

# EVALUACIÓN MÓDULO 07

---

## Descripción

El Banco Solar acaba de decidir invertir una importante suma de dinero para contratar un equipo de desarrolladores Full Stack que desarrollen un nuevo sistema de transferencias, y han anunciado que todo aquel que postule al cargo debe realizar un servidor con Node que utilice PostgreSQL para la gestión y persistencia de datos, y simular un sistema de transferencias.

El sistema debe permitir registrar nuevos usuarios con un balance inicial y basados en éstos, realizar transferencias de saldos entre ellos.

En este ejercicio contará con una aplicación cliente preparada para consumir las rutas que deberá crear en el servidor. A continuación se muestra una imagen con la interfaz mencionada.

# EVALUACIÓN MÓDULO 07

The screenshot shows the Banco Solar C.A. web application. At the top is the bank's logo and name. Below it are two main sections: 'Agregar Nuevo Usuario' and 'Realizar una Transferencia'. The 'Agregar Nuevo Usuario' section has input fields for 'Nombre' and 'Balance', and an 'Agregar' button. The 'Realizar una Transferencia' section has dropdown menus for 'Emisor' (Luis Vallejo) and 'Receptor' (Pedro Rivas), an input field for 'Monto', and a 'Realizar' button. Below these sections are two tables. The 'Tabla de usuarios' table lists two users: Pedro Rivas with a balance of 20,000 and Luis Vallejo with a balance of 40,000. Each user has 'Editar' and 'Eliminar' buttons. The 'Tabla de Transferencias' table lists four transactions with columns for 'Fecha', 'Emisor', 'Receptor', and 'Monto'.

Nombre	Balance		
Pedro Rivas	20.000	Editar	Eliminar
Luis Vallejo	40.000	Editar	Eliminar

Fecha	Emisor	Receptor	Monto
25/01/2021 10:00	Luis Vallejo	Pedro Rivas	60.000
25/01/2021 13:15	Pedro Rivas	Luis Vallejo	20.000
25/01/2021 17:25	Luis Vallejo	Pedro Rivas	30.000
25/01/2021 21:04	Pedro Rivas	Luis Vallejo	10.000

Las rutas que deberá crear son las siguientes:

- **/ GET:** Devuelve la aplicación cliente disponible en el apoyo de la prueba.
- **/usuario POST:** Recibe los datos de un nuevo usuario y los almacena en PostgreSQL.
- **/usuarios GET:** Devuelve todos los usuarios registrados con sus balances.
- **/usuario PUT:** Recibe los datos modificados de un usuario registrado y los actualiza.
- **/usuario DELETE:** Recibe el id de un usuario registrado y lo elimina.
- **/transferencia POST:** Recibe los datos para realizar una nueva transferencia. Se debe ocupar una transacción SQL en la consulta a la base de datos.

# EVALUACIÓN MÓDULO 07

- **/transferencias GET:** Devuelve todas las transferencias almacenadas en la base de datos en formato de arreglo.

Para iniciar con la persistencia de datos se deben ocupar las siguientes instrucciones SQL:

```
CREATE DATABASE bancosolar;
```

```
CREATE TABLE usuarios (id SERIAL PRIMARY KEY, nombre VARCHAR(50),  
balance FLOAT CHECK (balance >= 0));
```

```
CREATE TABLE transferencias (id SERIAL PRIMARY KEY, emisor INT, receptor  
INT, monto FLOAT, fecha TIMESTAMP, FOREIGN KEY (emisor) REFERENCES  
usuarios(id), FOREIGN KEY (receptor) REFERENCES usuarios(id));
```

## Requerimientos

1. Utilizar el paquete pg para conectarse a PostgreSQL y realizar consultas DML para la gestión y persistencia de datos.
2. Usar transacciones SQL para realizar el registro de las transferencias.
3. Servir una API RESTful en el servidor con los datos de los usuarios almacenados en PostgreSQL.



## EVALUACIÓN MÓDULO 07

4. Capturar los posibles errores que puedan ocurrir a través de bloques catch o parámetros de funciones callbacks para condicionar las funciones del servidor.
5. Devolver correctamente los códigos de estado según las diferentes situaciones.