

## SISTEMA BANCARIO – TRANSFERENCIAS

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

Creo un cliente (Maria) pero aun sin cuenta.

The screenshot shows a Postman interface with the following details:

- Request URL:** `http://localhost:8080/api/clientes`
- Method:** POST
- Body Content:**

```
1 {  
2   "dni": 29282589,  
3   "nombre": "María",  
4   "apellido": "Losada",  
5   "fechaNacimiento": "1982-02-26",  
6   "tipoPersona": "F",  
7   "banco": "Banco Nación"  
8 }  
9  
10  
11
```
- Response Status:** 200 OK
- Response Time:** 198 ms
- Response Body (Pretty JSON):**

```
1 Cliente creado con éxito: María
```

POST: http://localhost:8080/api/cuentas

Al cliente María le creo una cuenta CA y en pesos.

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/api/cuentas`. The request body is a JSON object:

```
1 {  
2   "dniTitular": 29282589,  
3   "tipoCuenta": "CAJA_AHORRO",  
4   "moneda": "PESOS",  
5   "balanceInicial": 1200000  
6 }  
7
```

The response is a 200 OK status with a response time of 39 ms and a size of 293 bytes. The response body is:

```
1 {  
2   "numeroCuenta": 887443824,  
3   "tipoCuenta": "CAJA_AHORRO",  
4   "moneda": "PESOS",  
5   "balance": 1200000.0,  
6   "titularNombreCompleto": "Maria Losada"  
7 }
```

POST: http://localhost:8080/api/cuentas

Al cliente María le creo otra cuenta CA pero en dólares.

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/api/cuentas`. The request body is a JSON object:

```
1 {  
2   "dniTitular": 29282589,  
3   "tipoCuenta": "CAJA_AHORRO",  
4   "moneda": "DOLARES",  
5   "balanceInicial": 12000  
6 }  
7
```

The response tab shows a successful `200 OK` response with a response time of 7 ms and a size of 293 bytes. The response body is:

```
1 {  
2   "numeroCuenta": 855377730,  
3   "tipoCuenta": "CAJA_AHORRO",  
4   "moneda": "DOLARES",  
5   "balance": 12000.0,  
6   "titularNombreCompleto": "Maria Losada"  
7 }
```

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

Al cliente María le creo una cuenta CC y en pesos

The screenshot shows the Postman interface with the following details:

- Request URL:** `http://localhost:8080/api/cuentas`
- Method:** POST
- Body:** Raw JSON (selected)
- JSON Content:**

```
1 {  
2   "dniTitular": 29282589,  
3   "tipoCuenta": "CUENTA_CORRIENTE",  
4   "moneda": "PESOS",  
5   "balanceInicial": 300000  
6 }  
7
```
- Response Status:** 200 OK
- Response Body (Pretty JSON):**

```
1 {  
2   "numeroCuenta": 380343383,  
3   "tipoCuenta": "CUENTA_CORRIENTE",  
4   "moneda": "PESOS",  
5   "balance": 300000.0,  
6   "titularNombreCompleto": "Maria Losada"  
7 }
```

GET: <http://localhost:8080/api/clientes/{dni}>

Veo al cliente María, sus cuentas y balances de las mismas.

```
{  
  "nombre": "Maria",  
  "apellido": "Losada",  
  "dni": 29282589,  
  "fechaNacimiento": "1982-02-26",  
  "tipoPersona": "PERSONA_FISICA",  
  "banco": "Banco Nación",  
  "fechaAlta": "2024-12-14",  
  "cuentas": [  
    {  
      "numeroCuenta": 855377730,  
      "fechaCreacion": "2024-12-14T12:14:49.4486234",  
      "balance": 12000.0,  
      "tipoCuenta": "CAJA_AHORRO",  
      "titular": {  
        "nombre": "Maria",  
        "apellido": "Losada",  
        "dni": 29282589,  
        "fechaNacimiento": "1982-02-26",  
        "tipoPersona": "PERSONA_FISICA",  
        "banco": "Banco Nación",  
        "fechaAlta": "2024-12-14",  
        "cuentas": [  
          {  
            "numeroCuenta": 380343383,  
            "fechaCreacion": "2024-12-14T12:14:49.4486234",  
            "balance": 300000.0,  
            "tipoCuenta": "CUENTA_CORRIENTE",  
            "titular": {  
              "nombre": "Maria",  
              "apellido": "Losada",  
              "dni": 29282589,  
              "fechaNacimiento": "1982-02-26",  
              "tipoPersona": "PERSONA_FISICA",  
              "banco": "Banco Nación",  
              "fechaAlta": "2024-12-14",  
              "cuentas": []  
            }  
          }  
        ]  
      }  
    }  
  ]  
}
```

```

        "moneda": "DOLARES",
        "movimientos": []
    },
    {
        "numeroCuenta": 887443824,
        "fechaCreacion": "2024-12-14T12:09:31.467719",
        "balance": 1200000.0,
        "tipoCuenta": "CAJA_AHORRO",
        "moneda": "PESOS",
        "movimientos": []
    },
    {
        "numeroCuenta": 380343383,
        "fechaCreacion": "2024-12-14T12:47:28.7329521",
        "balance": 300000.0,
        "tipoCuenta": "CUENTA_CORRIENTE",
        "moneda": "PESOS",
        "movimientos": []
    }
],
"edad": 42
}

