**Informe Desarrollo Móvil I**

**(Modas Americanas app)**

**Versiones del documento**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 17.09.2025 | 1.0 | Modas Americanas App, es un sistema de gestión de asistencias para tiendas que permite el control y registro de entrada y salida de empleados mediante códigos QR | 1. Carlos Ordoñez 2. Makdiel Tabora 3. Wilmer Sanchez |
|  |  |  |  |
|  |  |  |  |

Contenido

[Informe Desarrollo Móvil I 4](#_Toc169108811)

[1. DESCRIPCIÓN DE LA SOLUCIÓN 4](#_Toc169108812)

[1.1 Resumen gerencial 4](#_Toc169108813)

[1.2 Glosario 4](#_Toc169108814)

[1.3 Restricciones y metas de arquitectura 4](#_Toc169108815)

[1.4 Diagrama de Proceso 4](#_Toc169108816)

[1.5 Requerimientos del sistema 5](#_Toc169108817)

[1.5.1 Vista de Casos de uso de la aplicación 5](#_Toc169108818)

[2. Tecnología 5](#_Toc169108819)

[2.1 Desarrollo 5](#_Toc169108820)

[2.2 Modelo Entidad Relación 6](#_Toc169108821)

Informe Desarrollo Móvil I

# DESCRIPCIÓN DE LA SOLUCIÓN

## Resumen gerencial

Modas Americanas App es un sistema de gestión de asistencias para tiendas que permite el control y registro de entrada y salida de empleados mediante códigos QR. La aplicación móvil cuenta con tres tipos de usuarios: SuperAdmin (gestión global), Admin (gestión por tienda) y Empleado (registro personal). El sistema automatiza el proceso de control de asistencias, eliminando el registro manual y proporcionando reportes en tiempo real para la toma de decisiones gerenciales.

## Glosario

|  |  |
| --- | --- |
| Término | Definición |
| SuperAdmin | Usuario con permisos completos para gestionar tiendas y administradores del sistema |
| Admin | Usuario encargado de gestionar empleados y asistencias de una tienda específica |
| Empleado | Usuario que registra su asistencia mediante código QR personal |
| Código QR | Código bidimensional que contiene el ID único del empleado para registro de asistencia |
| Asistencia | Registro de entrada y salida de un empleado en una fecha específica |
| JWT | Token de seguridad utilizado para autenticación y autorización de usuarios |
| Api REST | Interfaz de programación que permite la comunicación entre el frontend y backend |

## Restricciones y metas de arquitectura

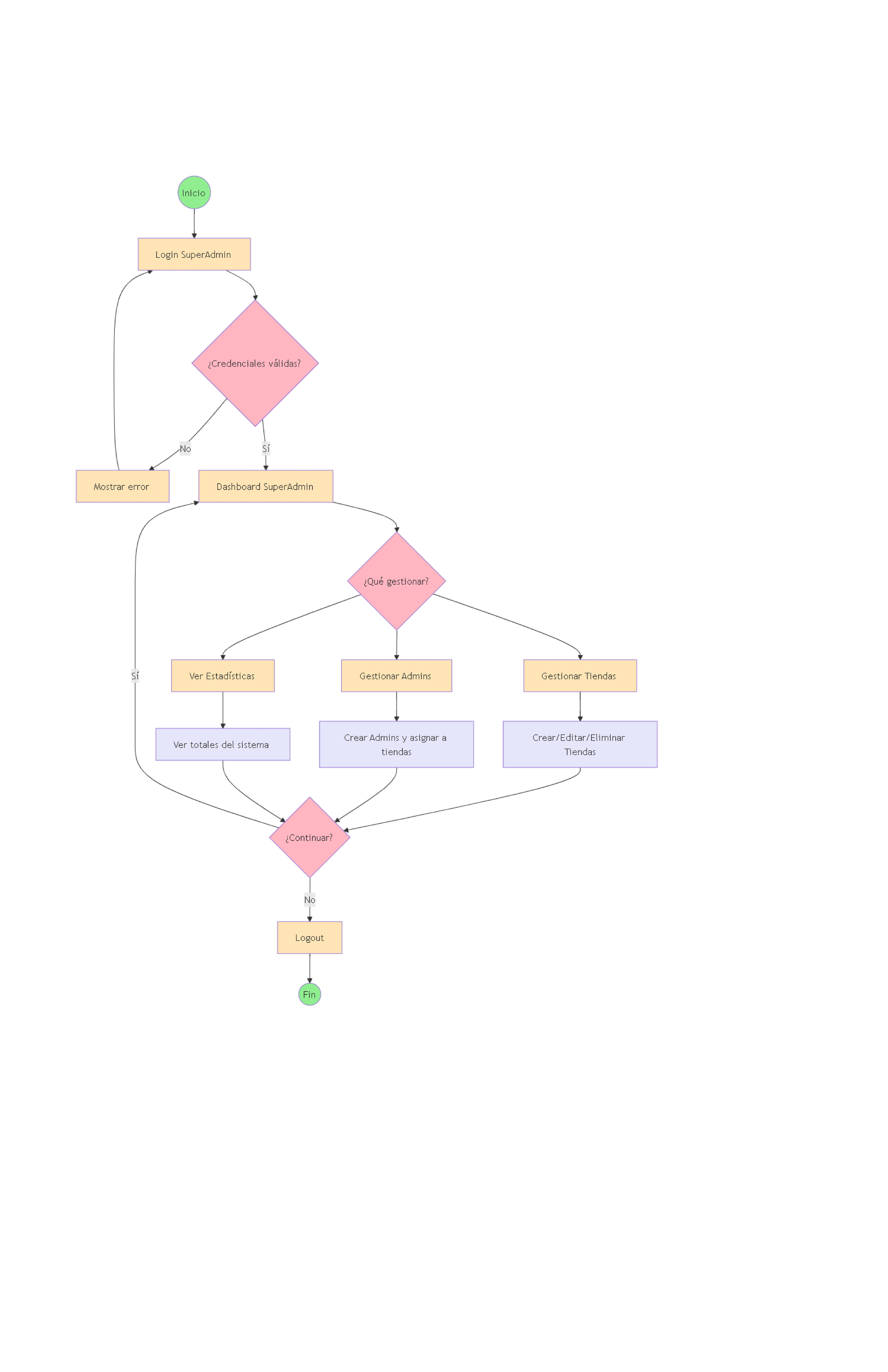
Restricciones:

