

Prototipo de Pago de Pasajes con Tarjeta Magnética y Código QR

1. Introducción

El transporte público en muchas ciudades aún depende de métodos tradicionales de pago, como el dinero en efectivo o tarjetas con tecnología costosa. Este proyecto propone una solución innovadora y de bajo presupuesto basada en **Arduino y códigos QR** para validar y procesar pagos electrónicos en autobuses.

2. Objetivo General

Desarrollar un **prototipo funcional** para el pago de pasajes en autobuses utilizando un **Arduino como validador de códigos QR**, los cuales estarán vinculados a cuentas de usuario mediante una aplicación móvil.

3. Objetivos Específicos

- Diseñar e implementar una **aplicación móvil** que genere códigos QR únicos para cada usuario.
- Desarrollar un **sistema de validación de pasajes** con un **Arduino y un módulo de escaneo QR**.
- Integrar una **base de datos** para asociar cada QR con la cuenta del usuario y registrar transacciones.
- Evaluar la **viabilidad y costos** del sistema en comparación con tecnologías existentes.

4. Descripción del Prototipo

El sistema estará compuesto por los siguientes elementos:

a) Dispositivo validador (Arduino)

- **Hardware:**
 - Arduino (modelo compatible con módulos de comunicación).
 - Módulo lector de **códigos QR** (como el **Zebra SE-1224** o **EM18**).
 - Conexión a una base de datos a través de **Wi-Fi o Bluetooth** (ESP8266 o ESP32).
 - Pantalla LCD para mostrar el estado de la validación.
- **Funcionamiento:**
 1. El usuario sube al autobús y escanea su **código QR** en el lector.
 2. El Arduino consulta la base de datos para verificar la validez del código.
 3. Si el QR es válido y el usuario tiene saldo suficiente, la pantalla mostrará un mensaje de **"Acceso aprobado"** y se descontará el pasaje.
 4. Si el saldo es insuficiente o el QR es inválido, el sistema indicará un **"Error"**.

b) Aplicación móvil

- Permite a los usuarios **registrarse, generar códigos QR únicos y recargar saldo**.
- Se conecta con una **base de datos en la nube** para gestionar información de los pasajeros.
- Puede incluir métodos de pago como **tarjeta de crédito, débito o billeteras digitales**.

c) Base de datos y backend

- Base de datos **centralizada** para almacenar usuarios, saldos y transacciones.
- Backend desarrollado en **Node.js con Express** y base de datos en **Firebase o MySQL**.

5. Beneficios del Proyecto

- **Bajo costo** en comparación con los sistemas de pago con tarjeta RFID.
- Mayor **accesibilidad**, ya que cualquier usuario con un smartphone puede generar un QR.
- Facilita la integración con **otros sistemas de pago**.
- **Seguridad** y control de acceso eficiente sin necesidad de contacto físico.

6. Tecnologías a Utilizar

- **Hardware:** Arduino, lector QR, módulo Wi-Fi/Bluetooth.
- **Software:** Aplicación en Flutter o React Native, backend en Node.js, base de datos Firebase/MySQL.
- **Comunicación:** API REST para consulta y validación de QR.

7. Implementación del Prototipo en el Sistema Real

7.1 Consideraciones para la Integración

Para llevar el prototipo a un entorno real, se deben analizar diversos factores técnicos, financieros y operativos. La implementación en el sistema de transporte público requiere pruebas piloto, ajustes en infraestructura y aceptación por parte de usuarios y operadores.

7.2 Desafíos y Soluciones

Desafío	Posible Solución
Conectividad en autobuses	Uso de módulos ESP32 con Wi-Fi y Bluetooth para sincronización de datos en tiempo real.
Velocidad de validación	Optimización del código en Arduino para reducir el tiempo de respuesta del escáner QR.
Seguridad de pagos	Implementación de cifrado en la base de datos y autenticación de usuario en la app.
Falsificación de códigos QR	Generación de códigos dinámicos con vencimiento o uso de autenticación en dos pasos.
Aceptación del sistema por parte de los usuarios	Campañas de concientización y programas de prueba gratuitos para incentivar su uso.

7.3 Etapas de Implementación

1. Prueba piloto:
- Implementación en **un grupo reducido de autobuses** en una ruta específica.
 - Monitoreo del rendimiento del lector QR y respuesta de los usuarios.

- Recolección de feedback para realizar mejoras.

2. Expansión progresiva:

- Ajustes en el hardware y software basados en resultados de la prueba piloto.
- Capacitación a conductores y operadores sobre el uso del sistema.
- Implementación en un mayor número de unidades de transporte.

3. Escalabilidad y mantenimiento:

- Creación de una **infraestructura en la nube** para almacenar datos de viajes y transacciones.
- Mantenimiento periódico del hardware y actualizaciones de software.
- Posible integración con **otras formas de pago** (NFC, tarjetas bancarias, billeteras digitales).

7.4 Evaluación de Costos

Para garantizar la viabilidad del proyecto, es necesario analizar el **costo-beneficio** de la implementación. Se consideran los siguientes gastos:

Recurso	Costo Aproximado
Arduino + Módulo QR	\$50 - \$80 por unidad
Desarrollo de App Móvil	\$3,000 - \$7,000 (según funcionalidades)
Infraestructura en la nube	\$100 - \$500 mensuales
Capacitación y mantenimiento	Variable

El objetivo es demostrar que esta solución es **más económica** que los sistemas actuales basados en tarjetas magnéticas o NFC.

7.5 Impacto Esperado

- **Reducción de costos operativos** al eliminar tarjetas físicas costosas.
- **Mayor accesibilidad** para los usuarios sin necesidad de dispositivos especializados.
- **Mejor control financiero** para empresas de transporte mediante digitalización de pagos.
- **Menor contacto físico**, contribuyendo a la higiene y seguridad sanitaria.

7.6 Conclusión

La implementación del sistema de pago con códigos QR en transporte público es **técnicamente viable y financieramente accesible**. Con una estrategia de **prueba piloto, ajustes progresivos y escalabilidad**, se puede lograr una modernización efectiva del sistema sin incurrir en altos costos.

La clave del éxito radica en la **usabilidad del sistema**, la aceptación por parte de los pasajeros y la optimización del hardware y software para un desempeño fluido en condiciones reales. ✍