

# Documentación del proyecto Sistema Bancario

## Introducción

En este proyecto tenemos una API REST implementada en Java utilizando Spring Boot, hecha para gestionar operaciones bancarias en un sistema de banco. La API permite realizar operaciones de cliente, cuentas, depósitos y préstamos.

## Comando útiles

**Instalar dependencias del proyecto:** mvn install

**Ejecutar el proyecto:** mvn spring-boot:run

**Ejecutar test unitarios:** mvn clean test

## Endpoints

### Cliente:

- **Controlador:** ClienteController
- **Función:** crearCliente
- **Ruta:** /api/cliente
- **Método:** POST
- **Ejemplo de input:**

```
{  
  "nombre": "Diego",  
  "apellido": "Bruno",  
  "dni": 42958843,  
  "fechaNacimiento": "2000-09-30",  
  "tipoPersona": "F"  
}
```

**Cliente:**

- **Controlador:** ClienteController
- **Función:** buscarClientePorDni
- **Ruta:** /api/cliente/{dni}
- **Metodo:** GET
- **Ejemplo de output:**

```
{
  "dni": 42958843,
  "nombre": "Diego",
  "apellido": "Bruno",
  "fechaNacimiento": "2000-09-30",
  "fechaAltaCliente": "2024-12-10",
  "banco": "Banco UTN",
  "tipoPersona": "PERSONA_FISICA",
  "cuentasDelCliente": [],
  "edad": 24
}
```

**Cliente:**

- **Controlador:** ClienteController
- **Función:** updateCliente
- **Ruta:** /api/cliente/{dni}
- **Metodo:** PUT
- **Ejemplo de input:**

```
{
  "nombre": "Diego",
  "apellido": "Bruno2",
  "dni": 42958843,
  "fechaNacimiento": "2000-09-30",
  "tipoPersona": "F"
}
```

**Cuenta:**

- **Controlador:** CuentaController
- **Función:** addCuenta
- **Ruta:** /api/cuenta/agregar
- **Metodo:** POST
- **Ejemplo de input:**

```
{
  "dniTitular": 42958843,
  "tipoMoneda": "ARS",
}
```

```
"tipoCuenta": "CA"  
}
```

#### Cuenta:

- **Controlador:** CuentaController
- **Función:** allCuentasByCliente
- **Ruta:** /api/cuenta/all/{dni}
- **Método:** GET
- **Ejemplo de output:**

```
[  
  {  
    "numeroCuenta": 9197130947687054224,  
    "saldo": 0.0,  
    "fechaAltaCuenta": "2024-12-10",  
    "tipoMoneda": "ARS",  
    "tipoCuenta": "CAJA_AHORRO",  
    "movimientosCuenta": []  
  }  
]
```

#### Cuenta:

- **Controlador:** CuentaController
- **Función:** depositarDinero
- **Ruta:** /api/cuenta/depositar
- **Método:** POST
- **Ejemplo de input:**

```
{  
  "numeroCuenta": "9197130947687054224",  
  "monto": 10000  
}
```

#### Préstamo:

- **Controlador:** PrestamoController
- **Función:** solicitarPrestamo
- **Ruta:** /api/prestamo
- **Método:** POST
- **Ejemplo de input:**

```
{
  "numeroCliente": 42958843,
  "plazoMeses": 3,
  "montoPrestamo": 30000,
  "moneda": "ARS"
}
```

- **Ejemplo de output:**

```
{
  "estado": "APROBADO",
  "mensaje": "El monto del préstamo fue acreditado en su cuenta.",
  "planPagos": [
    {
      "cuotaNro": 1,
      "monto": 10125.0
    },
    {
      "cuotaNro": 2,
      "monto": 10125.0
    },
    {
      "cuotaNro": 3,
      "monto": 10125.0
    }
  ]
}
```

**Préstamo:**

- **Controlador:** PrestamoController
- **Función:** solicitarPrestamo
- **Ruta:** /api/prestamo/{dni}
- **Método:** GET
- **Ejemplo de output:**

```
{
  "numeroCliente": 42958843,
  "prestamos": [
    {
      "idPrestamo": 1,
      "monto": 30000.0,
      "plazoMeses": 3,
      "pagosRealizados": 0,
    }
  ]
}
```

```
        "saldoRestante": 30375.0
    }
]
}
```

### Préstamo:

- **Controlador:** PrestamoController
- **Función:** solicitarPrestamo
- **Ruta:** /api/prestamo/pagar
- **Metodo:** POST
- **Ejemplo de input:**

```
{
  "numeroPrestamo": 1,
  "cantidadCuotas": 1,
  "montoPago": 10125
}
```

- **Ejemplo de output:**

```
{
  "numeroPrestamo": 1,
  "estado": "COMPLETADO",
  "mensaje": "El pago de 1 cuota(s) fue registrado exitosamente. Saldo restante: 20250.0"
}
```

## Manejo de errores

Aquí mostraré algunas salidas para demostrar como se muestran los errores a la hora de la ejecución del programa

### Cliente ya tiene cuenta en el banco

```
{
  "errorMessage": "El cliente con dni 42958843 ya es usuario de nuestro banco",
  "errorCode": 8000
}
```

### Cliente es menor de edad

```
{
  "errorMessage": "El cliente no puede ser menor a 18 años",
  "errorCode": 6000
}
```

### Apellido o nombre no pueden contener caracteres numericos

```
{
  "errorMessage": "El campo apellido no puede contener caracteres numericos",
  "errorCode": 6003
}
```

### Cuenta de tipo Cuenta Corriente no soporta USD como moneda

```
{
  "errorMessage": "La cuenta de tipo CUENTA_CORRIENTE y tipo moneda USD no es soportada",
  "errorCode": 6002
}
```

### Cliente no existe

```
{
  "errorMessage": "El cliente con dni 42958841 no existe",
  "errorCode": 7001
}
```

### Cliente no tiene cuentas

```
{
  "errorMessage": "El cliente no tiene cuentas",
  "errorCode": 7003
}
```

### No existe la cuenta con número “x”

```
{
  "errorMessage": "No existe cuenta con el número 9197130947687054224",
  "errorCode": 7003
}
```

### Cliente no tiene cuenta en la moneda del prestamo

```
{
  "errorMessage": "El cliente no tiene una cuenta en la moneda especificada.",
  "errorCode": 7003
}
```

### Plazo invalido en préstamo

```
{
  "errorMessage": "El plazo debe ser mayor a 0.",
  "errorCode": 8004
}
```

UTN TUP - Laboratorio III

Bruno Diego

### No tiene suficiente saldo en la cuenta para pagar cuota

```
{  
  "errorMessage": "No tiene suficiente saldo en su cuenta para realizar este pago.",  
  "errorCode": 6003  
}
```