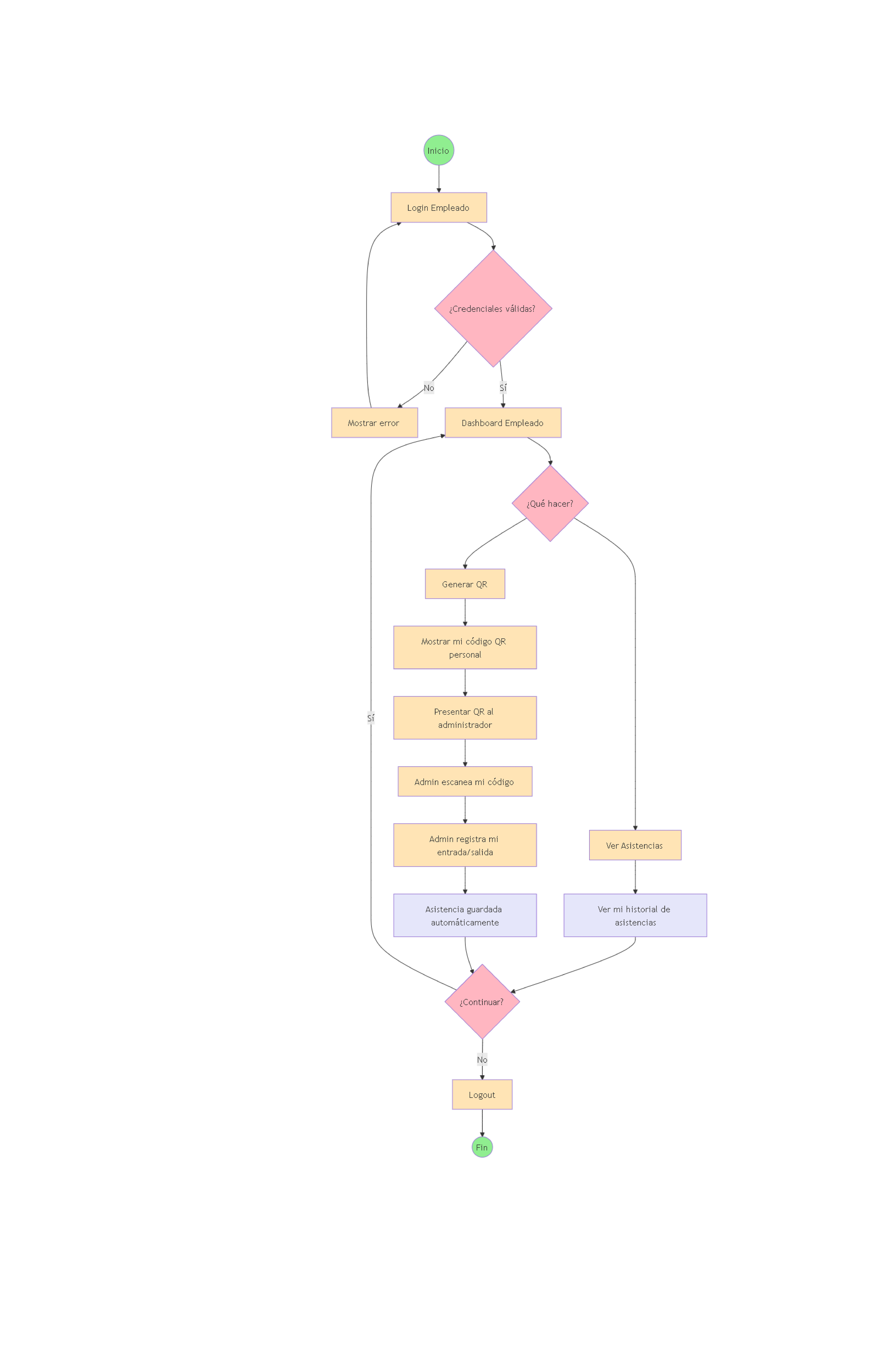
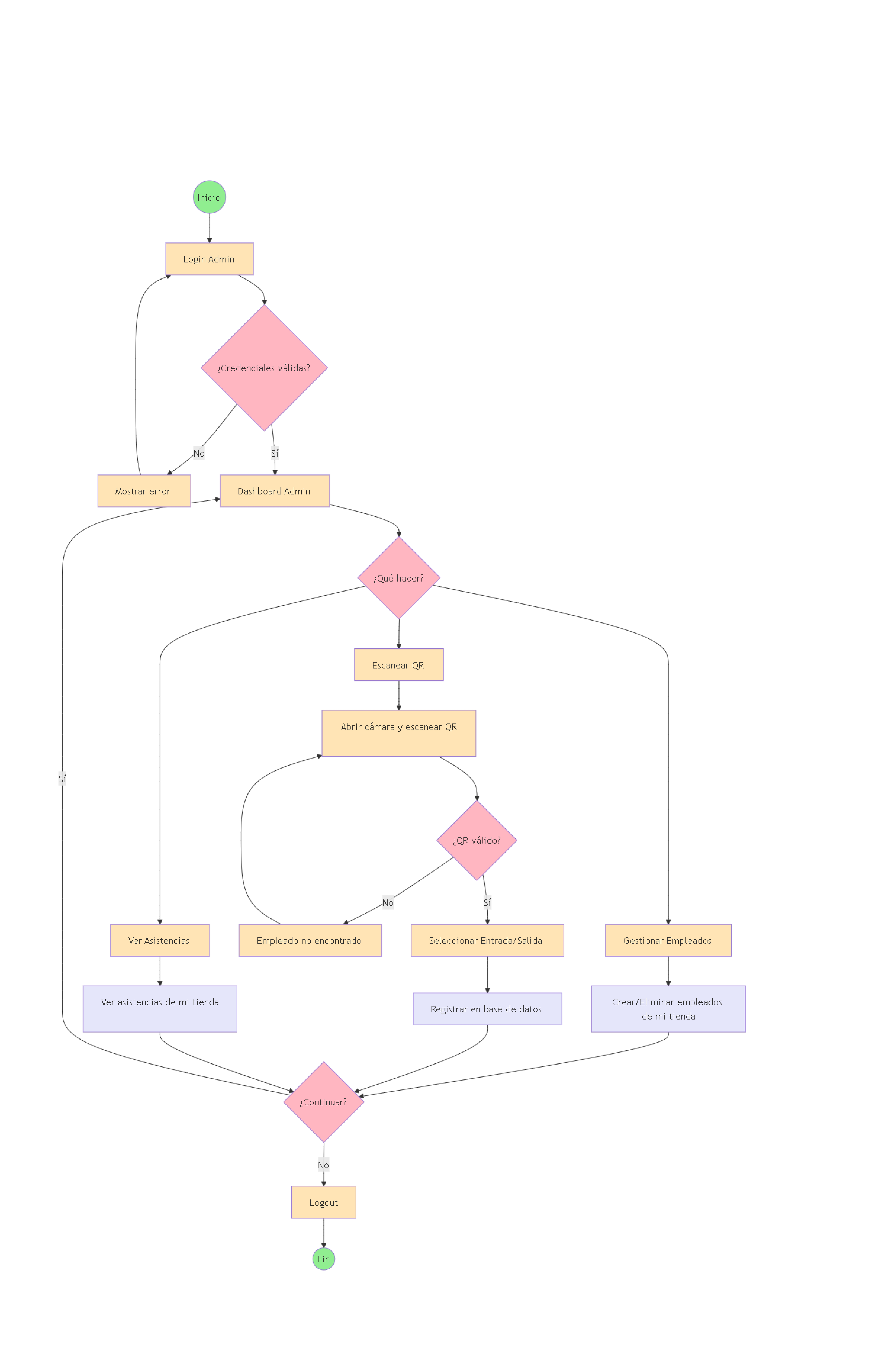
* La aplicación requiere acceso a cámara para escaneo de códigos QR
* Conexión a internet necesaria para sincronización con base de datos
* Dispositivos móviles con sistema operativo Android/iOS compatible con React Native
* Base de datos MySQL para almacenamiento de información

