

API Reference

# Banco

1.0: Banco

API de un banco para realizar operaciones sobre una cuenta

# INDEX

1. AUTENTICACION	3
1.1 POST /signUp	3
1.2 POST /login	4
2. CUENTA	4
2.1 POST /ingresar	5
2.2 POST /transferir	5
2.3 POST /comprarDolares	7
2.4 GET /cuenta	7

# API

## 1. AUTENTICACION

### 1.1 POST /signUp

Permite registrar un usuario y obtener una cuenta

#### PATH PARAMS

NAME	TYPE	REQUIRED	EXAMPLE	DESCRIPTION
------	------	----------	---------	-------------

#### REQUEST

REQUEST BODY - application/json

```
{
  username string required
  password string required
  email    string
}
```

#### EXAMPLE

```
{
  "username": "messiElLindo",
  "password": "18122023",
  "email": "string"
}
```

#### RESPONSES

STATUS CODE - 200: OK. Se creo el usuario y la cuenta vinculada

RESPONSE MODEL - application/json

```
{
  El usuario con la contraseña encriptada
  _id      string  El id generada
  username string
  password string  Contraseña encriptada
}
```

#### EXAMPLE

```
{
  "_id": "6552e4cf1c88bebc6179b7fa",
  "username": "messiElLindo",
  "password": "$2a$10$vlzBrEcmUi/AbJYiKdO.ru1r.gjneiY8CoLJUaEqI9997lgKTPHDO"
}
```

STATUS CODE - 400: BAD REQUEST

RESPONSE MODEL - application/json

```
{
  Descripcion de error
  error string
}
```

#### EXAMPLE

```
{
  "error": "Ya existe usuario con ese nombre"
}
```

## 1.2 POST /login

Permite obtener un token

#### PATH PARAMS

NAME	TYPE	REQUIRED	EXAMPLE	DESCRIPTION
------	------	----------	---------	-------------

#### REQUEST

REQUEST BODY - application/json

```
{
  username string required
  password string required
}
```

#### EXAMPLE

```
{
  "username": "messiElLindo",
  "password": "18122023"
}
```

#### RESPONSES

STATUS CODE - 200: OK. Se autenticó con éxito

RESPONSE MODEL - application/json

```
{
  token string El token generado
}
```

#### EXAMPLE

```
{
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkYXRhIjp7InVzZXkiOiI2NTU2YmU1ZDEyZGFjOGYyZmYyY2I"
}
```

STATUS CODE - 400: BAD REQUEST

RESPONSE MODEL - application/json

```
{
  Descripcion de error
  error string
}
```

#### EXAMPLE

```
{
  "error": "Credenciales inválidas"
}
```

## 2. CUENTA

## 2.1 POST /ingresar

Permite ingresar dinero a la cuenta

### PATH PARAMS

NAME	TYPE	REQUIRED	EXAMPLE	DESCRIPTION
------	------	----------	---------	-------------

### REQUEST

REQUEST BODY - application/json

```
{
  monto      number
  descripcion string
}
```

#### EXAMPLE

```
{
  "monto": 10000,
  "descripcion": "motivo"
}
```

### RESPONSES

STATUS CODE - 200: OK. Se ingreso el monto a la cuenta

RESPONSE MODEL - application/json

```
{
  monto      number
  moneda     string
  descripcion string
}
```

#### EXAMPLE

```
{
  "monto": 5000,
  "moneda": "$",
  "descripcion": "Ingreso de dinero"
}
```

STATUS CODE - 400: BAD REQUEST

RESPONSE MODEL - application/json

```
{
  Descripcion de error
  error string
}
```

#### EXAMPLE

```
{
  "error": "\"monto\" must be a positive number"
}
```

## 2.2 POST /transferir

Permite transferir plata a una cuenta

PATH PARAMS

NAME	TYPE	REQUIRED	EXAMPLE	DESCRIPTION
------	------	----------	---------	-------------

REQUEST

REQUEST BODY - application/json

```
ANY OF
OPTION 1
{
  monto number required
  alias string required
  cbu string
}
OPTION 2
{
  monto number required
  alias string
  cbu string required
}
```

EXAMPLE

```
{}
```

RESPONSES

STATUS CODE - 200: OK. Se transfirio el monto

RESPONSE MODEL - application/json

```
{
  monto number
  moneda string
  destinatario string
  descripcion string
}
```

EXAMPLE

```
{
  "monto": 5000,
  "moneda": "$",
  "destinatario": "1213301834242795152114",
  "descripcion": "Transferencia saliente a 1213301834242795152114"
}
```

STATUS CODE - 400: BAD REQUEST

RESPONSE MODEL - application/json

```
{
  Descripcion de error
  error string
}
```

EXAMPLE

```
{
  "error": "Cuenta destino no existente"
}
```

## 2.3 POST /comprarDolares

Comprar dólares con el dinero de la cuenta

### PATH PARAMS

NAME	TYPE	REQUIRED	EXAMPLE	DESCRIPTION
------	------	----------	---------	-------------

### REQUEST

REQUEST BODY - application/json

```
{
  monto      number  required
  descripcion string  required
}
```

### EXAMPLE

```
{
  "monto": 100,
  "descripcion": "motivo"
}
```

### RESPONSES

STATUS CODE - 200: (OK) el monto fue convertido a dolares || "Saldo Insuficiente"

