

## Desafío - Mi Banco

En este desafío validaremos nuestros conocimientos de Transacciones, Captura de errores en transacciones.

Lee todo el documento antes de comenzar el desarrollo **grupal**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

### Descripción

La empresa Mi Banco SPA está recién registrada y está en búsqueda de un desarrollador full stack developer que empiece a crear su sistema de transacciones. Previamente los dueños de esta empresa habían intentado crear una institución bancaria pero el software que compraron tenía vulnerabilidades y permitía hacer transacciones sin restricciones, por lo que están exigiendo de extrema necesidad controlar efectivamente los movimientos bancarios para no generar balances negativos y consecuencia con sus clientes.

En este desafío deberás realizar una aplicación Node que se conecte con PostgreSQL, utilice transacciones para simular el comportamiento de una transacción bancaria.

## Consideraciones y recomendaciones

Para la solución de este desafío, necesitarás:

- Crear una base de datos llamada **banco**.

```
CREATE DATABASE Banco;
```

- Crear una tabla **transferencias**.

```
CREATE TABLE transferencias  
(descripcion varchar(50), fecha varchar(10), monto DECIMAL,  
cuenta_origen INT, cuenta_destino INT);
```

- Crear una tabla **cuentas**.

```
CREATE TABLE cuentas (id INT, saldo DECIMAL CHECK (saldo >= 0) );
```

- Registrar por lo menos 2 cuentas en la tabla **cuentas** con un saldo inicial.

```
INSERT INTO cuentas values (1, 20000);  
INSERT INTO cuentas values (2, 10000);
```

Utiliza los argumentos de la línea de comando para definir los valores que usarán tus consultas SQL y la función a ejecutar.

## Requerimientos

1. Crear una función asíncrona que registre una nueva transferencia utilizando una transacción SQL. Debe mostrar por consola la última transferencia registrada.  
**(3 Puntos)**
2. Realizar una función asíncrona que consulte la tabla de transferencias y retorne los últimos 10 registros de una cuenta en específico.  
**(3 Puntos)**
3. Realizar una función asíncrona que consulte el saldo de una cuenta en específico.  
**(2 Puntos)**
4. En caso de haber un error en una transacción SQL, se debe retornar el error por consola.  
**(2 Puntos)**



¡Mucho éxito!