```

HTTP <http://localhost:8080/api/clientes/29282589>

GET <http://localhost:8080/api/clientes/29282589>

Params	Authorization	Headers (7)	Body	Scripts	Tests	Settings
Body	Cookies	Headers (5)	Test Results		200 OK	18
<input checked="" type="radio"/> Pretty <input type="radio"/> Raw <input type="radio"/> Preview <input type="radio"/> Visualize <input type="radio"/> JSON						
<pre> 1  { 2      "nombre": "Maria", 3      "apellido": "Losada", 4      "dni": 29282589, 5      "fechaNacimiento": "1982-02-26", 6      "tipoPersona": "PERSONA_FISICA", 7      "banco": "Banco Nación", 8      "fechaAlta": "2024-12-14", 9      "cuentas": [ 10         { 11             "numeroCuenta": 855377730, 12             "fechaCreacion": "2024-12-14T12:14:49.4486234", 13             "balance": 12000.0, 14             "tipoCuenta": "CAJA_AHORRO", 15             "moneda": "DOLARES", 16             "movimientos": [] 17         }, 18         { 19             "numeroCuenta": 887443824, 20             "fechaCreacion": "2024-12-14T12:09:31.467719", 21             "balance": 1200000.0, 22             "tipoCuenta": "CAJA_AHORRO", 23             "moneda": "PESOS", 24             "movimientos": [] 25         }, 26         { 27             "numeroCuenta": 380343383, 28             "fechaCreacion": "2024-12-14T12:47:28.7329521", 29             "balance": 300000.0, </pre>						

POST: <http://localhost:8080/api/clientes/con-cuenta>

Creo un cliente (Mauro) con cuenta CA y pesos.

The screenshot shows the Postman interface with the following details:

- Request URL:** `http://localhost:8080/api/clientes/con-cuenta`
- Method:** POST
- Body:** JSON (selected)
- Request Body Content:**

```
1 {  
2     "nombre": "Mauro",  
3     "apellido": "Sabbatini",  
4     "dni": 25665407,  
5     "fechaNacimiento": "1977-03-15",  
6     "tipoPersona": "F",  
7     "banco": "Banco Provincia",  
8     "tipoCuenta": "CAJA_AHORRO",  
9     "moneda": "PESOS",  
10    "balance": 1500000  
11 }
```
- Response Status:** 200 OK
- Response Time:** 41 ms
- Response Body Content:**

```
1 {  
2     "nombre": "Mauro",  
3     "apellido": "Sabbatini",  
4     "dni": 25665407,  
5     "fechaNacimiento": "1977-03-15",  
6     "tipoPersona": "PERSONA_FISICA",  
7     "banco": "Banco Provincia",  
8     "fechaAlta": "2024-12-14",  
9     "cuentas": [  
10         {  
11             "numeroCuenta": 529392355,  
12             "fechaCreacion": "2024-12-14T12:17:40.2087471",  
13             "balance": 1500000.0,  
14             "tipoCuenta": "CAJA_AHORRO",  
15             "moneda": "PESOS",  
16             "movimientos": []  
17         },  
18     ],  
19     "edad": 47  
20 }
```

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

Al cliente Mauro le creo una cuenta CA y dólares

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/api/cuentas`. The request body is a JSON object:

```
1 {  
2   "dniTitular": 25665407,  
3   "tipoCuenta": "CAJA_AHORRO",  
4   "moneda": "DOLARES",  
5   "balanceInicial": 10000  
6 }  
7
```

The response status is 200 OK with a response time of 10 ms and a size of 296 bytes. The response body is also a JSON object:

```
1 {  
2   "numeroCuenta": 969846825,  
3   "tipoCuenta": "CAJA_AHORRO",  
4   "moneda": "DOLARES",  
5   "balance": 10000.0,  
6   "titularNombreCompleto": "Mauro Sabbatini"  
7 }
```

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

El cliente María le transfiere al cliente Mauro 15000 pesos, es sin comisión.