STATUS CODE - 400: Bad Request

STATUS CODE - 500: Error en el Servidor

## 2.4 GET /cuenta

Permite obtener datos de la cuenta

### PATH PARAMS

NAME	TYPE	REQUIRED	EXAMPLE	DESCRIPTION
------	------	----------	---------	-------------

### RESPONSES

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  _id          string
  usuario      string
  saldoPesos   number
  saldoDolares number
  alias        string
  cbu          string
  movimientos [
    {
      monto      number
      moneda     string
      descripcion string
      destinatario string  CBU en caso de transferencia
    }
  ]
}
```

```

    createdAt    string
  }
]
}

```

#### EXAMPLE

```

{
  "_id": "6556be5d12dac8f2ff2caf92",
  "usuario": "6556be5d12dac8f2ff2caf90",
  "saldoPesos": 15000,
  "saldoDolares": 0,
  "alias": "THESE.SALE.YOUTH",
  "cbu": "1213301834242795152114",
  "movimientos": [
    {
      "monto": 10000,
      "moneda": "$",
      "descripcion": "Prueba",
      "_id": "6556be8a12dac8f2ff2caf96",
      "createdAt": "2023-11-17T01:14:50.642Z"
    },
    {
      "monto": -100,
      "moneda": "$",
      "destinatario": "2701202381676402620197",
      "descripcion": "Transferencia saliente a 2701202381676402620197",
      "_id": "6556beba12dac8f2ff2caf9e",
      "createdAt": "2023-11-17T01:15:38.013Z"
    },
    {
      "monto": 10000,
      "moneda": "$",
      "descripcion": "motivo",
      "_id": "655d71f8e124ae85304bc575",
      "createdAt": "2023-11-22T03:14:00.100Z"
    },
    {
      "monto": -10000,
      "moneda": "$",
      "destinatario": "0987942627514195265899",
      "descripcion": "Transferencia saliente a 0987942627514195265899",
      "_id": "655d7212e124ae85304bc57f",
      "createdAt": "2023-11-22T03:14:26.009Z"
    },
    {
      "monto": 5000,
      "moneda": "$",
      "descripcion": "Ingreso de dinero",
      "_id": "655e9d88c8b577736d4ab45b",
      "createdAt": "2023-11-23T00:32:08.583Z"
    },
    {
      "monto": -100,
      "moneda": "$",

```



```

"destinatario": "2701202381676402620197",
"descripcion": "Transferencia saliente a 2701202381676402620197",
"_id": "655e9dd2c8b577736d4ab468",
"createdAt": "2023-11-23T00:33:22.702Z"
},
{
  "monto": -100,
  "moneda": "$",
  "destinatario": "0987942627514195265899",
  "descripcion": "Transferencia saliente a 0987942627514195265899",
  "_id": "655e9de7c8b577736d4ab478",
  "createdAt": "2023-11-23T00:33:43.273Z"
},
{
  "monto": -100,
  "moneda": "$",
  "destinatario": "2701202381676402620197",
  "descripcion": "Transferencia saliente a 2701202381676402620197",
  "_id": "655e9e2dc8b577736d4ab494",
  "createdAt": "2023-11-23T00:34:53.005Z"
},
{
  "monto": -100,
  "moneda": "$",
  "destinatario": "1213301834242795152114",
  "descripcion": "Transferencia saliente a 1213301834242795152114",
  "_id": "655e9eb03276176c9c9da05a",
  "createdAt": "2023-11-23T00:37:04.375Z"
},
{
  "monto": 100,
  "moneda": "$",
  "descripcion": "Transferencia entrante de 1213301834242795152114",
  "_id": "655e9eb03276176c9c9da05b",
  "createdAt": "2023-11-23T00:37:04.376Z"
},
{
  "monto": -100,
  "moneda": "$",
  "destinatario": "1213301834242795152114",
  "descripcion": "Transferencia saliente a 1213301834242795152114",
  "_id": "655e9f283276176c9c9da074",
  "createdAt": "2023-11-23T00:39:04.409Z"
},
{
  "monto": 100,
  "moneda": "$",
  "descripcion": "Transferencia entrante de 1213301834242795152114",
  "_id": "655e9f283276176c9c9da075",
  "createdAt": "2023-11-23T00:39:04.409Z"
},
{
  "monto": -100,

```

```

    "moneda": "$",
    "destinatario": "1213301834242795152114",
    "descripcion": "Transferencia saliente a 1213301834242795152114",
    "_id": "655e9f393276176c9c9da093",
    "createdAt": "2023-11-23T00:39:21.739Z"
  },
  {
    "monto": 100,
    "moneda": "$",
    "descripcion": "Transferencia entrante de 1213301834242795152114",
    "_id": "655e9f393276176c9c9da094",
    "createdAt": "2023-11-23T00:39:21.739Z"
  },
  {
    "monto": -100,
    "moneda": "$",
    "destinatario": "1213301834242795152114",
    "descripcion": "Transferencia saliente a 1213301834242795152114",
    "_id": "655e9f53daa055746afd6866",
    "createdAt": "2023-11-23T00:39:47.982Z"
  },
  {
    "monto": 100,
    "moneda": "$",
    "descripcion": "Transferencia entrante de 1213301834242795152114",
    "_id": "655e9f53daa055746afd6867",
    "createdAt": "2023-11-23T00:39:47.983Z"
  }
]
}

```

## Security and Authentication

### SECURITY SCHEMES

KEY	TYPE	DESCRIPTION
JWT	http, bearer, JWT	

# API List

## Autenticacion

METHOD	API
post	/signUp
post	/login

## Cuenta

METHOD	API
post	/ingresar
post	/transferir
post	/comprarDolares
get	/cuenta