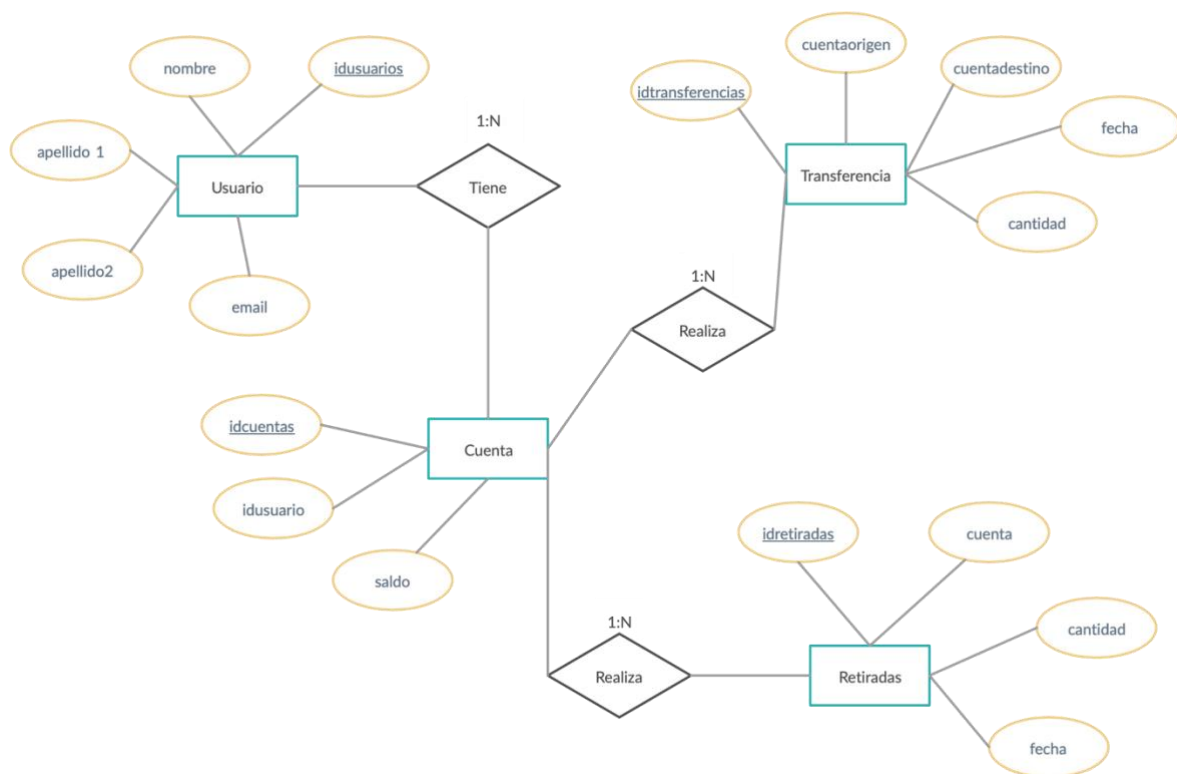
### 1.1. Entrypoint

<http://localhost:8080/Banco/api>

### 1.2. Recursos

1. Usuario
2. Usuario (concreto)
3. Cuenta bancaria
4. Transferencia
5. Retiradas de efectivo
6. Movimiento
7. Información para el móvil de un usuario

### 1.3. Diagrama Entidad-Relación



## 1.4. Resumen del diseño del servicio

### 1.4.1. Usuario

<b>URI</b>	Banco/api/usuarios
<b>Descripción</b>	Crear un perfil de usuario de banco
<b>Método</b>	POST
<b>Cadena de consulta</b>	-
<b>Cuerpo</b>	POX (usuarios/id_usuario + XML)
<b>Devuelve</b>	201 Created y location=usuarios/{id_usuarios} 406 Not Acceptable 500 Internal Server Error

<b>URI</b>	Banco/api/usuarios
<b>Descripción</b>	Listado de todos los usuarios (y <i>su saldo</i> )
<b>Método</b>	GET
<b>Cadena de consulta</b>	cash=value start=number end=number
<b>Cuerpo</b>	-
<b>Devuelve</b>	200 OK y POX(/usuarios) 400 Bad request 500 Internal Server Error

### 1.4.2. Usuario (concreto)

<b>URI</b>	Banco/api/usuarios/id_usuario
<b>Descripción</b>	Ver los datos básicos de un usuario
<b>Método</b>	GET
<b>Cadena de consulta</b>	-
<b>Cuerpo</b>	-
<b>Devuelve</b>	200 OK y POX (/usuarios/{id_usuario} + JSON/XML) 404 Not found 500 Internal Server Error

<b>URI</b>	Banco/api/usuarios/id_usuario
<b>Descripción</b>	Cambiar datos básicos del usuario
<b>Método</b>	PUT
<b>Cadena de consulta</b>	
<b>Cuerpo</b>	usuarios/id_usuario + XML
<b>Devuelve</b>	200 OK y POX (/usuarios/{id_usuario} + XML) 404 Not found 500 Internal Server Error

<b>URI</b>	Banco/api/usuarios/id_usuario
<b>Descripción</b>	Borrar todos los datos de un usuario
<b>Método</b>	DELETE
<b>Cadena de consulta</b>	-
<b>Cuerpo</b>	-
<b>Devuelve</b>	204 No Content 404 Not found 406 Not Acceptable 500 Internal Server Error

#### 1.4.3. Cuenta bancaria

<b>URI</b>	Banco/api/cuentas
<b>Descripción</b>	Crear una cuenta bancaria para un usuario
<b>Método</b>	POST
<b>Cadena de consulta</b>	-
<b>Cuerpo</b>	usuarios/id_usuario/cuentas/id_cuenta + XML
<b>Devuelve</b>	201 Created y location=cuentas/{id_cuenta} 404 Not found 500 Internal Server Error

<b>URI</b>	Banco/api/cuentas/id_cuenta
<b>Descripción</b>	Eliminar la cuenta bancaria
<b>Método</b>	DELETE
<b>Cadena de consulta</b>	-
<b>Cuerpo</b>	-
<b>Devuelve</b>	204 No Content 404 Not found 500 Internal Server Error

#### 1.4.4. Transferencia

<b>URI</b>	Banco/api/transferencias
<b>Descripción</b>	Realizar una transferencia entre cuentas
<b>Método</b>	POST
<b>Cadena de consulta</b>	-
<b>Cuerpo</b>	transferencias/id_transferencia + XML
<b>Devuelve</b>	201 Created y location=transferencias/{id_transferencia} 404 Not found 406 Not Acceptable 500 Internal Server Error

<b>URI</b>	Banco/api/transferencias/id_transferencia
<b>Descripción</b>	Eliminar una transferencia
<b>Método</b>	DELETE
<b>Cadena de consulta</b>	-
<b>Cuerpo</b>	-
<b>Devuelve</b>	204 No Content 404 Not found 500 Internal Sever Error

<b>URI</b>	Banco/api/cuentas/id_cuenta/transferencias
<b>Descripción</b>	Listado de todas las transferencias de una cuenta
<b>Método</b>	GET
<b>Cadena de consulta</b>	date=date start=primerElemento end=ultimoElemento
<b>Cuerpo</b>	-
<b>Devuelve</b>	200 OK y POX(/cuentas/{id_cuenta}/transferencias+JSON/XML) 400 Bad request 404 Not found 500 Internal Sever Error

#### 1.4.5. Retirada

<b>URI</b>	Banco/api/retiradas
<b>Descripción</b>	Realizar una retirada de efectivo
<b>Método</b>	POST
<b>Cadena de consulta</b>	
<b>Cuerpo</b>	retiradas/id_retirada + XML
<b>Devuelve</b>	201 Created y location=retiradas/{id_retirada} 404 Not found 406 Not Acceptable 500 Internal Server Error

<b>URI</b>	Banco/api/usuarios/id_usuario/retiradas
<b>Descripción</b>	Listado de las retiradas de efectivo realizadas por un usuario
<b>Método</b>	GET
<b>Cadena de consulta</b>	cash=value date=date
<b>Cuerpo</b>	-
<b>Devuelve</b>	200 OK y POX(/usuarios/{id_usuario}/retiradas) 400 Bad request 404 Not found 500 Internal Server Error

#### 1.4.6. Movimientos

<b>URI</b>	Banco/api/usuarios/id_usuario/movimientos
<b>Descripción</b>	Listado de todos los movimientos
<b>Método</b>	GET
<b>Cadena de consulta</b>	cash=value date=date
<b>Cuerpo</b>	-
<b>Devuelve</b>	200 OK y POX(/usuarios/{id_usuario}/movimientos) 400 Bad request 404 Not found 500 Internal Server Error

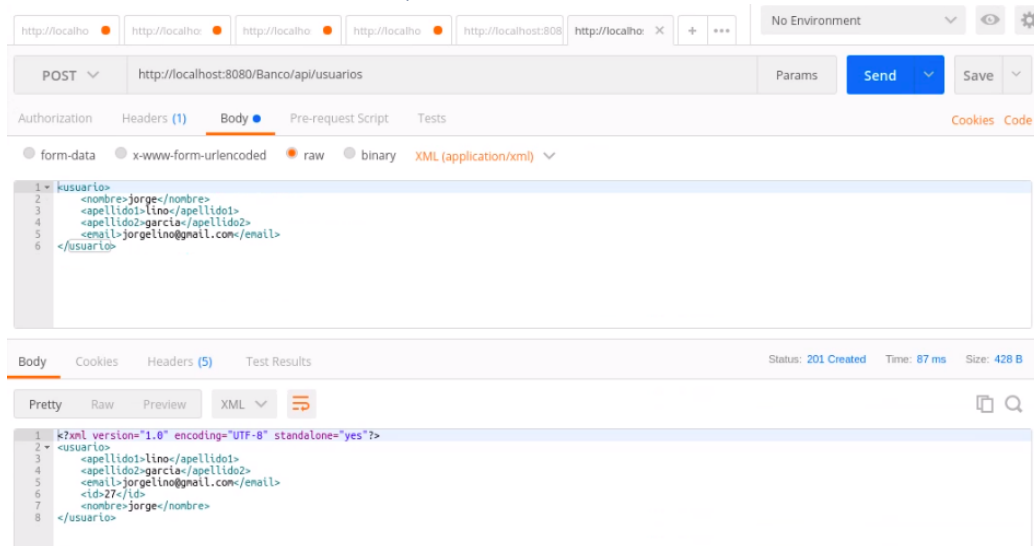
#### 1.4.7. Información para el móvil de un usuario

<b>URI</b>	Banco/api/usuarios/id_usuario/info
<b>Descripción</b>	Mostrar los datos básicos de un cliente, sus cuentas y su saldo, los últimos 10 movimientos realizados en todas sus cuentas
<b>Método</b>	GET
<b>Cadena de consulta</b>	-
<b>Cuerpo</b>	-
<b>Devuelve</b>	200 OK y POX(/usuarios/{id_usuario}/info) 400 Bad request 404 Not found 500 Internal Sever Error

## 2. Capturas de la ejecución de las operaciones

### 2.1. Usuario

#### 2.1.1. Crear un perfil de usuario de banco



The screenshot shows a Postman interface with a POST request to `http://localhost:8080/Banco/api/usuarios`. The request body is an XML document:

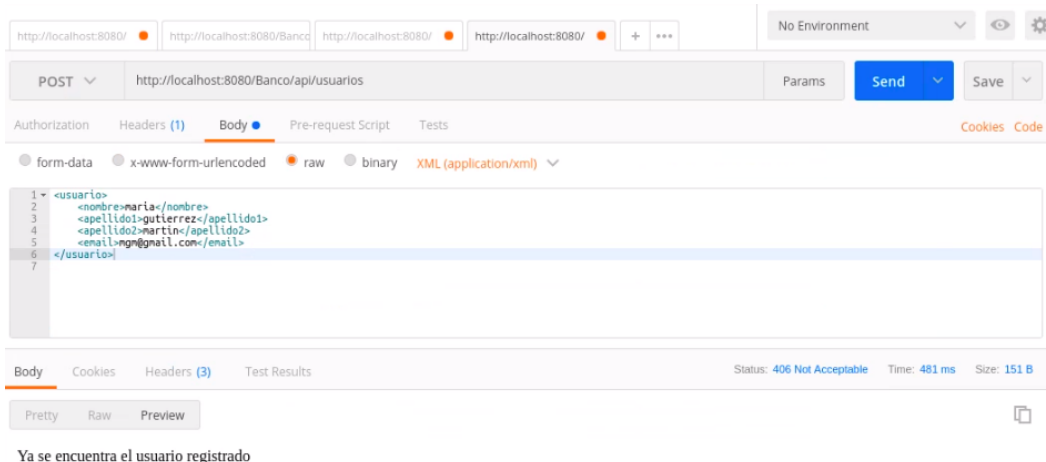
```
1 <usuario>
2   <nombre>jorge</nombre>
3   <apellido1>lino</apellido1>
4   <apellido2>garcia</apellido2>
5   <email>jorgelino@gmail.com</email>
6 </usuario>
```

The response is also in XML format, showing a successful status 201:

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <usuario>
3   <apellido1>lino</apellido1>
4   <apellido2>garcia</apellido2>
5   <email>jorgelino@gmail.com</email>
6   <id>27</id>
7   <nombre>jorge</nombre>
8 </usuario>
```

Status: 201 Created, Time: 87 ms, Size: 428 B.

#### ○ Si el usuario ya existe



The screenshot shows a POST request to `http://localhost:8080/Banco/api/usuarios` with the following XML body:

```
1 <usuario>
2   <nombre>maria</nombre>
3   <apellido1>gutierrez</apellido1>
4   <apellido2>martin</apellido2>
5   <email>ngn@gmail.com</email>
6 </usuario>
```

The response is a 406 Not Acceptable status:

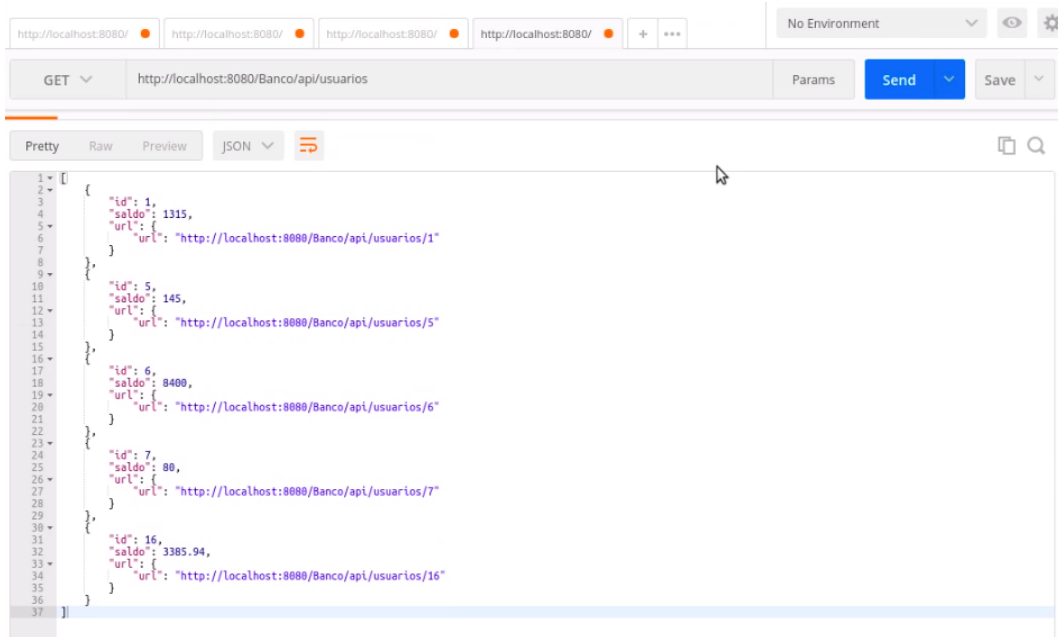
```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <error>
3   <code>406</code>
4   <message>Not Acceptable</message>
5 </error>
```

Status: 406 Not Acceptable, Time: 481 ms, Size: 151 B.

Ya se encuentra el usuario registrado



### 2.1.2. Listado de todos los usuarios (y su saldo)

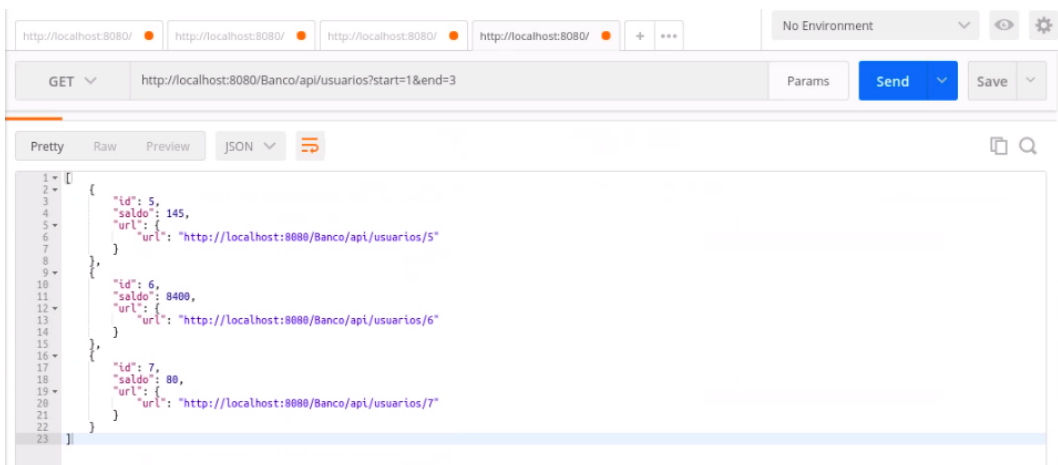


GET http://localhost:8080/Banco/api/usuarios

```

1- [
2-   {
3-     "id": 1,
4-     "saldo": 1315,
5-     "url": {
6-       "url": "http://localhost:8080/Banco/api/usuarios/1"
7-     }
8-   },
9-   {
10-    "id": 5,
11-    "saldo": 145,
12-    "url": {
13-      "url": "http://localhost:8080/Banco/api/usuarios/5"
14-    }
15-   },
16-   {
17-    "id": 6,
18-    "saldo": 8400,
19-    "url": {
20-      "url": "http://localhost:8080/Banco/api/usuarios/6"
21-    }
22-   },
23-   {
24-    "id": 7,
25-    "saldo": 80,
26-    "url": {
27-      "url": "http://localhost:8080/Banco/api/usuarios/7"
28-    }
29-   },
30-   {
31-    "id": 16,
32-    "saldo": 3385.94,
33-    "url": {
34-      "url": "http://localhost:8080/Banco/api/usuarios/16"
35-    }
36-   }
37- ]
  
```

- Limitar número de usuarios



GET http://localhost:8080/Banco/api/usuarios?start=1&end=3

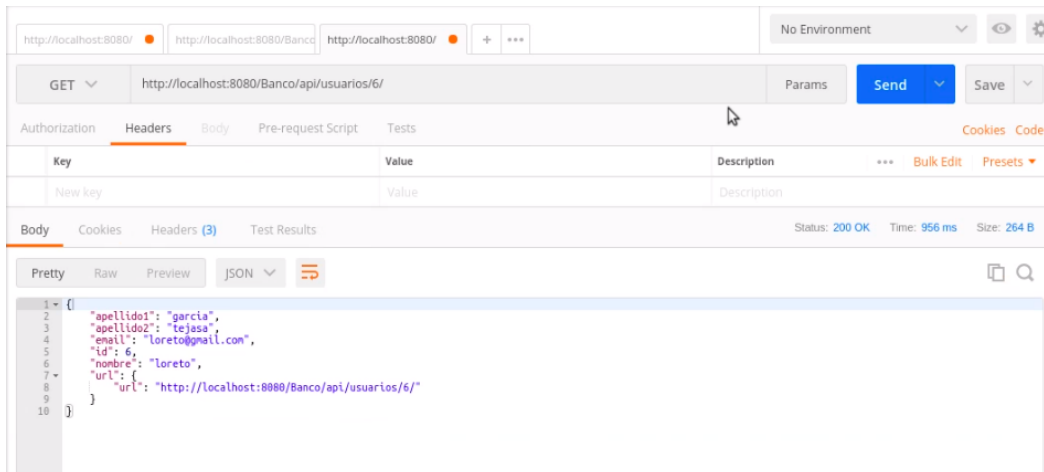
```

1- [
2-   {
3-     "id": 5,
4-     "saldo": 145,
5-     "url": {
6-       "url": "http://localhost:8080/Banco/api/usuarios/5"
7-     }
8-   },
9-   {
10-    "id": 6,
11-    "saldo": 8400,
12-    "url": {
13-      "url": "http://localhost:8080/Banco/api/usuarios/6"
14-    }
15-   },
16-   {
17-    "id": 7,
18-    "saldo": 80,
19-    "url": {
20-      "url": "http://localhost:8080/Banco/api/usuarios/7"
21-    }
22-   }
23- ]
  
```

## 2.2. Usuario (concreto)

### 2.2.1. Ver los datos básicos de un usuario

#### ○ Json

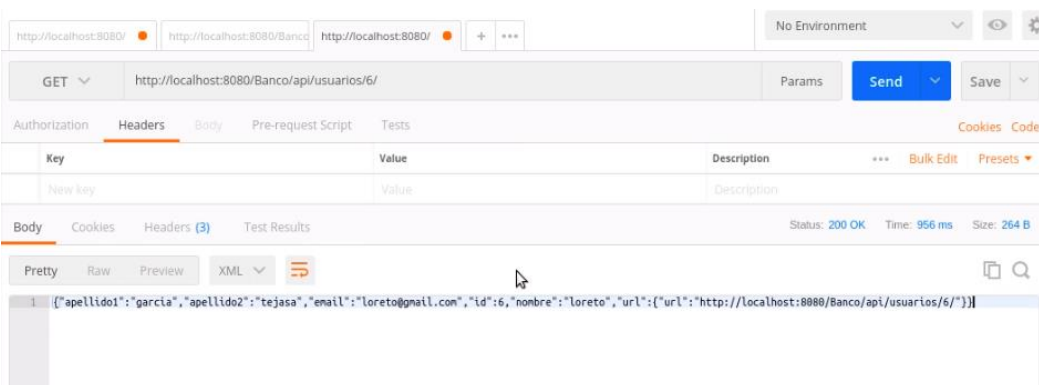


The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/Banco/api/usuarios/6/`
- Method:** GET
- Status:** 200 OK
- Time:** 956 ms
- Size:** 264 B
- Body (JSON):**

```
{
  "apellido1": "garcia",
  "apellido2": "tejasas",
  "email": "loreto@gmail.com",
  "id": 6,
  "nombre": "loreto",
  "url": {
    "url": "http://localhost:8080/Banco/api/usuarios/6/"
  }
}
```

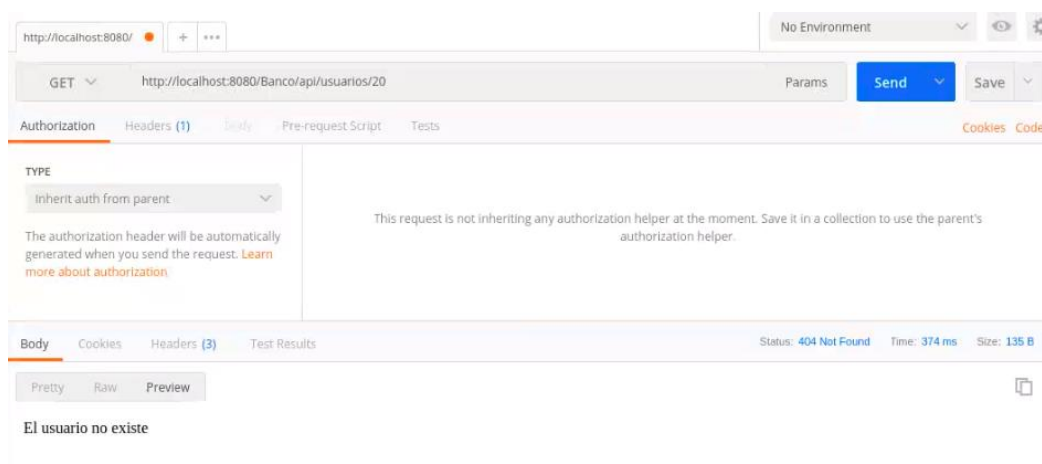
#### ○ Xml



The screenshot shows the same REST client interface, but the response body is displayed in XML format:

```
{ "apellido1": "garcia", "apellido2": "tejasas", "email": "loreto@gmail.com", "id": 6, "nombre": "loreto", "url": { "url": "http://localhost:8080/Banco/api/usuarios/6/" } }
```

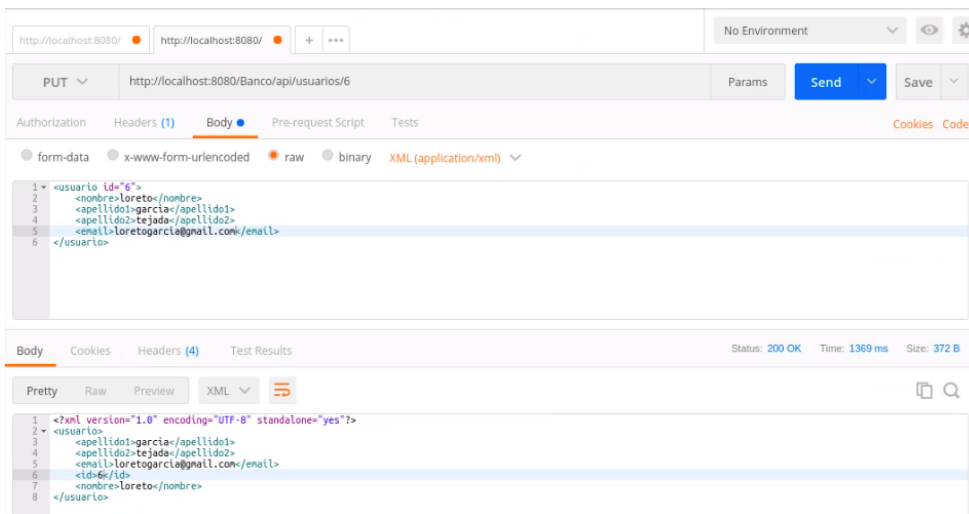
#### ○ Si el usuario no existe



The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/Banco/api/usuarios/20`
- Method:** GET
- Status:** 404 Not Found
- Time:** 374 ms
- Size:** 135 B
- Body (Text):** El usuario no existe

## 2.2.2. Cambiar datos básicos del usuario



PUT `http://localhost:8080/Banco/api/usuarios/6` Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary XML (application/xml)

```

1 <usuario id="6">
2   <nombre>loredo</nombre>
3   <apellido1>garcia</apellido1>
4   <apellido2>tejada</apellido2>
5   <email>loredogarcia@gmail.com</email>
6 </usuario>

```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 1369 ms Size: 372 B

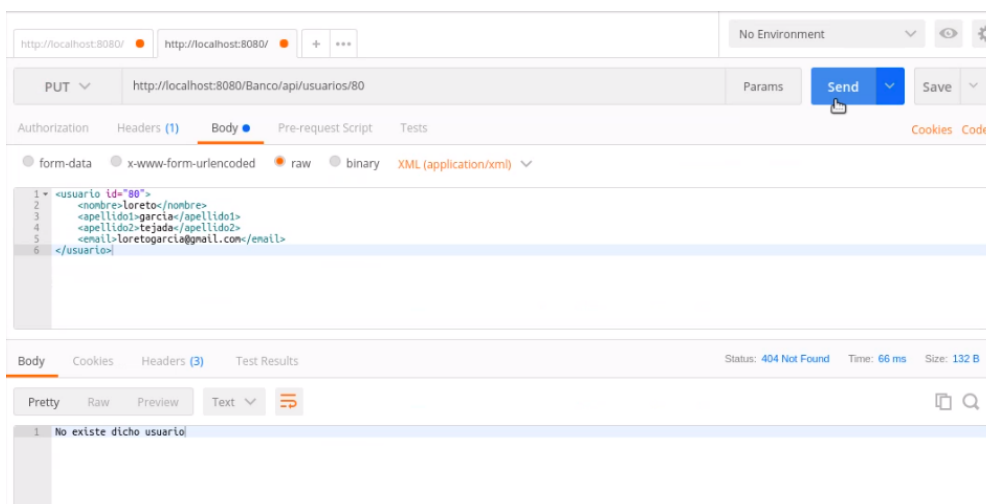
Pretty Raw Preview XML

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <usuario>
3   <apellido1>garcia</apellido1>
4   <apellido2>tejada</apellido2>
5   <email>loredogarcia@gmail.com</email>
6   <id>6</id>
7   <nombre>loredo</nombre>
8 </usuario>

```

- Si el usuario no existe



PUT `http://localhost:8080/Banco/api/usuarios/80` Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary XML (application/xml)

```

1 <usuario id="80">
2   <nombre>loredo</nombre>
3   <apellido1>garcia</apellido1>
4   <apellido2>tejada</apellido2>
5   <email>loredogarcia@gmail.com</email>
6 </usuario>

```

Body Cookies Headers (3) Test Results Status: 404 Not Found Time: 66 ms Size: 132 B

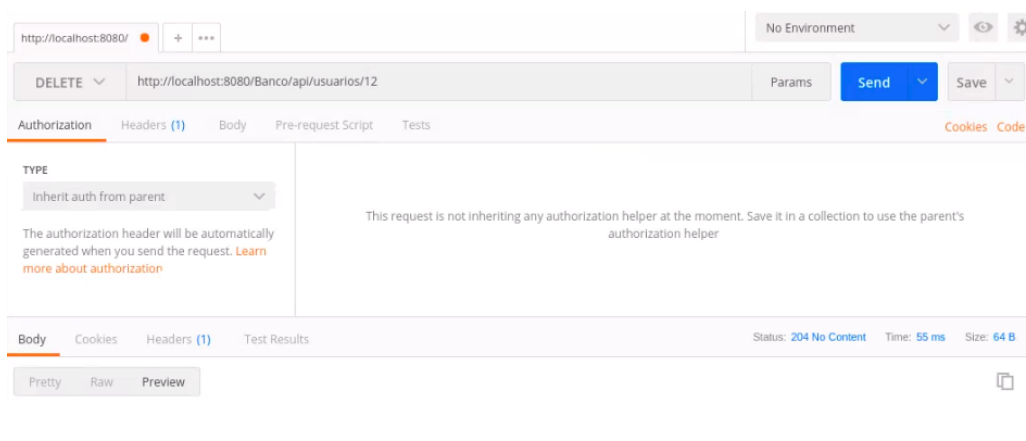
Pretty Raw Preview Text

```

1 No existe dicho usuario

```

## 2.2.3. Borrar todos los datos de un usuario



DELETE `http://localhost:8080/Banco/api/usuarios/12` Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

TYPE Inherit auth from parent

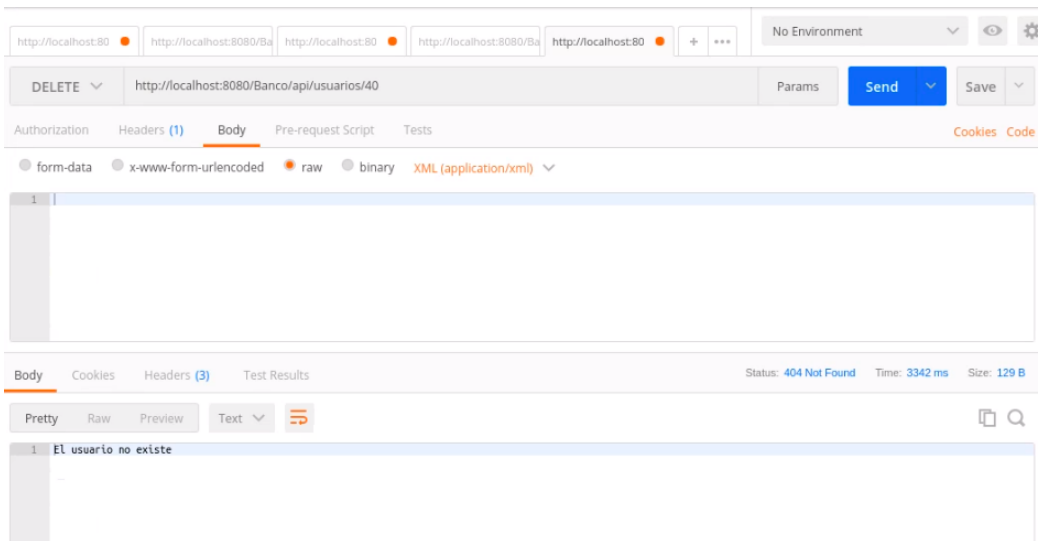
The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper

Body Cookies Headers (1) Test Results Status: 204 No Content Time: 55 ms Size: 64 B

Pretty Raw Preview

○ Si el usuario no existe



DELETE `http://localhost:8080/Banco/api/usuarios/40` Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary XML (application/xml)

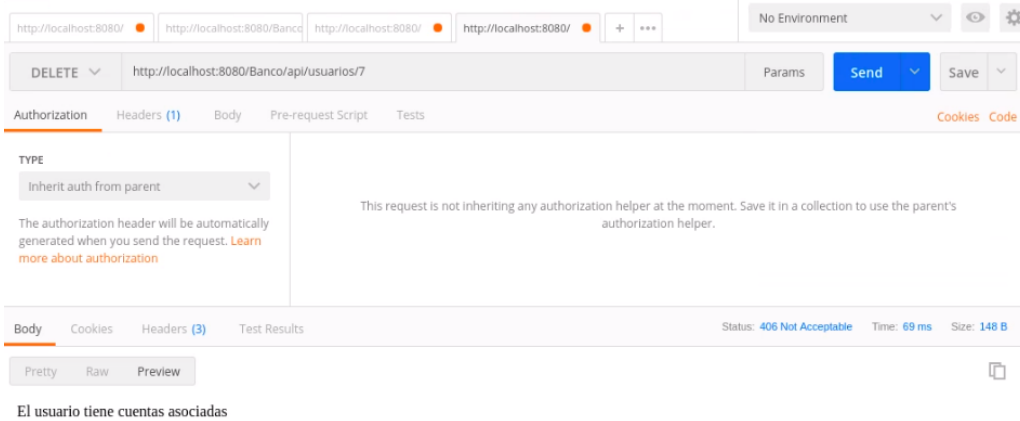
1

Body Cookies Headers (3) Test Results Status: 404 Not Found Time: 3342 ms Size: 129 B

Pretty Raw Preview Text

1 El usuario no existe

○ Si el usuario tiene cuentas abiertas



DELETE `http://localhost:8080/Banco/api/usuarios/7` Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