The screenshot shows a Postman request and response interface. The request is a POST to <http://localhost:8080/api/transferencias>. The Body tab is selected, showing a raw JSON payload:

```
1 {  
2   "cuentaOrigen": 887443824, // Número de cuenta del Cliente María  
3   "cuentaDestino": 529392355, // Número de cuenta del Cliente Mauro  
4   "monto": 15000,           // Monto a transferir  
5   "moneda": "PESOS"        // Moneda de la transferencia  
6 }  
7
```

The response status is 200 OK. The Body tab shows the response JSON:

```
1 {  
2   "estado": "EXITOSA",  
3   "mensaje": "Transferencia exitosa"  
4 }
```

GET: <http://localhost:8080/api/cuentas/887443824/movimientos>

Veo en la cuenta de Maria el monto transferido a la cuenta de Mauro.

The screenshot shows a Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8080/api/cuentas/887443824/movimientos
- Params:** Key (Value) - Description
- Body:** Cookies, Headers (5), Test Results, 200 OK
- JSON Response (Pretty):**

```
1 {  
2   "numeroCuenta": 887443824,  
3   "movimientos": [  
4     {  
5       "descripcionBreve": "Transferencia realizada a la cuenta 529392355",  
6       "monto": 15000.0,  
7       "tipo": "DÉBITO",  
8       "fecha": "2024-12-14T13:06:48.6525786"  
9     }  
10   ]  
11 }
```

GET: <http://localhost:8080/api/cuentas/529392355/movimientos>

Veo en la cuenta de Mauro el monto que le transfirió María.

The screenshot shows a Postman request to `http://localhost:8080/api/cuentas/529392355/movimientos`. The response is a 200 OK status with the following JSON body:

```
1 {  
2   "numeroCuenta": 529392355,  
3   "movimientos": [  
4     {  
5       "descripcionBreve": "Transferencia recibida de la cuenta 887443824",  
6       "monto": 15000.0,  
7       "tipo": "CRÉDITO",  
8       "fecha": "2024-12-14T13:06:48.6536148"  
9     }  
10   ]  
11 }
```

GET: <http://localhost:8080/api/clientes/{dni}>

Puedo ver al cliente María, sus cuentas, sus movimientos. En esta solicitud se ve la transferencia que le realizó al cliente Mauro y se ve el balance modificado.

```
{  
  "nombre": "Maria",  
  "apellido": "Losada",  
  "dni": 29282589,  
  "fechaNacimiento": "1982-02-26",  
  "tipoPersona": "PERSONA_FISICA",
```

```
"banco": "Banco Nación",
"fechaAlta": "2024-12-14",
"cuentas": [
  {
    "numeroCuenta": 855377730,
    "fechaCreacion": "2024-12-14T12:14:49.4486234",
    "balance": 12000.0,
    "tipoCuenta": "CAJA_AHORRO",
    "moneda": "DOLARES",
    "movimientos": []
  },
  {
    "numeroCuenta": 887443824,
    "fechaCreacion": "2024-12-14T12:09:31.467719",
    "balance": 1185000.0,
    "tipoCuenta": "CAJA_AHORRO",
    "moneda": "PESOS",
    "movimientos": [
      {
        "numeroCuenta": 887443824,
        "tipo": "DÉBITO",
        "monto": 15000.0,
        "descripcion": "Débito por transferencia",
        "fecha": "2024-12-14T13:06:48.6525786",
        "id": 1
      }
    ]
  },
  {
    "numeroCuenta": 380343383,
    "fechaCreacion": "2024-12-14T12:47:28.7329521",
    "balance": 300000.0,
    "tipoCuenta": "CUENTA_CORRIENTE",
    "moneda": "PESOS",
    "movimientos": []
  }
],
"edad": 42
}
```

HTTP <http://localhost:8080/api/clientes/29282589>

GET <http://localhost:8080/api/clientes/29282589>

Params Authorization Headers (7) Body Scripts Tests Settings

Body Cookies Headers (5) Test Results ⚙ 200 OK

Pretty Raw Preview Visualize JSON

```
12 |     "fechaCreacion": "2024-12-14T12:14:49.4486234",
13 |     "balance": 12000.0,
14 |     "tipoCuenta": "CAJA_AHORRO",
15 |     "moneda": "DOLARES",
16 |     "movimientos": []
17 |
18 },
19 {
20     "numeroCuenta": 887443824,
21     "fechaCreacion": "2024-12-14T12:09:31.467719",
22     "balance": 1185000.0,
23     "tipoCuenta": "CAJA_AHORRO",
24     "moneda": "PESOS",
25     "movimientos": [
26         {
27             "numeroCuenta": 887443824,
28             "tipo": "DÉBITO",
29             "monto": 15000.0,
30             "descripcion": "Débito por transferencia",
31             "fecha": "2024-12-14T13:06:48.6525786",
32             "id": 1
33         }
34     ],
35 },
36 {
37     "numeroCuenta": 380343383,
38     "fechaCreacion": "2024-12-14T12:47:28.7329521",
39     "balance": 300000.0,
40     "tipoCuenta": "CUENTA_CORRIENTE",
41     "moneda": "PESOS",
42 }
```

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

Ahora realizo una transferencia de María a Mauro con comisión porque supera el millón de pesos.

