Documentación Breve del Proyecto

1. Descripción del Proyecto:

El proyecto es una aplicación de banco en línea que permite a los usuarios realizar operaciones bancarias como consultas de saldo, transferencias, retiros, y visualización de transacciones. La aplicación consta de un frontend desarrollado en React y un backend construido con Node.js y Express. La comunicación entre el frontend y el backend se realiza a través de solicitudes HTTP utilizando el cliente Axios.

2. Tecnologías Utilizadas:

Frontend (React):

* Se eligió React como el framework de frontend debido a su flexibilidad, rendimiento y capacidad para construir interfaces de usuario interactivas.
* Se utilizó el sistema de componentes de React para organizar y reutilizar el código de manera eficiente.
* Axios se empleó para manejar las solicitudes HTTP desde el frontend hacia el backend.
* Material UI se empleo para diferentes usos en los componentes como lo fueron las alertas, iconos y modales

Backend (Node.js y Express):

* Node.js fue seleccionado para el desarrollo del backend debido a su naturaleza basada en eventos y capacidad para manejar múltiples conexiones de manera eficiente.
* Express.js se utilizó como el marco web de Node.js para simplificar el enrutamiento y la gestión de solicitudes.
* Se implementaron modelos de datos con Mongoose, una biblioteca de modelado de objetos para MongoDB, para interactuar con la base de datos.

Base de Datos (MongoDB):

* MongoDB fue elegida como la base de datos principal debido a su flexibilidad en el almacenamiento de datos no estructurados y su integración fluida con aplicaciones Node.js a través de Mongoose.
* Se creó un modelo de datos para las transacciones utilizando Mongoose, que permite realizar operaciones de base de datos de manera sencilla.

Estilo y Diseño:

* Se utilizó CSS para el diseño y estilo de la aplicación.
* Se aplicaron técnicas de diseño responsivo para garantizar una experiencia de usuario consistente en diferentes dispositivos.

Seguridad:

* Se implementaron medidas de seguridad, como la autenticación de usuarios utilizando tokens JWT (JSON Web Tokens) para proteger las rutas sensibles.
* Se manejaron errores y excepciones de manera adecuada para evitar la exposición de información sensible.

3. Solución a Problemas Específicos:

Manejo de CORS:

* Se configuró CORS en el backend para permitir solicitudes desde el frontend, abordando el problema de CORS mediante la configuración adecuada de encabezados.

Orden de Transacciones:

* Se ajustó la lógica del servidor para que las transacciones se devuelvan en el orden en que fueron creadas, facilitando la presentación de la información más reciente primero en el frontend.

Problemas de Transferencia:

* Se resolvieron problemas relacionados con transferencias y actualización de saldos al realizar transferencias entre usuarios.

4. Mejoras Futuras:

* Implementación de más características, como histórico detallado de transacciones y filtrado.
* Añadir validaciones adicionales y manejo de errores más robusto.
* Mejorar la interfaz de usuario y agregar funciones de administración de cuentas.