

01

Requerimientos -

Se desea construir una app de transacciones digital, que valide pagos a través de una interfaz QR para poder realizar pagos digitales.

02

Frontend: React Native

Setup del Proyecto:

- Configura un nuevo proyecto de React Native usando ``npx react-native init DigitalWalletDemo``.
- Instala librerías necesarias como ``react-navigation`` para la navegación y ``react-native-camera`` o ``react-native-qrcode-scanner`` para escanear códigos QR.

Pantallas Principales

Home Screen: Muestra el saldo actual del usuario y un botón para escanear el código QR.

- **Payment Screen:** Muestra la información del pago a realizar después de escanear el QR, con el monto a pagar y un botón para confirmar el pago.

03

Flujo de Trabajo

- En la Home Screen, el usuario puede presionar un botón para abrir la cámara y escanear un código QR.
- Al escanear el código, se redirige al usuario a la Payment Screen, donde se muestra el monto que debe pagar.
- El usuario confirma el pago, y se actualiza el saldo.

04

Backend: Node.js

1. Setup del Proyecto:

- Crea un nuevo proyecto Node.js usando ``npm init`` y ``express`` para manejar las rutas del servidor.

Opcional login:

- Usa ``jsonwebtoken`` para manejar la autenticación de usuarios.

- Utiliza ``qrcode`` para generar códigos QR.

2. Rutas API:

- **GET /generate-qr:** Genera un código QR que contiene información como el monto a pagar y el ID de la transacción.

- **POST /process-payment:** Recibe el ID de la transacción desde la app móvil y procesa el pago, actualizando la base de datos con la nueva información.

3. Flujo de Trabajo:

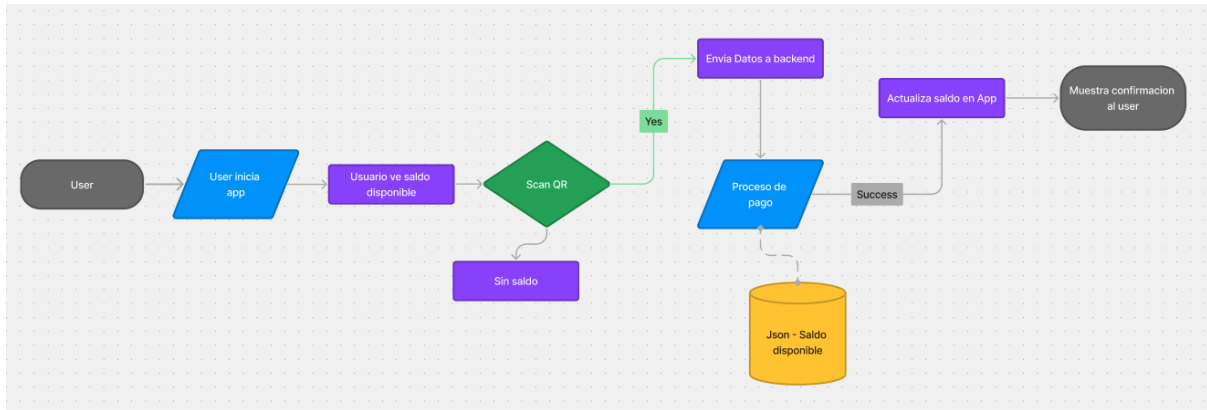
- Cuando se solicita un pago, el backend genera un código QR con los detalles de la transacción.

- La app móvil escanea el código QR y envía los detalles al backend para procesar el pago.

- El backend responde con el resultado del pago (éxito o error) y la app lo muestra al usuario.

Licenciatura en Sistemas de Información Arq. Programación Móvil - 2023

Diagramas de Flujo:



class

