



DOCUMENTO TECNICO

Sistema de Control de Asistencia Digital

Tecnologia, funcionalidades, instalacion y despliegue en la nube

React + Vite

Node.js + Express

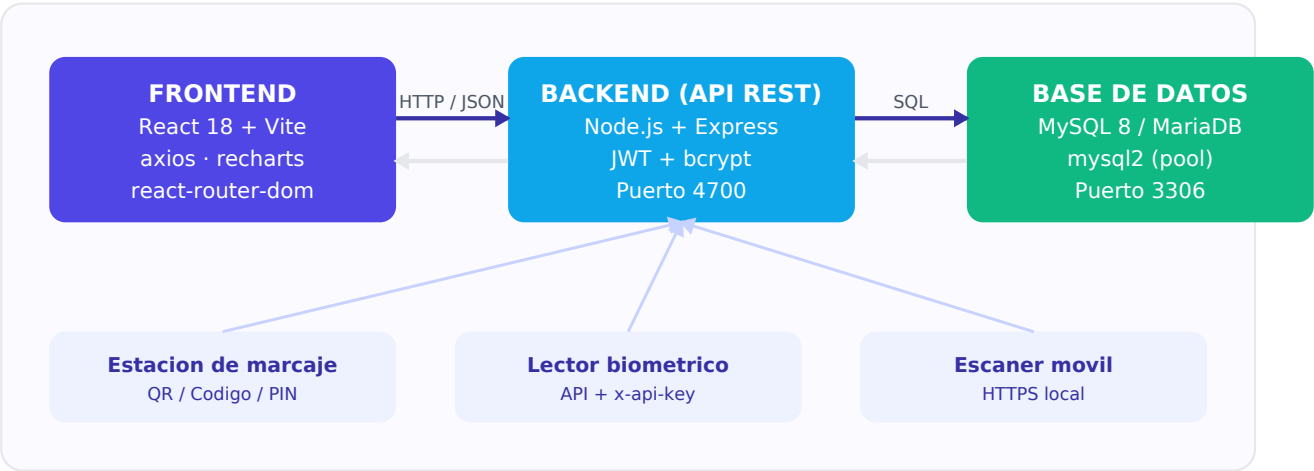
MySQL

JWT / bcrypt

Preparado para: Vito
Fecha: 01 de julio de 2026

1. Tecnologia con la que ha sido desarrollado el sistema

El sistema es una aplicacion web full-stack dividida en dos proyectos independientes (frontend y backend) que se comunican mediante una API REST. A continuacion se muestra la arquitectura general:



Capa	Tecnologia	Uso en el proyecto
Frontend	React 18 + Vite 5	Interfaz de usuario (SPA), recarga instantanea en desarrollo
	axios	Cliente HTTP para consumir la API
	react-router-dom	Enrutamiento entre paginas
	recharts	Graficos del dashboard
	html5-qrcode, qrcode, jsbarcode	Lectura y generacion de QR / codigo de barras
Backend	lucide-react	Iconografia de la interfaz
	Node.js + Express 4	API REST, logica de negocio y rutas
	mysql2	Conexion y consultas a la base de datos (pool)
	jsonwebtoken (JWT)	Autenticacion de sesion (token 8h)
	bcryptjs	Cifrado de contraseñas
Base de datos	dotenv, cors, nodemon	Configuracion, seguridad y recarga en desarrollo
	MySQL 8 / MariaDB 10.4+	Persistencia de empleados, asistencia, permisos, etc.

Seguridad y control de acceso

JWT

Token de sesion,
expira en 8h

bcrypt

Contrasenas cifradas
(hash)

Roles

admin / rrhh /
empleado

Roles disponibles: administrador (acceso total, incluye eliminar registros), recursos humanos (gestion de personal y aprobacion de solicitudes) y empleado (consulta y marcaje propio).