The screenshot shows the Postman interface with a successful API call. The URL is <http://localhost:8080/api/transferencias>. The request method is POST. The Body tab is selected, showing a JSON payload:

```
1 {  
2   "cuentaOrigen": 887443824, // Número de cuenta del Cliente María  
3   "cuentaDestino": 529392355, // Número de cuenta del Cliente Mauro  
4   "monto": 1000001,          // Monto a transferir  
5   "moneda": "PESOS"         // Moneda de la transferencia  
6 }  
7
```

The response status is 200 OK, with a response time of 8 ms and a size of 21 bytes. The response body is:

```
1 {  
2   "estado": "EXITOSA",  
3   "mensaje": "Transferencia exitosa"  
4 }
```

GET: <http://localhost:8080/api/cuentas/{nrodecuenta}/movimientos>

En esta solicitud se ve la cuenta del cliente Mauro que recibió el monto desde la cuenta de María.

The screenshot shows a Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8080/api/cuentas/529392355/movimientos
- Params:** Key
- Query Params:** Key
- Body:** JSON response (Pretty, Raw, Preview, Visualize) showing the following data:

```
1 {  
2   "numeroCuenta": 529392355,  
3   "movimientos": [  
4     {  
5       "descripcionBreve": "Transferencia recibida de la cuenta 887443824",  
6       "monto": 15000.0,  
7       "tipo": "CRÉDITO",  
8       "fecha": "2024-12-14T13:06:48.6536148"  
9     },  
10    {  
11      "descripcionBreve": "Transferencia recibida de la cuenta 887443824",  
12      "monto": 1000001.0,  
13      "tipo": "CRÉDITO",  
14      "fecha": "2024-12-14T13:25:47.2946151"  
15    }  
16  ]  
17 }
```

The response status is 200 OK.

GET: <http://localhost:8080/api/cuentas/887443824/movimientos>

En esta solicitud se ve las transferencias realizadas de María a Mauro. Y en la última como el monto superaba el millón de pesos se le cobró una comisión. El monto a transferir era de \$1000001 y se le debito \$1020001,02.

The screenshot shows a Postman request configuration for a GET request to `http://localhost:8080/api/cuentas/887443824/movimientos`. The response status is 200 OK with a duration of 7 ms and a size of 48 bytes. The response body is displayed in JSON format, showing the account number and two movement objects. The first movement has a monto of 15000.0 and the second has a monto of 1020001.02, reflecting a commission deduction.

```
1 {  
2     "numeroCuenta": 887443824,  
3     "movimientos": [  
4         {  
5             "descripcionBreve": "Transferencia realizada a la cuenta 529392355",  
6             "monto": 15000.0,  
7             "tipo": "DÉBITO",  
8             "fecha": "2024-12-14T13:06:48.6525786"  
9         },  
10        {  
11            "descripcionBreve": "Transferencia realizada a la cuenta 529392355",  
12            "monto": 1020001.02,  
13            "tipo": "DÉBITO",  
14            "fecha": "2024-12-14T13:25:47.2946151"  
15        }  
16    ]  
17 }
```

## Ahora voy a documentar casos pocos felices

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

Cuando quiero crear un cliente que ya existe.

HTTP <http://localhost:8080/api/clientes>

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

Params Authorization Headers (9) Body Scripts Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL  JSON

```
1 {
2   "dni": 29282589,
3   "nombre": "Maria",
4   "apellido": "Losada",
5   "fechaNacimiento": "1982-02-26",
6   "tipoPersona": "F",
7   "banco": "Banco Nación"
8 }
```

Body Cookies Headers (4) Test Results 400 Bad Request

Pretty Raw Preview Visualize JSON

```
1 {
2   "errorCode": 4003,
3   "errorMessage": "Error: Ya existe un cliente con el DNI: 29282589"
4 }
```

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

Cuando quiero crear un cliente, pero escribe mal el DNI.

The screenshot shows a Postman interface with the following details:

- HTTP Method:** POST
- URL:** <http://localhost:8080/api/clientes>
- Body:** Raw JSON (selected)
- JSON Body Content:**

```
1 {  
2   "dni": 292289,  
3   "nombre": "Maria",  
4   "apellido": "Losada",  
5   "fechaNacimiento": "1982-02-26",  
6   "tipoPersona": "F",  
7   "banco": "Banco Nación"  
8 }
```
- Response Status:** 400 Bad Request
- Response Body (Pretty JSON):**

```
1 {  
2   "errorCode": 4004,  
3   "errorMessage": "El DNI debe ser un número positivo de 7 u 8 dígitos"  
4 }
```

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

Cuando quiero crear un cliente, pero es menor de edad.