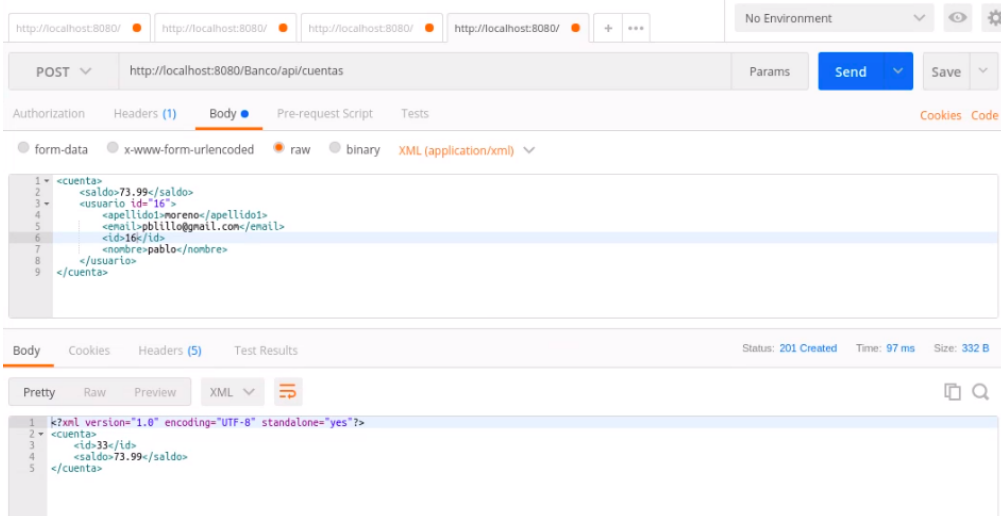
Body Cookies Headers (3) Test Results Status: 406 Not Acceptable Time: 69 ms Size: 148 B

Pretty Raw Preview

El usuario tiene cuentas asociadas

## 2.3. Cuenta bancaria

### 2.3.1. Crear una cuenta bancaria para un usuario

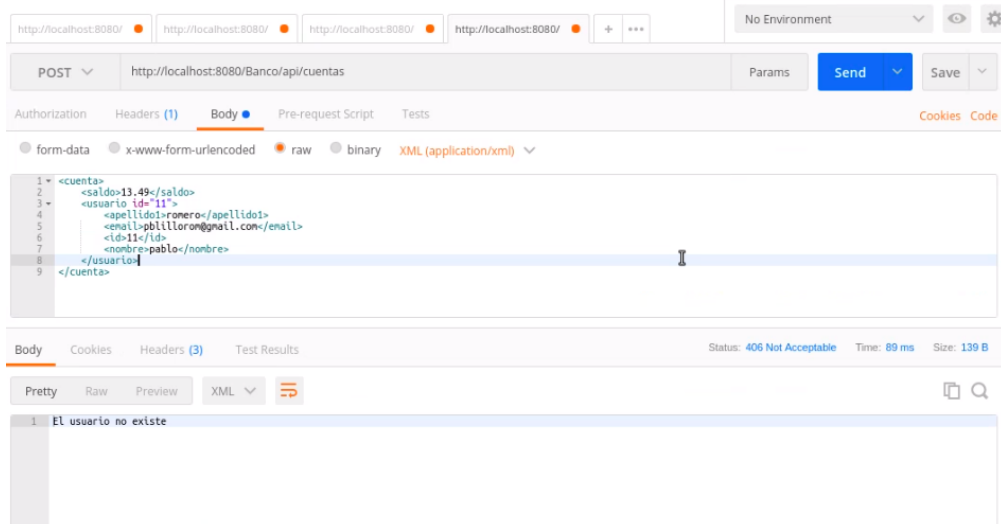


The screenshot shows a Postman interface with a POST request to `http://localhost:8080/Banco/api/cuentas`. The request body is an XML document:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cuanta>
  <saldo>73.99</saldo>
  <usuario id="16">
    <apellido>moreno</apellido>
    <email>pbillor@gmail.com</email>
    <id>16</id>
    <nombre>pablo</nombre>
  </usuario>
</cuanta>
```

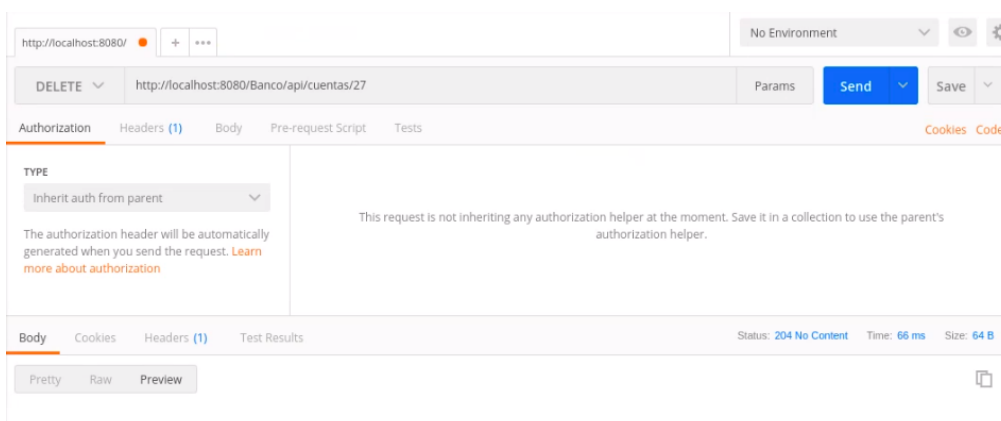
The response status is **201 Created**, with a time of 97 ms and a size of 332 B. The response body is the same XML document as the request.

#### ○ Si el usuario no existe



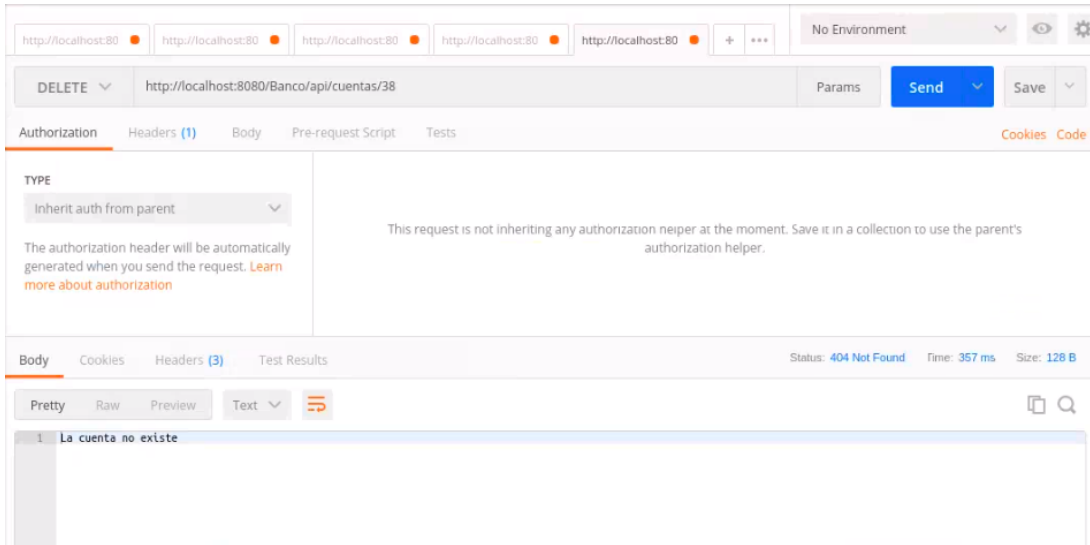
The screenshot shows a POST request to `http://localhost:8080/Banco/api/cuentas` with an XML body for a user with ID 11. The response status is **406 Not Acceptable**, with a time of 89 ms and a size of 139 B. The response body is the text: `El usuario no existe`.

### 2.3.2. Eliminar la cuenta bancaria



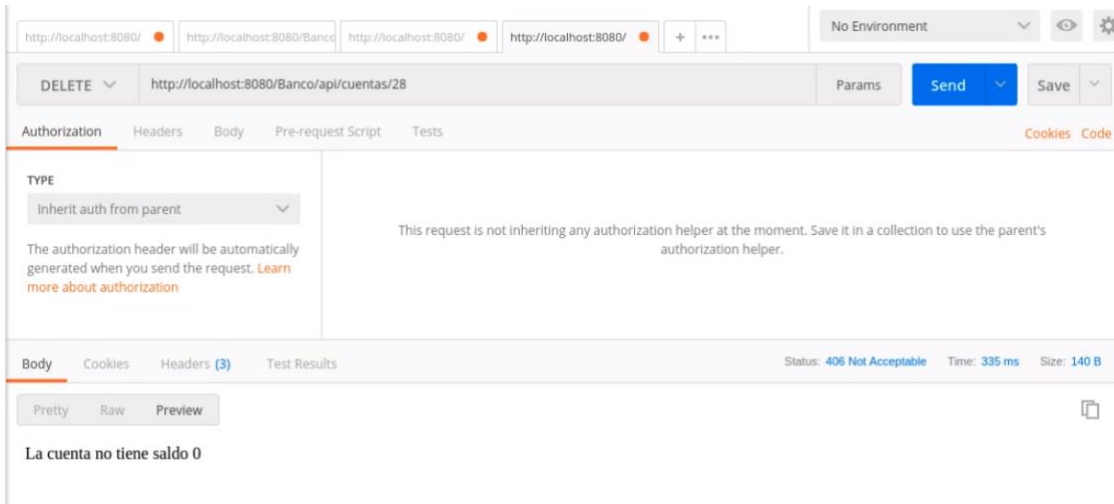
The screenshot shows a DELETE request to `http://localhost:8080/Banco/api/cuentas/27`. The response status is **204 No Content**, with a time of 66 ms and a size of 64 B. The response body is empty.

○ Si la cuenta no existe



The screenshot shows a Postman interface for a DELETE request to `http://localhost:8080/Banco/api/cuentas/38`. The request is configured with "Inherit auth from parent" and no body. The response status is **404 Not Found** with a time of 357 ms and a size of 128 B. The response body, displayed in "Text" view, contains the message: **La cuenta no existe**.

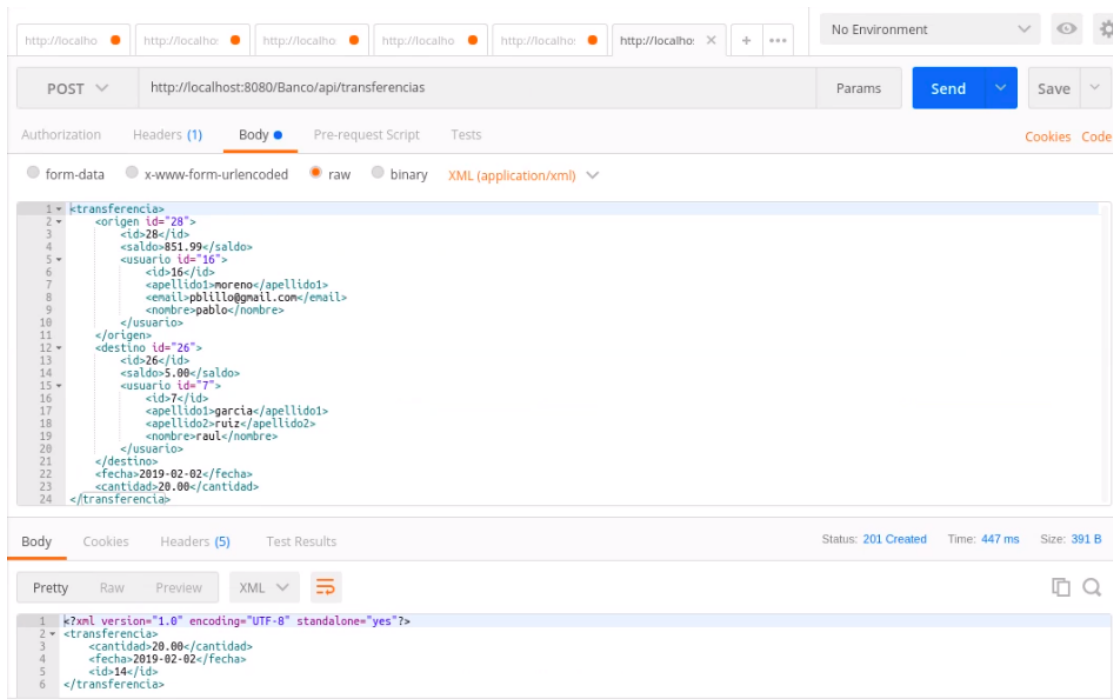
○ Si la cuenta no tiene saldo cero



The screenshot shows a Postman interface for a DELETE request to `http://localhost:8080/Banco/api/cuentas/28`. The request is configured with "Inherit auth from parent" and no body. The response status is **406 Not Acceptable** with a time of 335 ms and a size of 140 B. The response body, displayed in "Text" view, contains the message: **La cuenta no tiene saldo 0**.

## 2.4. Transferencia

### 2.4.1. Realizar una transferencia entre cuentas



POST http://localhost:8080/Banco/api/transferencias

Body (XML):

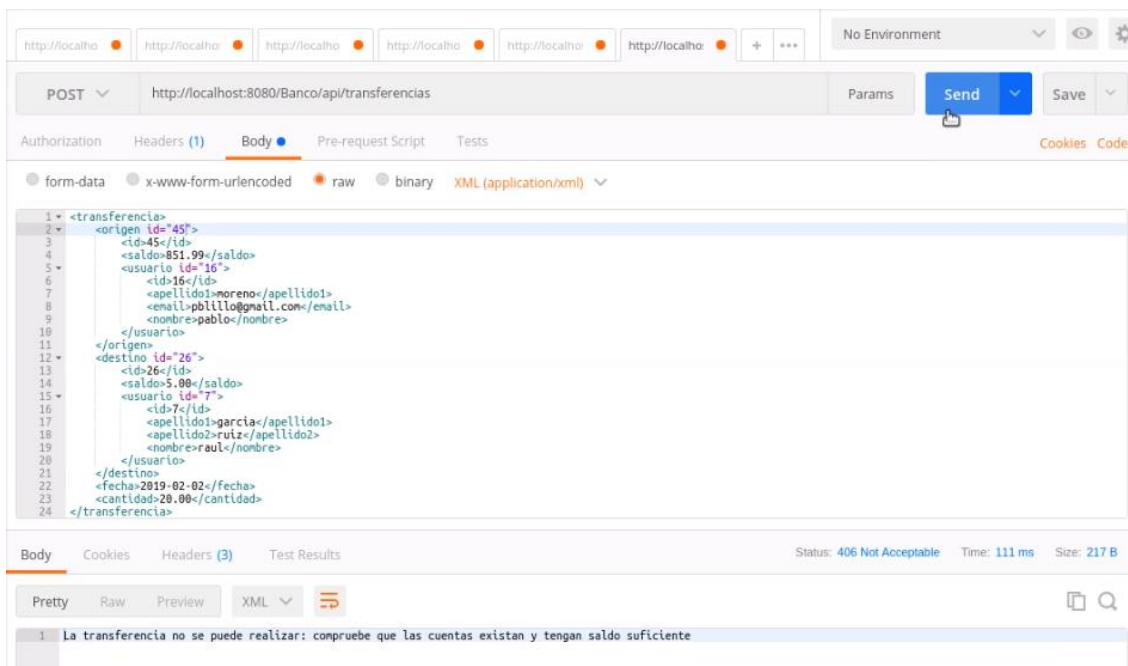
```
<transferencia>
  <origen id="28">
    <id>28</id>
    <saldo>851.99</saldo>
    <usuario id="16">
      <id>16</id>
      <apellido1>noreno</apellido1>
      <email>pblillo@gmail.com</email>
      <nombre>pablo</nombre>
    </usuario>
  </origen>
  <destino id="26">
    <id>26</id>
    <saldo>5.00</saldo>
    <usuario id="7">
      <id>7</id>
      <apellido1>garcia</apellido1>
      <apellido2>ruiz</apellido2>
      <nombre>raul</nombre>
    </usuario>
  </destino>
  <fecha>2019-02-02</fecha>
  <cantidad>20.00</cantidad>
</transferencia>
```

Status: 201 Created Time: 447 ms Size: 391 B

Body (Pretty):

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<transferencia>
  <cantidad>20.00</cantidad>
  <fecha>2019-02-02</fecha>
  <id>14</id>
</transferencia>
```