2. Funcionalidades del sistema

El sistema cubre todo el ciclo de control de personal: desde el marcaje de entrada/salida hasta la generación de reportes y credenciales. Estos son los módulos incluidos:

Dashboard KPIs y gráficos de asistencia del personal	Empleados Alta, edición, baja y búsqueda de personal	Departamentos y cargos Estructura organizacional y remuneraciones
Horarios y turnos Jornadas con entrada, salida y tolerancia	Estacion de marcaje Registro por QR, código de barras, PIN o biométrico	Registro de asistencia Historial, tardanzas y horas trabajadas
Permisos Solicitud, aprobación y rechazo de permisos	Vacaciones Solicitud con cálculo de días y aprobación	Reportes Filtros por fecha y departamento, exportación CSV
Credenciales / fotocheck Impresión con foto, QR o código de barras	Dispositivos Biométricos, lectores y kioscos con clave API	Escaner por cámara Escaneo QR desde PC o celular por HTTPS

Marcaje multi-metodo (plan B ante fallas)

El sistema no depende de un único dispositivo. Si el biométrico o un lector falla, el personal puede seguir marcando con QR, código de barras o PIN, y RR.HH. puede registrar manualmente. Esto asegura que el control de asistencia nunca se detiene.

Metodo	Como funciona	Hardware requerido
Fotocheck QR	Fotocheck con QR imprimible, escaneado con cámara en la Estación de Marcaje	Solo una cámara o celular
Código de barras	Mismo fotocheck leído con lector USB o tecleado manualmente	Lector USB opcional
PIN	El empleado teclea su PIN en el teclado numérico de la estación	Ninguno
Biométrico	Dispositivo externo envía la marca por API con su clave x-api-key	Lector biométrico
Manual (RR.HH.)	Registro manual desde el módulo Asistencia	Ninguno

3. Instalacion paso a paso en una computadora nueva

Sigue estos pasos en orden. Se asume una computadora con Windows, macOS o Linux recién formateada, sin ningún programa instalado todavía.



Paso 1 - Instalar Node.js

Ir a <https://nodejs.org> y descargar la versión LTS (recomendada) para tu sistema operativo. Ejecutar el instalador y aceptar las opciones por defecto (incluye npm automáticamente). Para verificar la instalación, abrir una terminal (CMD, PowerShell o Terminal) y ejecutar:

```
node -v
npm -v
```

Paso 2 - Instalar MySQL

Descargar MySQL Community Server desde <https://dev.mysql.com/downloads/installer/> (en Windows usar el 'MySQL Installer'; en macOS el paquete .dmg; en Linux usar el gestor de paquetes, por ejemplo `sudo apt install mysql-server`). Durante la instalación, definir una contraseña para el usuario root (o dejarla en blanco para pruebas locales) y confirmar que el servicio quede escuchando en el puerto 3306. Opcionalmente instalar un cliente visual como MySQL Workbench o DBeaver para revisar las tablas.

Paso 3 - Copiar el proyecto

Copiar la carpeta completa `neg-sistema-control-asistencia-digital` a la nueva computadora (por USB, carpeta compartida, o clonando el repositorio Git si el proyecto está subido a GitHub/GitLab). La carpeta debe contener `backend/` y `frontend/`.

Paso 4 - Configurar la base de datos (backend/.env)

Dentro de la carpeta backend, crear (o editar) el archivo `.env` con estos valores:

```
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=      # tu contraseña de MySQL, o vacío
DB_NAME=neg_control_asistencia
DB_PORT=3306
```

Si MySQL tiene contraseña para root, colocarla en DB_PASSWORD.

Paso 5 - Instalar dependencias y crear la base de datos

Abrir una terminal en la carpeta del backend y ejecutar:

```
cd backend
npm install
npm run init-db
```

Esto crea automáticamente la base neg_control_asistencia, sus tablas y datos de ejemplo (alternativa manual: ejecutar backend/db/schema.sql en tu cliente MySQL).

Paso 6 - Iniciar el backend (API)

```
cd backend
npm start
# API disponible en http://localhost:4700
```

Paso 7 - Iniciar el frontend (interfaz)

En una segunda terminal (dejando la del backend abierta):

```
cd frontend
npm install
npm run dev
# Interfaz disponible en http://localhost:5173
```

Abrir <http://localhost:5173> en el navegador. El frontend redirige automáticamente las llamadas a /api hacia el backend, sin configuración adicional.