HTTP <http://localhost:8080/api/clientes>

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

Params Authorization Headers (9) Body Scripts Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL  JSON

```
1 {  
2   "dni": 1234567,  
3   "nombre": "Maria",  
4   "apellido": "Losada",  
5   "fechaNacimiento": "2020-02-26",  
6   "tipoPersona": "F",  
7   "banco": "Banco Nación"  
8 }  
9  
10  
11  
12
```

Body Cookies Headers (4) Test Results | ⚙ 400 Bad Request

Pretty Raw Preview Visualize JSON

```
1 {  
2   "errorCode": 4004,  
3   "errorMessage": "El cliente debe ser mayor de 18 años"  
4 }
```

⚙️ Postbox 🚧 Runner ⚡ Start Preview

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

Al querer crear un cliente escribo mal la fecha de nacimiento.

The screenshot shows a Postman interface with the following details:

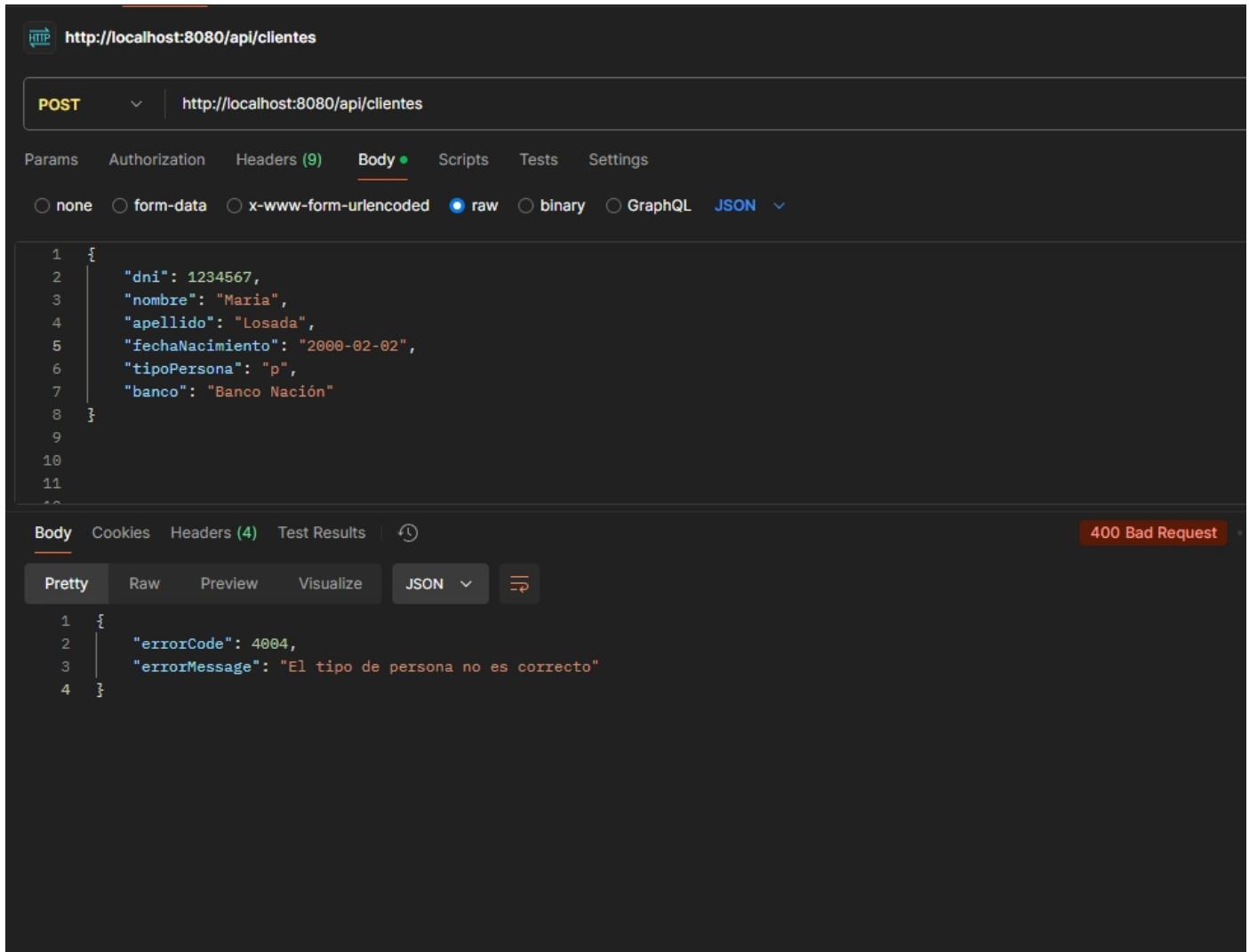
- Method:** POST
- URL:** <http://localhost:8080/api/clientes>
- Body (JSON):**

```
1 {  
2   "dni": "1234567",  
3   "nombre": "Maria",  
4   "apellido": "Losada",  
5   "fechaNacimiento": "21-02-2000",  
6   "tipoPersona": "F",  
7   "banco": "Banco Nación"  
8 }  
9  
10  
11  
12
```
- Response Status:** 400 Bad Request
- Response Body (Pretty JSON):**

```
1 {  
2   "errorCode": 4005,  
3   "errorMessage": "La fecha de nacimiento debe usar el formato año-mes-dia (YYYY-MM-DD)."  
4 }
```