Metas de arquitectura:

* Arquitectura modular con separación clara entre frontend (React Native) y backend (Node.js)
* Sistema escalable que permita agregar múltiples tiendas sin afectar rendimiento
* Seguridad robusta mediante autenticación JWT y roles de usuario
* Interfaz intuitiva que requiera mínima capacitación para usuarios finales

## Diagrama de Proceso





## Requerimientos del sistema

## Vista de Casos de uso de la aplicación

|  |  |  |
| --- | --- | --- |
| Nombre del caso de uso | Descripción de la funcionalidad | Ruta del Caso de Uso |
| Inicio de sesión | Permite el acceso al sistema con email y contraseña según el rol del usuario | • Validar credenciales en base de datos  • Generar token JWT  • Redirigir a dashboard según rol (SuperAdmin/Admin/Empleado) |
| Gestión de tiendas | SuperAdmin puede crear, editar y eliminar tiendas del sistema | • Crear nueva tienda con datos básicos  • Asignar administradores a tiendas  • Modificar información de tiendas existentes  • Eliminar tiendas sin empleados asignados |
| Gestión de empleados | Admin puede crear y eliminar empleados de su tienda asignada | • Registrar nuevo empleado con credenciales  • Asignar empleado automáticamente a tienda del admin  • Eliminar empleados de la tienda |
| Registro de asistencia por QR | Admin escanea código QR del empleado para registrar entrada/salida | • Activar cámara del dispositivo  • Escanear código QR del empleado  • Validar ID de empleado  • Seleccionar tipo de registro (entrada/salida)  • Guardar asistencia con fecha y hora actual |
| Visualización de asistencias | Usuarios pueden consultar registros de asistencia según su rol | • Empleado: ver su historial personal  • Admin: ver asistencias de empleados de su tienda  • SuperAdmin: acceso a estadísticas globales |
| Generación de código QR | Empleado genera su código QR personal para registro de asistencia | • Mostrar código QR con ID de empleado  • Código accesible desde dispositivo móvil  • Presentar código al administrador para escaneo |

# Tecnología

## Desarrollo

**React Native**: Framework multiplataforma que permite desarrollar aplicaciones móviles nativas para iOS y Android usando JavaScript y React. Proporciona componentes nativos y APIs que se comunican directamente con el sistema operativo.

**Node.js**: Entorno de ejecución de JavaScript del lado del servidor que permite crear APIs REST escalables. Utiliza un modelo de I/O no bloqueante que lo hace eficiente para aplicaciones en tiempo real.

**Express.js**: Framework web minimalista para Node.js que facilita la creación de APIs REST con middleware para autenticación, CORS y manejo de errores.

**MySQL**: Sistema de gestión de base de datos relacional que almacena la información de usuarios, tiendas y asistencias con integridad referencial y transacciones ACID.

**JWT (JSON Web Token):** Estándar para transmitir información de manera segura entre cliente y servidor. Contiene la información del usuario y permisos codificados que se validan en cada petición.

**Expo Camera**: Biblioteca que proporciona acceso nativo a la cámara del dispositivo con capacidades de escaneo de códigos QR integradas.

**bcryptjs**: Librería para el hash seguro de contraseñas que protege las credenciales de usuarios almacenadas en la base de datos. es una plataforma de desarrollo versátil y potente que permite a los desarrolladores crear una amplia variedad de aplicaciones para diferentes sistemas operativos y dispositivos.

**Api Rest**: conjunto de principios arquitectónicos para el diseño de sistemas de software distribuidos. Estos principios se centran en la creación de servicios web que sean escalables, interoperables, y fáciles de mantener.

## 

## Modelo Entidad Relación