Paso 8 - Ingresar con un usuario de prueba

Rol	Usuario	Contraseña	Permisos
Administrador	admin	admin123	Acceso total, incluye eliminar registros
Recursos Humanos	rrhh	rrhh123	Gestión de personal, aprobar solicitudes
Empleado	andrea	emp123	Consulta y marcaje

4. Hosting web recomendado

Como el sistema usa MySQL (no PostgreSQL), conviene un hosting con base de datos MySQL administrada y despliegue simple desde un repositorio Git. Estas son las opciones evaluadas:

Opcion	Tipo	MySQL nativo	Ideal para
Railway (recomendado)	PaaS administrada	Si, 1 clic	Despliegue rapido sin servidor propio
Render	PaaS administrada	Si (con disco)	Alternativa solida, similar a Railway
Hostinger VPS	VPS / servidor propio	Se instala manual	Control total, requiere mas configuracion

Recomendacion: Railway (railway.com). Es una plataforma administrada (PaaS) que permite desplegar el backend Node.js, la base de datos MySQL y el frontend desde un mismo panel, conectado directamente a GitHub. No requiere configurar servidores, Nginx ni certificados SSL manualmente: todo el despliegue queda automatizado con cada cambio subido al repositorio. Su plan de entrada (Hobby, aproximadamente 5 USD/mes de credits de uso) es suficiente para un sistema de asistencia de una empresa pequena o mediana.

Alternativas validas: Render (muy similar a Railway, tambien soporta MySQL) y Hostinger VPS (mas economico a largo plazo, pero requiere configurar Docker/Nginx manualmente, recomendable si ya tienes experiencia con servidores Linux).

5. Paso a paso para subir el sistema a Railway



Paso 1 - Cuenta en Railway y repositorio en GitHub

Crear una cuenta gratuita en <https://railway.com> (se puede usar la cuenta de GitHub para iniciar sesion). Si el proyecto todavia no esta en GitHub, crear un repositorio nuevo y subir la carpeta completa neg-sistema-control-asistencia-digital con git:

```
git init
git add .
git commit -m "Version inicial"
git branch -M main
git remote add origin https://github.com/tu-usuario/tu-repo.git
git push -u origin main
```

Paso 2 - Crear el proyecto en Railway

En el panel de Railway, clic en New Project y elegir Deploy from GitHub repo. Autorizar el acceso a GitHub y seleccionar el repositorio del sistema.

Paso 3 - Anadir la base de datos MySQL

Dentro del proyecto, clic en + New y elegir Database → MySQL. Railway crea la instancia automaticamente y muestra sus credenciales en la pestana Connect (host, usuario, contrasena, puerto y nombre de base de datos).

Paso 4 - Desplegar el backend

Anadir un segundo servicio senalando a la carpeta backend del repositorio (Root Directory = backend). Railway detecta automaticamente que es un proyecto Node.js y ejecuta `npm install` y `npm start`.

Paso 5 - Variables de entorno del backend

En la pestaña Variables del servicio backend, agregar (usando los datos de conexión de la base de datos creada en el paso 3):

```
DB_HOST=<host de Railway>
DB_USER=<usuario de Railway>
DB_PASSWORD=<contraseña de Railway>
DB_NAME=<nombre de la base>
DB_PORT=<puerto de Railway>
JWT_SECRET=<clave larga y aleatoria>
PORT=4700
```

Luego ejecutar una sola vez el script de inicialización (desde la terminal de Railway o localmente apuntando a la base remota) equivalente a `npm run init-db` para crear las tablas.

Paso 6 - Desplegar el frontend

Añadir un tercer servicio con Root Directory = frontend. Definir como comando de build `npm run build` (genera la carpeta dist/) y servirla como sitio estático, o usar el plugin de sitios estáticos de Railway. Configurar la variable de entorno con la URL pública del backend para que el frontend sepa a donde enviar las peticiones /api.

Paso 7 - Dominio y verificación final

Railway asigna automáticamente un dominio del tipo `tu-app.up.railway.app` con HTTPS incluido. Si se desea un dominio propio, en Settings → Networking → Custom Domain se agrega el dominio y se configura el registro CNAME indicado en el proveedor de dominios. Finalmente, abrir la URL pública del frontend, iniciar sesión con un usuario de prueba y validar que el marcate, los reportes y el dashboard funcionen correctamente antes de cambiar las contraseñas por defecto.

Importante antes de pasar a producción: cambiar JWT_SECRET por una clave larga y aleatoria, cambiar las contraseñas de los usuarios de prueba (admin, rrhh, andrea) y activar copias de seguridad automáticas de la base de datos MySQL en Railway.