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

Al querer crear un cliente escribo mal el tipo de persona.



The screenshot shows a POST request to `http://localhost:8080/api/clientes`. The request body is a JSON object with the following structure:

```
1 {  
2   "dni": 1234567,  
3   "nombre": "Maria",  
4   "apellido": "Losada",  
5   "fechaNacimiento": "2000-02-02",  
6   "tipoPersona": "p",  
7   "banco": "Banco Nación"  
8 }  
9  
10  
11  
12
```

The response status is **400 Bad Request**. The response body is a JSON object with the following structure:

```
1 {  
2   "errorCode": 4004,  
3   "errorMessage": "El tipo de persona no es correcto"  
4 }
```

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

Al querer crear un cliente dejo vacío el campo de banco.

http://localhost:8080/api/clientes

POST http://localhost:8080/api/clientes

Params Authorization Headers (9) Body Scripts Tests Settings

Body (raw JSON)

```
1 {  
2   "dni": "1234567",  
3   "nombre": "Maria",  
4   "apellido": "Losada",  
5   "fechaNacimiento": "2000-02-02",  
6   "tipoPersona": "F",  
7   "banco": ""  
8 }  
9  
10  
11  
12
```

Body Cookies Headers (4) Test Results 400 Bad Request

Pretty Raw Preview Visualize JSON

```
1 {  
2   "errorCode": 4004,  
3   "errorMessage": "El banco no puede ser solo espacios en blanco"  
4 }
```

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

Cuando quiero crear un cliente, pero no escribió su nombre.

HTTP <http://localhost:8080/api/clientes>

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

Params Authorization Headers (9) **Body** Scripts Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON**

```
1 {
2   "dni": 1234567,
3   "nombre": "",
4   "apellido": "Losada",
5   "fechaNacimiento": "1982-02-26",
6   "tipoPersona": "F",
7   "banco": "Banco Nación"
8 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 7

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "errorCode": 4004,
3   "errorMessage": "El nombre no puede estar vacío"
4 }
```

GET: <http://localhost:8080/api/clientes/{dni}>

Quiero que me muestre un cliente inexistente.

The screenshot shows a Postman request configuration and its results. The URL is `http://localhost:8080/api/clientes/31000000`. The method is set to `GET`. In the `Params` tab, there is a single entry for `Key` and `Value`. The response status is `400 Bad Request`, and the response body is a JSON object:

```
1 {  
2     "errorCode": 4000,  
3     "errorMessage": "Cliente no encontrado con DNI: 31000000"  
4 }
```

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

Cuando quiero crear una cuenta, pero el cliente escribió mal el tipo de cuenta.

The screenshot shows a Postman interface with the following details:

- Request URL:** `http://localhost:8080/api/cuentas`
- Method:** POST
- Body:** Raw JSON (selected)

```
1 {  
2   "dniTitular": 29282589,  
3   "tipoCuenta": "CAJA_AHOR",  
4   "moneda": "PESOS",  
5   "balanceInicial": 1  
6 }  
7
```
- Response Status:** 400 Bad Request
- Response Body (Pretty JSON):**

```
1 {  
2   "errorCode": 4004,  
3   "errorMessage": "El tipo de cuenta es inválido. Debe ser CAJA_AHORRO o CUENTA_CORRIENTE."  
4 }
```

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

Cuando quiero crear una cuenta, pero el cliente escribió mal el tipo de moneda.

The screenshot shows a Postman interface with the following details:

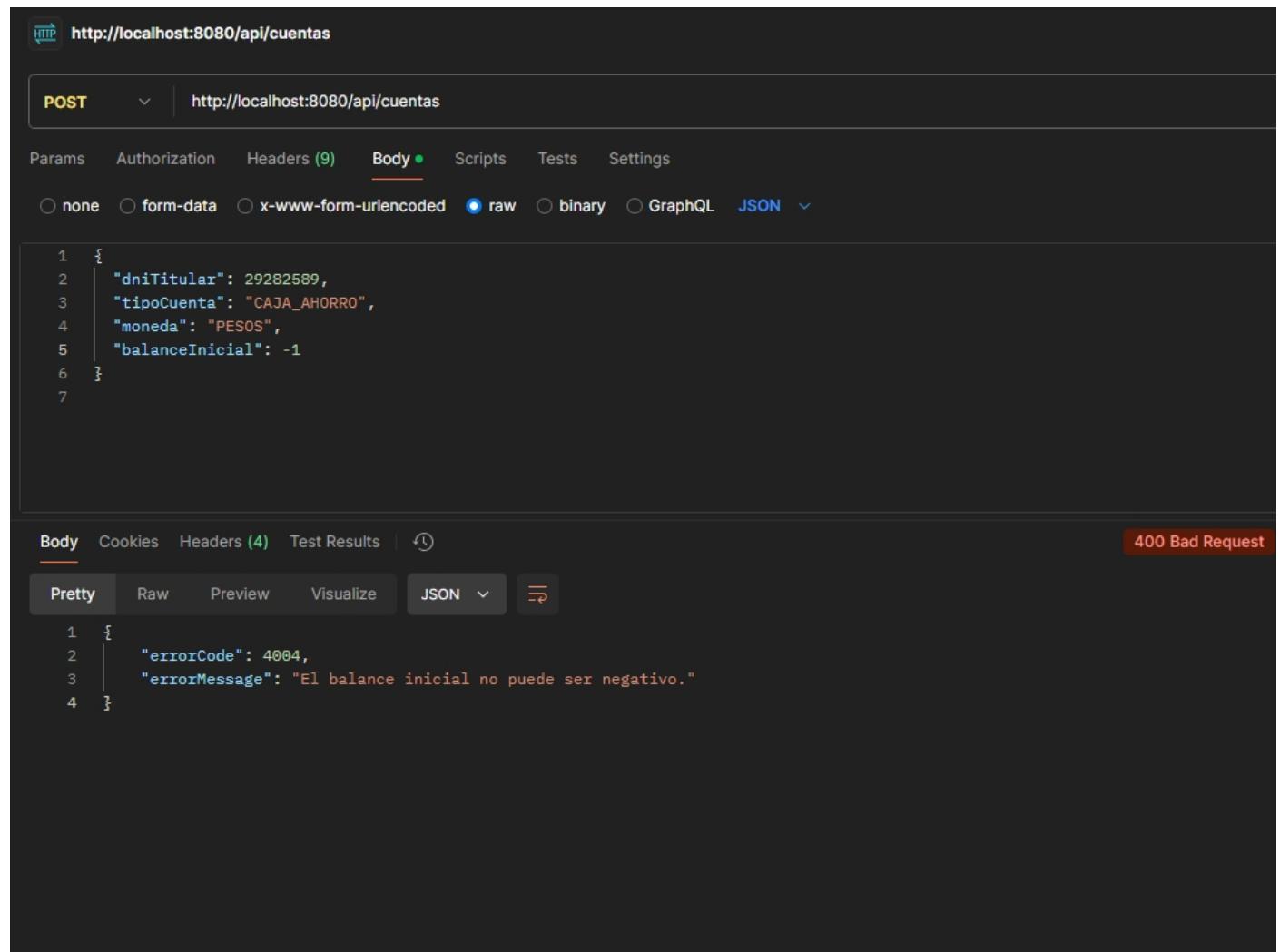
- Request Method:** POST
- URL:** <http://localhost:8080/api/cuentas>
- Body:** JSON (selected)
- Body Content:**

```
1 {  
2     "dniTitular": 29282589,  
3     "tipoCuenta": "CAJA_AHORRO",  
4     "moneda": "SOLES",  
5     "balanceInicial": 1  
6 }  
7
```
- Response Status:** 400 Bad Request
- Response Body (Pretty JSON):**

```
1 {  
2     "errorCode": 4004,  
3     "errorMessage": "La moneda es inválida. Debe ser PESOS o DOLARES."  
4 }
```

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

Cuando quiero crear una cuenta con balance negativo.



The screenshot shows a Postman interface with the following details:

- HTTP Method:** POST
- URL:** http://localhost:8080/api/cuentas
- Body:** Raw JSON (selected)
- JSON Content:**

```
1 {  
2   "dniTitular": 29282589,  
3   "tipoCuenta": "CAJA_AHORRO",  
4   "moneda": "PESOS",  
5   "balanceInicial": -1  
6 }  
7
```
- Response Status:** 400 Bad Request
- Response Body (Pretty JSON):**

```
1 {  
2   "errorCode": 4004,  
3   "errorMessage": "El balance inicial no puede ser negativo."  
4 }
```

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

Cuando quiero crear una cuenta, pero el cliente ya tiene una cuenta en ese tipo y moneda.