- Si la cuenta origen no existe



POST http://localhost:8080/Banco/api/transferencias

Body (XML):

```
<transferencia>
  <origen id="45">
    <id>45</id>
    <saldo>851.99</saldo>
    <usuario id="16">
      <id>16</id>
      <apellido1>noreno</apellido1>
      <email>pblillo@gmail.com</email>
      <nombre>pablo</nombre>
    </usuario>
  </origen>
  <destino id="26">
    <id>26</id>
    <saldo>5.00</saldo>
    <usuario id="7">
      <id>7</id>
      <apellido1>garcia</apellido1>
      <apellido2>ruiz</apellido2>
      <nombre>raul</nombre>
    </usuario>
  </destino>
  <fecha>2019-02-02</fecha>
  <cantidad>20.00</cantidad>
</transferencia>
```

Status: 406 Not Acceptable Time: 111 ms Size: 217 B

Body (Pretty):

```
1 La transferencia no se puede realizar: compruebe que las cuentas existan y tengan saldo suficiente
```

- Si no hay saldo suficiente en la cuenta origen

POST http://localhost:8080/Banco/api/transferencias

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary XML (application/xml)

```

1 <transferencia>
2   <origen id="27">
3     <id>27</id>
4     <saldo>0.0</saldo>
5     <usuario id="6">
6       <id>6</id>
7       <apellido1>garcia</apellido1>
8       <apellido2>tejada</apellido2>
9       <email>loretogarcia@gmail.com</email>
10      <nombre>loreto</nombre>
11    </usuario>
12  </origen>
13  <destino id="26">
14    <id>26</id>
15    <saldo>5.00</saldo>
16    <usuario id="7">
17      <id>7</id>
18      <apellido1>garcia</apellido1>
19      <apellido2>raul</apellido2>
20      <nombre>raul</nombre>
21    </usuario>
22  </destino>
23  <fecha>2019-02-02</fecha>
24  <cantidad>20.00</cantidad>
25 </transferencia>

```

Status: 406 Not Acceptable Time: 381 ms Size: 217 B

Body Cookies Headers (3) Test Results

Pretty Raw Preview XML

1 La transferencia no se puede realizar: compruebe que las cuentas existan y tengan saldo suficiente

## 2.4.2. Eliminar una transferencia

DELETE http://localhost:8080/Banco/api/transferencias/4

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Status: 204 No Content Time: 61 ms Size: 64 B

Body Cookies Headers (1) Test Results

Pretty Raw Preview

- Si la transferencia no existe

DELETE http://localhost:8080/Banco/api/transferencias/22

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Status: 404 Not Found Time: 33 ms Size: 138 B

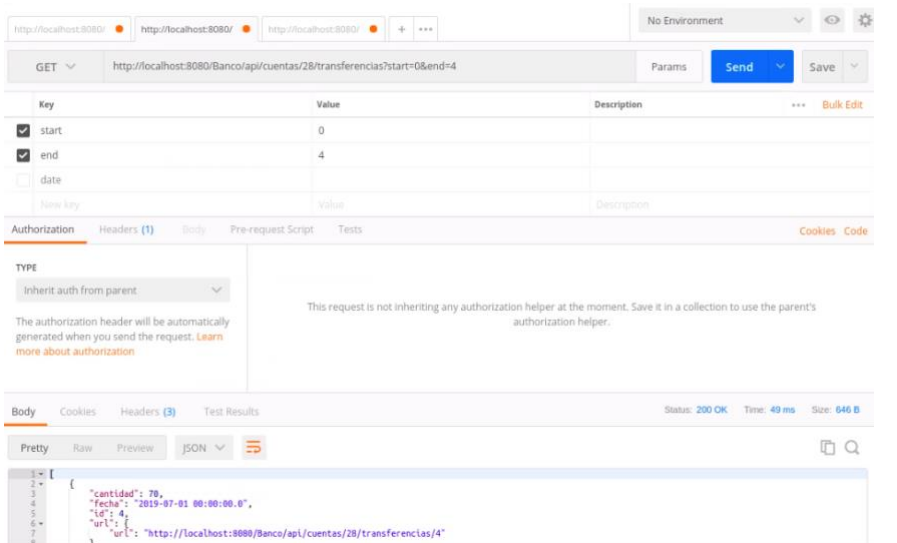
Body Cookies Headers (3) Test Results

Pretty Raw Preview Text

1 No existe dicha transferencia



### 2.4.3. Listado de todas las transferencias de una cuenta



Key | Value | Description

start	0	
end	4	
date		

Authorization | Headers (1) | Body | Pre-request Script | Tests

TYPE: Inherit auth from parent

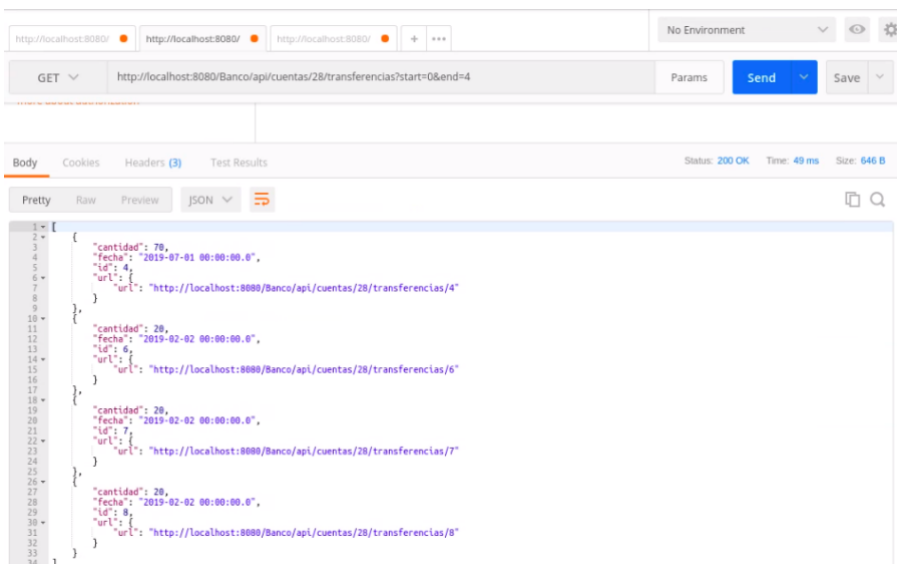
The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Status: 200 OK Time: 49 ms Size: 646 B

Body | Cookies | Headers (3) | Test Results

Pretty | Raw | Preview | JSON

```
1- [
2- {
3-   "cantidad": 70,
4-   "fecha": "2019-07-01 00:00:00.0",
5-   "id": 4,
6-   "url": {
7-     "url": "http://localhost:8080/Banco/api/cuentas/28/transferencias/4"
8-   }
9- }
10- ]
```



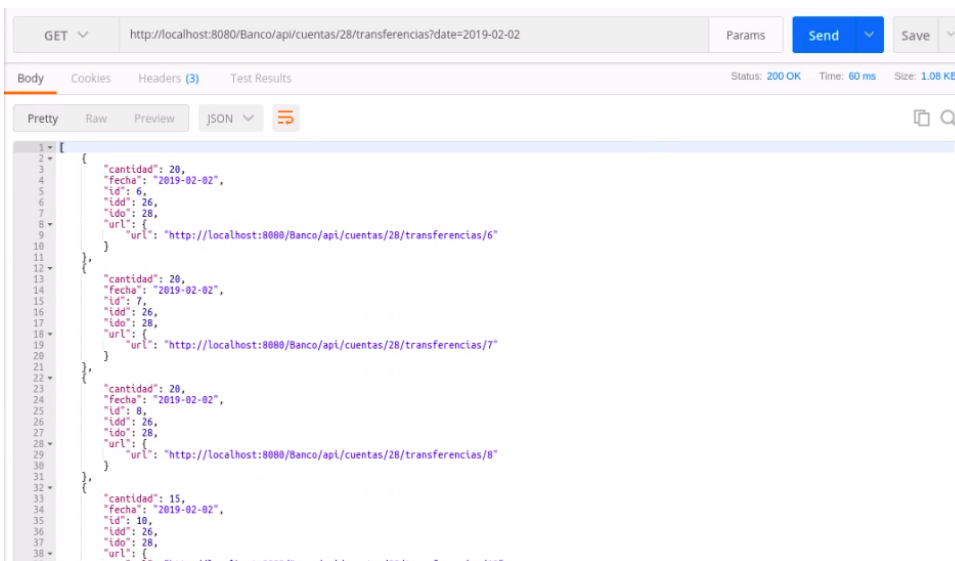
Status: 200 OK Time: 49 ms Size: 646 B

Body | Cookies | Headers (3) | Test Results

Pretty | Raw | Preview | JSON

```
1- [
2- {
3-   "cantidad": 70,
4-   "fecha": "2019-07-01 00:00:00.0",
5-   "id": 4,
6-   "url": {
7-     "url": "http://localhost:8080/Banco/api/cuentas/28/transferencias/4"
8-   }
9- }
10- {
11-   "cantidad": 20,
12-   "fecha": "2019-02-02 00:00:00.0",
13-   "id": 6,
14-   "url": {
15-     "url": "http://localhost:8080/Banco/api/cuentas/28/transferencias/6"
16-   }
17- }
18- {
19-   "cantidad": 20,
20-   "fecha": "2019-02-02 00:00:00.0",
21-   "id": 7,
22-   "url": {
23-     "url": "http://localhost:8080/Banco/api/cuentas/28/transferencias/7"
24-   }
25- }
26- {
27-   "cantidad": 20,
28-   "fecha": "2019-02-02 00:00:00.0",
29-   "id": 8,
30-   "url": {
31-     "url": "http://localhost:8080/Banco/api/cuentas/28/transferencias/8"
32-   }
33- }
34- ]
```

#### ○ Filtrado por fecha



GET | http://localhost:8080/Banco/api/cuentas/28/transferencias?date=2019-02-02 | Params | Send | Save

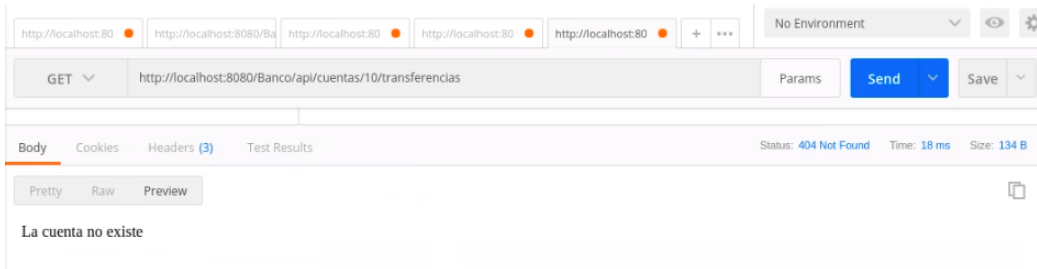
Status: 200 OK Time: 60 ms Size: 1.08 KB

Body | Cookies | Headers (3) | Test Results

Pretty | Raw | Preview | JSON

```
1- [
2- {
3-   "cantidad": 20,
4-   "fecha": "2019-02-02",
5-   "id": 6,
6-   "lido": 26,
7-   "lido": 28,
8-   "url": {
9-     "url": "http://localhost:8080/Banco/api/cuentas/28/transferencias/6"
10-  }
11- }
12- {
13-   "cantidad": 20,
14-   "fecha": "2019-02-02",
15-   "id": 7,
16-   "lido": 26,
17-   "lido": 28,
18-   "url": {
19-     "url": "http://localhost:8080/Banco/api/cuentas/28/transferencias/7"
20-   }
21- }
22- {
23-   "cantidad": 20,
24-   "fecha": "2019-02-02",
25-   "id": 8,
26-   "lido": 26,
27-   "lido": 28,
28-   "url": {
29-     "url": "http://localhost:8080/Banco/api/cuentas/28/transferencias/8"
30-   }
31- }
32- ],
33- {
34-   "cantidad": 15,
35-   "fecha": "2019-02-02",
36-   "id": 10,
37-   "lido": 26,
38-   "lido": 28,
39-   "url": {
40-     "url": "http://localhost:8080/Banco/api/cuentas/28/transferencias/10"
41-   }
42- }
43- ]
```

○ Si la cuenta no existe

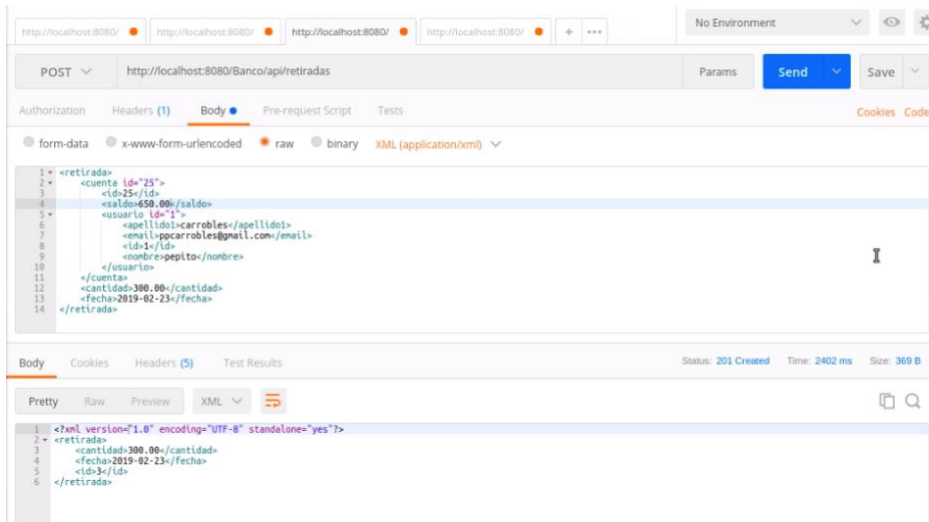


The screenshot shows a REST client interface with the following details:

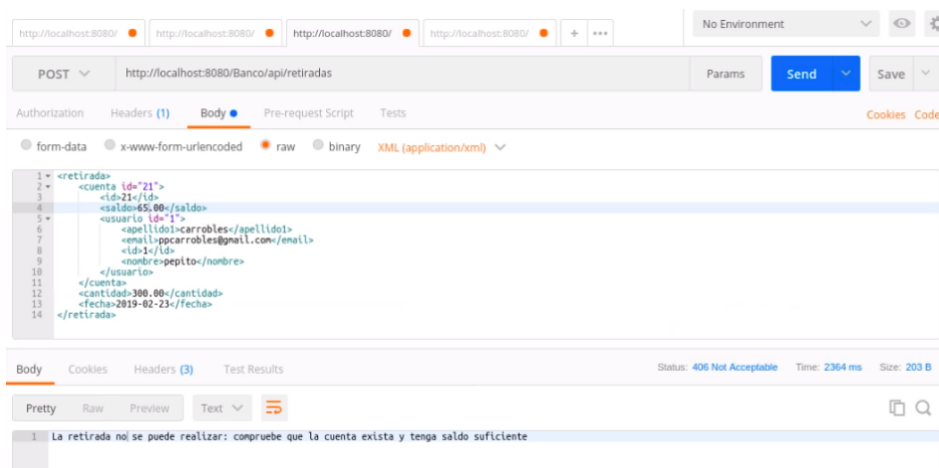
- URLs:** http://localhost:80, http://localhost:8080/Ba, http://localhost:80, http://localhost:80, http://localhost:80, +, ...
- Environment:** No Environment
- Method:** GET
- URL:** http://localhost:8080/Banco/api/cuentas/10/transferencias
- Params:** (empty)
- Buttons:** Send, Save
- Response:**
  - Status:** 404 Not Found
  - Time:** 18 ms
  - Size:** 134 B
  - Body:** La cuenta no existe

## 2.5. Retiradas

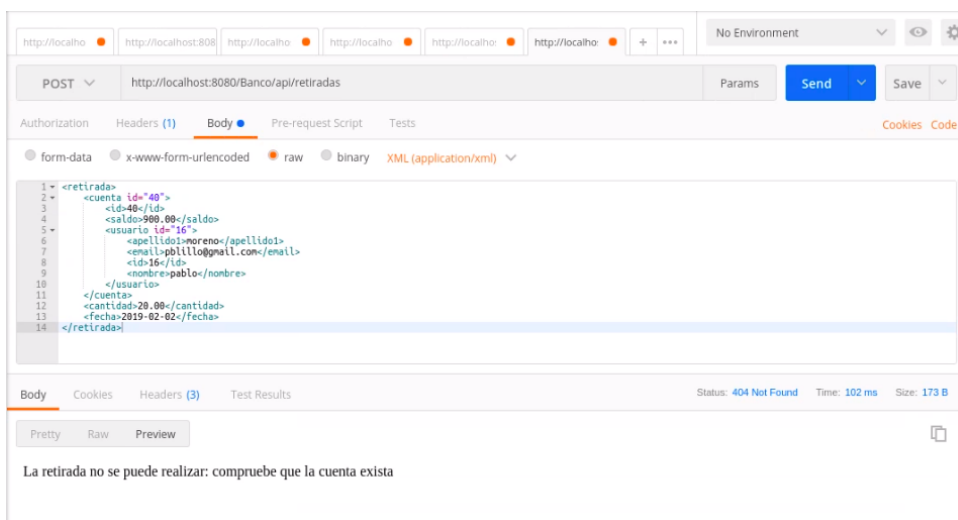
### 2.5.1. Realizar una retirada de efectivo



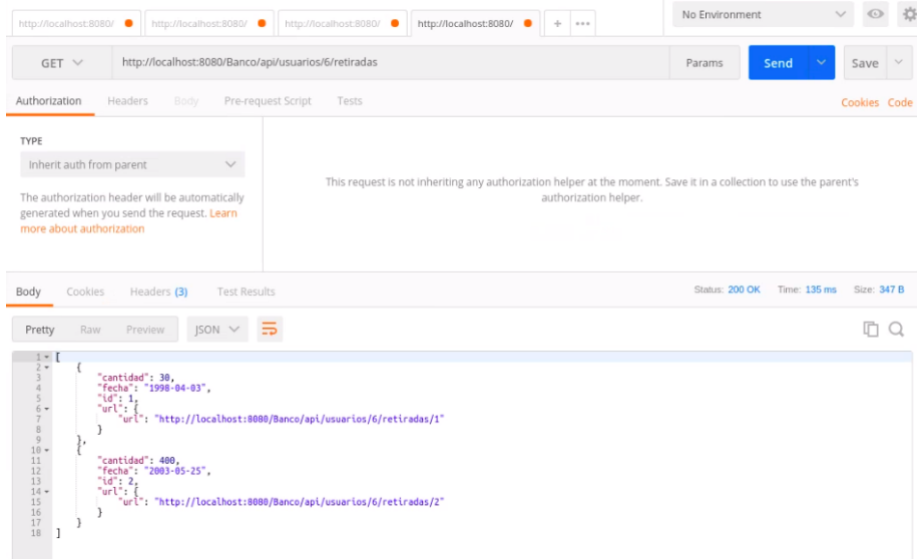
- Si la cuenta no tiene saldo suficiente