HTTP <http://localhost:8080/api/cuentas>

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

Params Authorization Headers (9) **Body** Scripts Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON**

```
1 {  
2   "dniTitular": 29282589,  
3   "tipoCuenta": "CAJA_AHORRO",  
4   "moneda": "PESOS",  
5   "balanceInicial": 1  
6 }  
7
```

Body Cookies Headers (4) Test Results 400 Bad Request

Pretty Raw Preview Visualize **JSON**

```
1 {  
2   "errorCode": 4002,  
3   "errorMessage": "Error: El cliente ya tiene una cuenta del tipo CAJA_AHORRO y moneda PESOS"  
4 }
```

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

Cuando la cuenta origen no tiene fondos suficientes

HTTP <http://localhost:8080/api/transferencias>

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

Params Authorization Headers (9) Body Scripts Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL [JSON](#)

```
1 {
2   "cuentaOrigen": 887443824, // Número de cuenta del Cliente Maria
3   "cuentaDestino": 529392355, // Número de cuenta del Cliente Mauro
4   "monto": 1000001,           // Monto a transferir
5   "moneda": "PESOS"          // Moneda de la transferencia
6 }
7
```

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

Pretty Raw Preview Visualize [JSON](#)

```
1 {
2   "errorCode": 4000,
3   "errorMessage": "La cuenta origen no tiene fondos suficientes."
4 }
```

Postbot  
Ctrl Alt P

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

El cliente quiere transferir un monto negativo.

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/api/transferencias`. The request body is a JSON object:

```
1 {  
2   "cuentaOrigen": 663823400, // Número de cuenta del Cliente Maria  
3   "cuentaDestino": 737389159, // Número de cuenta del Cliente Mauro  
4   "monto": -5,           // Monto a transferir  
5   "moneda": "PESOS"       // Moneda de la transferencia  
6 }  
7
```

The response status is **400 Bad Request**. The response body is:

```
1 {  
2   "errorCode": 4000,  
3   "errorMessage": "No se puede debitar una cantidad negativa."  
4 }
```

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

El cliente quiere hacer una transferencia en dólares, pero las cuentas son en pesos.

The screenshot shows the Postman interface with the following details:

- HTTP Method:** POST
- URL:** http://localhost:8080/api/transferencias
- Body:** JSON (selected)
- Request Body Content:**

```
1 {
2     "cuentaOrigen": 663823400, // Número de cuenta del Cliente Maria
3     "cuentaDestino": 737389159, // Número de cuenta del Cliente Mauro
4     "monto": 1,               // Monto a transferir
5     "moneda": "DOLARES"      // Moneda de la transferencia
6 }
7
```
- Response Status:** 400 Bad Request
- Response Body Content (Pretty JSON):**

```
1 {
2     "errorCode": 4000,
3     "errorMessage": "La cuenta destino no maneja la misma moneda que la transferencia."
4 }
```

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

Se quiere realizar una transferencia, pero la cuenta origen no existe.

The screenshot shows a POST request to <http://localhost:8080/api/transferencias>. The request body is a JSON object:

```
1 {  
2   "cuentaOrigen": 66382340, // Número de cuenta del Cliente María  
3   "cuentaDestino": 737389159, // Número de cuenta del Cliente Mauro  
4   "monto": 1, // Monto a transferir  
5   "moneda": "DOLARES" // Moneda de la transferencia  
6 }  
7
```

The response status is 400 Bad Request. The response body is:

```
1 {  
2   "errorCode": 4000,  
3   "errorMessage": "La cuenta origen no existe."  
4 }
```

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

Se quiere realizar una transferencia, pero la cuenta destino no existe.

The screenshot shows a Postman interface with the following details:

- Request URL:** `http://localhost:8080/api/transferencias`
- Method:** POST
- Body:** Raw JSON (selected)
- JSON Body Content:**

```
1 {
2     "cuentaOrigen": 699788613, // Número de cuenta del Cliente
3     "cuentaDestino": 2721925, // Número de cuenta del Cliente
4     "monto": 1111,           // Monto a transferir
5     "moneda": "PESOS"        // Moneda de la transferencia
6 }
7
```
- Response Status:** 400 Bad Request
- Response Body (Pretty JSON):**

```
1 {
2     "estado": "FALLIDA",
3     "mensaje": "La cuenta destino no existe."
4 }
```

GET: <http://localhost:8080/api/cuentas/{numeroCuenta}/movimientos>

Quiero ver los movimientos de una cuenta inexistente.

The screenshot shows a Postman request for the URL `http://localhost:8080/api/cuentas/699788613/movimientos`. The method is set to GET. The 'Params' tab is selected, showing a single parameter named 'Key'. The 'Body' tab is selected, displaying a JSON response with the following content:

```
1 {  
2   "estado": "ERROR",  
3   "mensaje": "La cuenta con número 699788613 no existe."  
4 }
```

The status bar at the bottom right indicates a **404 Not Found** error.