- Si la cuenta no existe



## 2.5.2. Listado de las retiradas de efectivo realizadas por un usuario



GET `http://localhost:8080/Banco/api/usuarios/6/retiradas` Params Send Save

Authorization Headers Body Pre-request Script Tests Cookies Code

TYPE: Inherit auth from parent

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

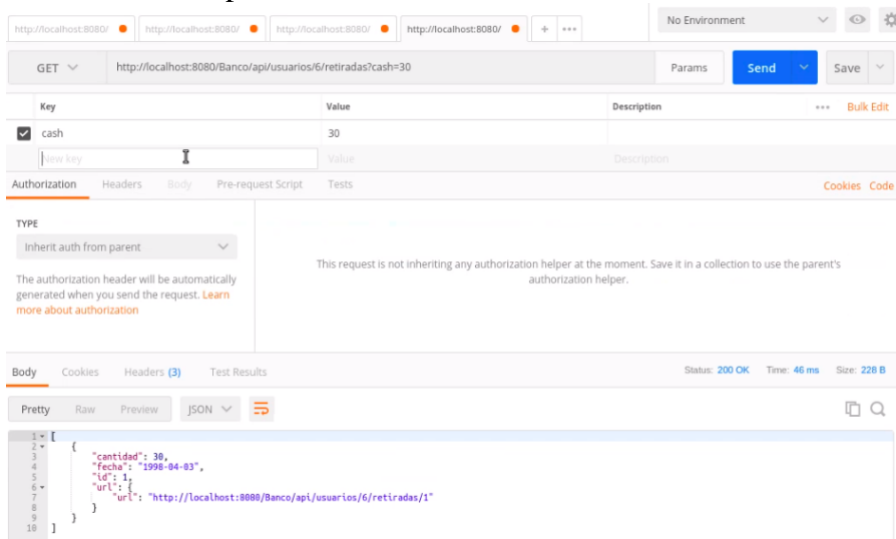
Body: Status: 200 OK Time: 135 ms Size: 347 B

```

1 {
2   {
3     "cantidad": 30,
4     "fecha": "1998-04-03",
5     "id": 1,
6     "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/1"
7   },
8   {
9     "cantidad": 400,
10    "fecha": "2003-05-25",
11    "id": 2,
12    "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/2"
13  }
14 }
15
16
17
18

```

### ○ Filtrado por cantidad



GET `http://localhost:8080/Banco/api/usuarios/6/retiradas?cash=30` Params Send Save

Key Value Description Bulk Edit

<input checked="" type="checkbox"/> cash	30		
<input type="text" value="New key"/>	<input type="text" value="Value"/>	<input type="text" value="Description"/>	

Authorization Headers Body Pre-request Script Tests Cookies Code

TYPE: Inherit auth from parent

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

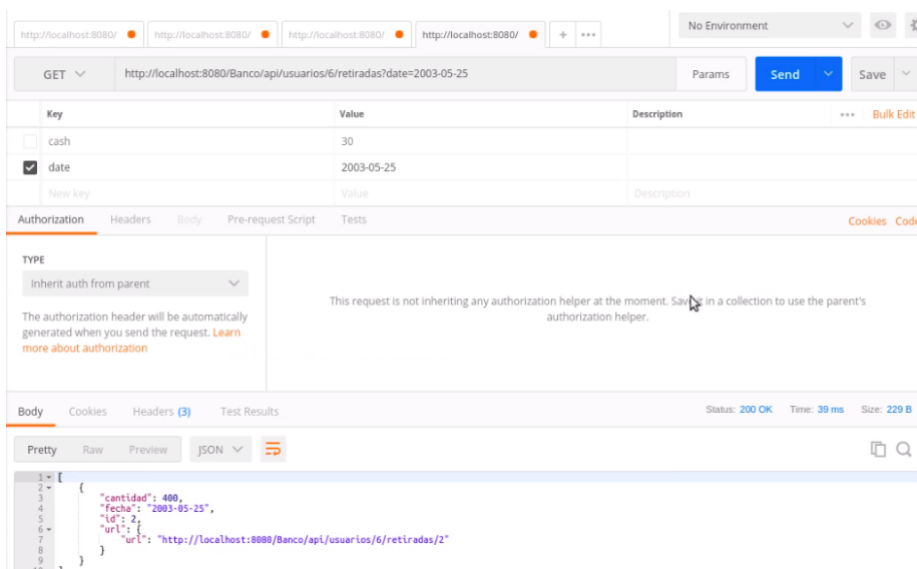
Body: Status: 200 OK Time: 46 ms Size: 228 B

```

1 {
2   {
3     "cantidad": 30,
4     "fecha": "1998-04-03",
5     "id": 1,
6     "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/1"
7   }
8 }
9
10

```

### ○ Filtrado por fecha



GET `http://localhost:8080/Banco/api/usuarios/6/retiradas?date=2003-05-25` Params Send Save

Key Value Description Bulk Edit

<input type="checkbox"/> cash	30		
<input checked="" type="checkbox"/> date	2003-05-25		
<input type="text" value="New key"/>	<input type="text" value="Value"/>	<input type="text" value="Description"/>	

Authorization Headers Body Pre-request Script Tests Cookies Code

TYPE: Inherit auth from parent

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

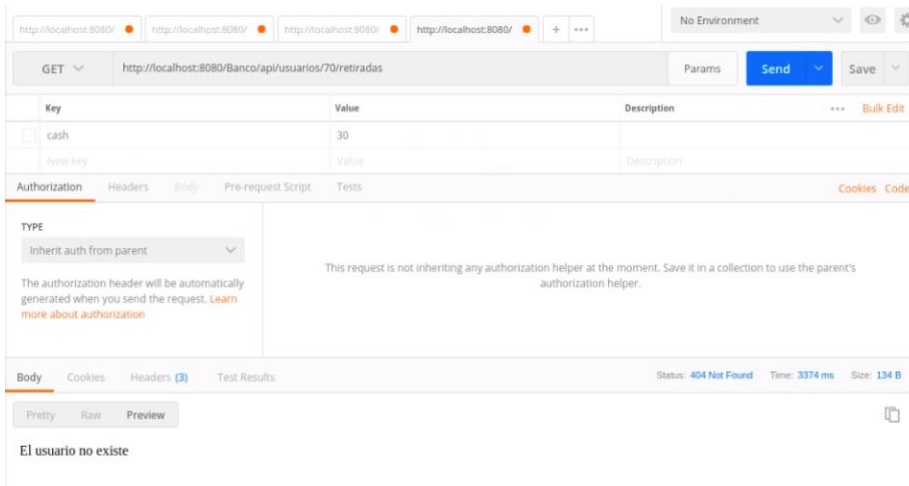
Body: Status: 200 OK Time: 39 ms Size: 229 B

```

1 {
2   {
3     "cantidad": 400,
4     "fecha": "2003-05-25",
5     "id": 2,
6     "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/2"
7   }
8 }
9
10

```

○ Si el usuario no existe



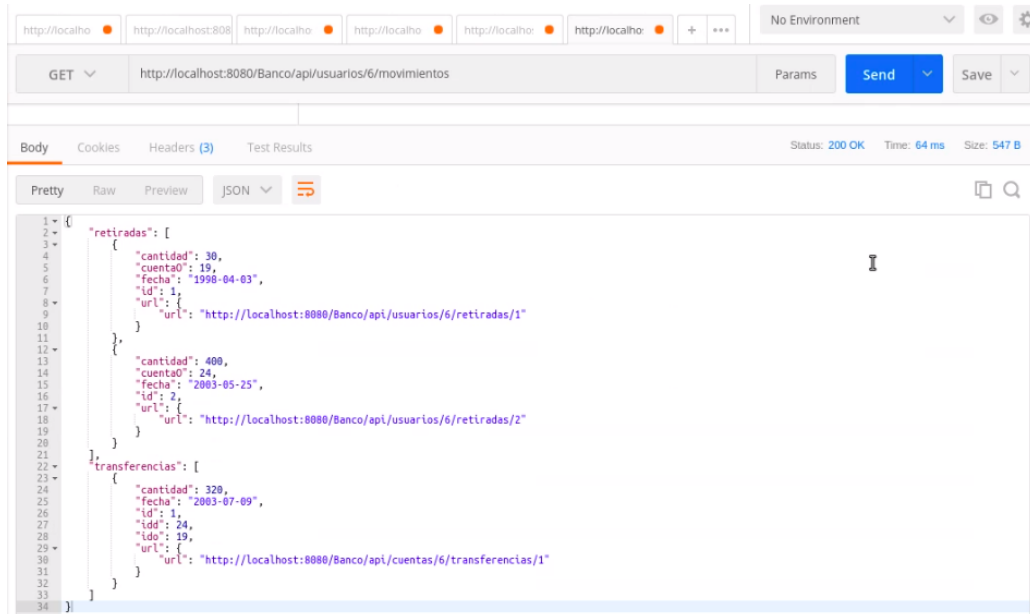
The screenshot shows a REST client interface with the following details:

- Environment:** No Environment
- Request:** GET `http://localhost:8080/Banco/api/usuarios/70/retiradas`
- Params:** A table with one entry: 

Key	Value	Description
cash	30	
- Authorization:** TYPE: Inherit auth from parent. Description: This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.
- Response:** Status: 404 Not Found, Time: 3374 ms, Size: 134 B. Body: El usuario no existe

## 2.6. Movimientos

### 2.6.1. Listado de todos los movimientos



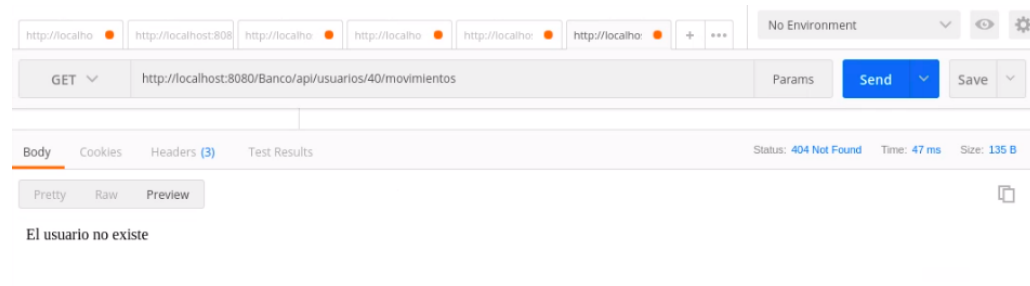
GET http://localhost:8080/Banco/api/usuarios/6/movimientos

Status: 200 OK Time: 64 ms Size: 547 B

```

{
  "retiradas": [
    {
      "cantidad": 30,
      "cuenta0": 19,
      "fecha": "1998-04-03",
      "id": 1,
      "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/1"
    },
    {
      "cantidad": 400,
      "cuenta0": 24,
      "fecha": "2003-05-25",
      "id": 2,
      "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/2"
    }
  ],
  "transferencias": [
    {
      "cantidad": 320,
      "fecha": "2003-07-09",
      "id": 1,
      "ido": 19,
      "url": "http://localhost:8080/Banco/api/cuentas/6/transferencias/1"
    }
  ]
}
  
```

#### ○ Si el usuario no existe



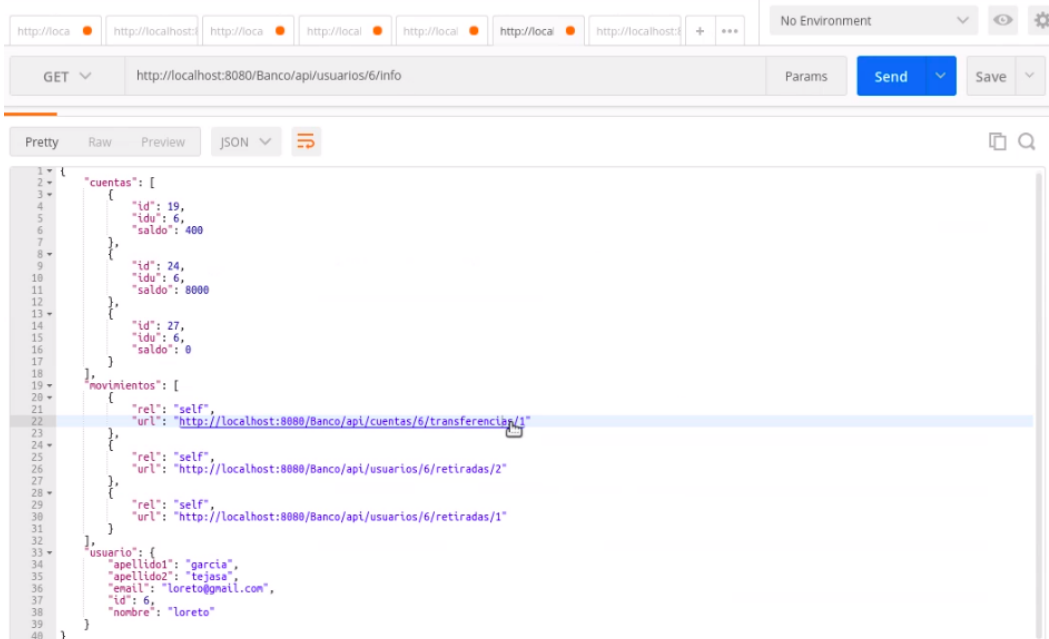
GET http://localhost:8080/Banco/api/usuarios/40/movimientos

Status: 404 Not Found Time: 47 ms Size: 135 B

El usuario no existe

## 2.7. Información para el móvil de un usuario

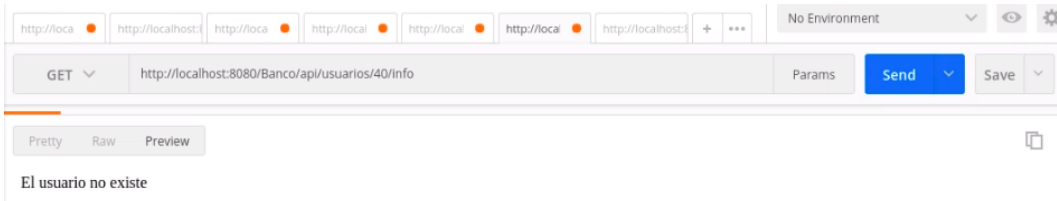
### 2.7.1. Mostrar los datos básicos de un cliente, sus cuentas y su saldo, los últimos 10 movimientos realizados en todas sus cuentas



```

1 {
2   "cuentas": [
3     {
4       "id": 19,
5       "idu": 6,
6       "saldo": 400
7     },
8     {
9       "id": 24,
10      "idu": 6,
11      "saldo": 8000
12    },
13    {
14      "id": 27,
15      "idu": 6,
16      "saldo": 0
17    }
18  ],
19  "movimientos": [
20    {
21      "rel": "self",
22      "url": "http://localhost:8080/Banco/api/cuentas/6/transferencias/1"
23    },
24    {
25      "rel": "self",
26      "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/2"
27    },
28    {
29      "rel": "self",
30      "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/1"
31    }
32  ],
33  "usuario": {
34    "apellido1": "garcia",
35    "apellido2": "tejasa",
36    "email": "loreto@gmail.com",
37    "id": 6,
38    "nombre": "loreto"
39  }
40 }
  
```

#### ○ Si el usuario no existe



```

GET http://localhost:8080/Banco/api/usuarios/40/info
{
  "El usuario no existe"
}
  
```

### 3. Capturas de la ejecución del cliente

Ver usuario existente -> 200

Estado: 200

Entidad:

```
{"apellido1":"gutierrez","apellido2":"martin","email":"mgm@gmail.com","id":5,"nombre":"maria","url":{"url":"http://localhost:8080/Banco/api/usuarios/5"}}
```

Ver usuario que no exista -> 404

Estado: 404

```
Entidad: <!doctype html><html lang="es"><head><title>Estado HTTP 404 – Not Found</title><style type="text/css">h1 { font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:22px;} h2 { font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:16px;} h3 { font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:14px;} body { font-family:Tahoma,Arial,sans-serif;color:black;background-color:white;} b { font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;} p { font-family:Tahoma,Arial,sans-serif;background:white;color:black;font-size:12px;} a { color:black;} a.name { color:black;} .line { height:1px;background-color:#525D76;border:none;}</style></head><body><h1>Estado HTTP 404 – Not Found</h1><hr class="line" /><p><b>Tipo</b> Informe de estado</p><p><b>mensaje</b> Not Found</p><p><b>descripción</b> El recurso requerido no está disponible.</p><hr class="line" /><h3>Apache Tomcat/9.0.6</h3></body></html>
```

Crear usuario -> 201 OK

Estado: 201

```
Entidad: <?xml version="1.0" encoding="UTF-8" standalone="yes"?><usuario><apellido1>fernandez</apellido1><apellido2>garcia</apellido2><email>josecarlosf@gmail.com</email><id>17</id><nombre>jose carlos</nombre></usuario>
Location: http://localhost:8080/Banco/api/usuarios/17
```

Crear usuario que ya exista -> 406

Estado: 406

Entidad: Ya se encuentra el usuario registrado

Modificar usuario -> 200

Estado: 200

```
Entidad: <?xml version="1.0" encoding="UTF-8" standalone="yes"?><usuario><apellido1>garcia</apellido1><apellido2>tejada</apellido2><email>loretogarcia@gmail.com</email><id>6</id><nombre>loreto</nombre></usuario>
```

Modificar usuario que no existe -> 404

Estado: 404

Entidad: No existe dicho usuario

Borrar usuario -> 204

Estado: 204

Borrar usuario que no existe -> 404



Estado: 404

Borrar usuario que tenga cuentas abiertas -> 406

Estado: 406

Crear cuenta -> 201

Estado: 201

Entidad: <?xml version="1.0" encoding="UTF-8" standalone="yes"?>< cuenta><id>33</id><idu>12</idu><saldo>50</saldo></ cuenta>  
Location: <http://localhost:8080/Banco/api/cuentas/33>

Crear cuenta a usuario no existente -> 404

Estado: 404

Entidad: El usuario no existe

Borrar cuenta -> 204

Estado: 204

Borrar cuenta con saldo distinto de 0 -> 406

Estado: 406

Borrar cuenta con cuenta inexistente -> 404

Estado: 404

Crear transferencia -> 201 OK

Estado: 201

Entidad: <?xml version="1.0" encoding="UTF-8" standalone="yes"?>< transferencia><cantidad>100</cantidad><fecha>2020-04-14</fecha><id>14</id><idd>30</idd><ido>22</ido></ transferencia>  
Location: <http://localhost:8080/Banco/api/transferencias/14>

Crear transferencia con cuenta origen no existente -> 404

Estado: 404

Entidad: La transferencia no se puede realizar: compruebe que las cuentas existan

Crear transferencia con saldo insuficiente -> 406

Estado: 406

Entidad: La transferencia no se puede realizar: compruebe que las cuentas existan y tengan saldo suficiente

Eliminar transferencias -> 204

Estado: 204

Eliminar transferencias cuando no existe -> 404

Estado: 404

Listar transferencias -> 200

Estado: 200

Entidad: [{"cantidad":70.00,"fecha":"2019-07-01","id":4,"idd":23,"ido":28,"url":{"url":"<http://localhost:8080/Banco/api/cuentas/28/transfer>"}}

```
encias/4"}},{ "cantidad":20.00,"fecha":"2019-02-02","id":6,"idd":26,"ido":28,"url":{"url":"http://localhost:8080/Banco/api/cuentas/28/transferencias/6"}},{ "cantidad":20.00,"fecha":"2019-02-02","id":7,"idd":26,"ido":28,"url":{"url":"http://localhost:8080/Banco/api/cuentas/28/transferencias/7"}},{ "cantidad":20.00,"fecha":"2019-02-02","id":8,"idd":26,"ido":28,"url":{"url":"http://localhost:8080/Banco/api/cuentas/28/transferencias/8"}},{ "cantidad":15.00,"fecha":"2019-02-02","id":11,"idd":26,"ido":28,"url":{"url":"http://localhost:8080/Banco/api/cuentas/28/transferencias/11"}},{ "cantidad":15.00,"fecha":"2019-02-02","id":12,"idd":26,"ido":28,"url":{"url":"http://localhost:8080/Banco/api/cuentas/28/transferencias/12"}},{ "cantidad":10.00,"fecha":"2019-02-02","id":13,"idd":26,"ido":28,"url":{"url":"http://localhost:8080/Banco/api/cuentas/28/transferencias/13"}}}
```

Listar transferencias cuando no existe cuenta -> 404

Estado: 404

Entidad: La cuenta no existe

Crear retirada -> 201 OK

Estado: 201

Entidad: <?xml version="1.0" encoding="UTF-8" standalone="yes"?><retirada><cantidad>20</cantidad><cuentaO>19</cuentaO><fecha>2020-03-23</fecha><id>3</id></retirada>

Location: <http://localhost:8080/Banco/api/retiradas/3>

Crear retirada con cuenta inexistente -> 404

Estado: 404

Entidad: La retirada no se puede realizar: compruebe que la cuenta exista

Crear retirada con saldo insuficiente -> 406

Estado: 406

Entidad: La retirada no se puede realizar: compruebe que la cuenta tenga saldo suficiente

Listar retiradas -> 200 OK

Estado: 200

Entidad: [{"cantidad":30.00,"cuentaO":19,"fecha":"1998-04-03","id":1,"url":{"url":"http://localhost:8080/Banco/api/usuarios/6/retiradas/1"}},{ "cantidad":400.00,"cuentaO":24,"fecha":"2003-05-25","id":2,"url":{"url":"http://localhost:8080/Banco/api/usuarios/6/retiradas/2"}},{ "cantidad":20.00,"cuentaO":19,"fecha":"2020-03-23","id":3,"url":{"url":"http://localhost:8080/Banco/api/usuarios/6/retiradas/3"}}]

Listar retiradas cuando usuario inexistente -> 404

Estado: 404

Entidad: El usuario no existe

Listar movimientos -> 200

Estado: 200

Entidad: {"retiradas":[{"cantidad":30.00,"cuentaO":19,"fecha":"1998-04-

```
03", "id":1, "url":{ "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/1" } }, { "cantidad":400.00, "cuentaO":24, "fecha": "2003-05-25", "id":2, "url":{ "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/2" } }, { "cantidad":20.00, "cuentaO":19, "fecha": "2020-03-23", "id":3, "url":{ "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/3" } } ], "transferencias": [ { "cantidad":320.00, "fecha": "2003-07-09", "id":1, "idd":24, "ido":19, "url":{ "url": "http://localhost:8080/Banco/api/cuentas/6/transferecias/1" } } ] }
```

Listar movimientos cuando usuario inexistente -> 404  
Estado: 404  
Entidad: El usuario no existe

Listar usuarios -> 200 OK  
Estado: 200  
Entidad:

```
[ { "id":1, "saldo":1515.00, "url":{ "url": "http://localhost:8080/Banco/api/usuarios/1" } }, { "id":5, "saldo":145.00, "url":{ "url": "http://localhost:8080/Banco/api/usuarios/5" } }, { "id":6, "saldo":8380.00, "url":{ "url": "http://localhost:8080/Banco/api/usuarios/6" } }, { "id":7, "saldo":25.00, "url":{ "url": "http://localhost:8080/Banco/api/usuarios/7" } }, { "id":12, "saldo":50.00, "url":{ "url": "http://localhost:8080/Banco/api/usuarios/12" } }, { "id":16, "saldo":3372.96, "url":{ "url": "http://localhost:8080/Banco/api/usuarios/16" } } ] }
```

Listar info -> 200 OK  
Estado: 200  
Entidad:

```
{ "cuentas": [ { "id":19, "idu":6, "saldo":380.00 }, { "id":24, "idu":6, "saldo":8000.00 } ], "movimientos": [ { "rel": "self", "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/3" }, { "rel": "self", "url": "http://localhost:8080/Banco/api/cuentas/6/transferecias/1" }, { "rel": "self", "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/2" }, { "rel": "self", "url": "http://localhost:8080/Banco/api/usuarios/6/retiradas/1" } ], "usuario": { "apellido1": "garcia", "apellido2": "tejada", "email": "loretogarcia@gmail.com", "id":6, "nombre": "loreto" } }
```

Listar info -> 404  
Estado: 404  
Entidad: El usuario no existe

**Nota:** aunque no se pedía implícitamente, hemos configurado los “GET” de transferencia y retirada, cuyas “URIs” son las mostradas en los ejemplos.