# Ambiente de Desarrollo

Proyecto: Plataforma de Gestión de Ingreso de Vehículos al Taller – PepsiCo Chile

Cliente: PepsiCo Chile – Flota Nacional

Preparado por: Equipo de Desarrollo TI

Fecha: Octubre 2025

## 1. Objetivo

Definir el entorno técnico y las configuraciones necesarias para el desarrollo, pruebas y despliegue del sistema, asegurando uniformidad, trazabilidad y control en todas las fases del proyecto.

## 2. Componentes del Ambiente

### 2.1. Hardware

| Tipo de Servidor | Descripción | Especificación |
| --- | --- | --- |
| Estación de Desarrollo | Equipos locales de programadores | CPU i7 o Ryzen 7 / 16 GB RAM / SSD 512 GB |
| Servidor de Base de Datos (Pruebas) | Firebase / IndexedDB | Hosting cloud, almacenamiento dinámico |
| Servidor Web de Pruebas | Hosting interno o emulado (localhost / Render / Firebase Hosting) | Node.js / Nginx |
| Servidor de Producción | Entorno Cloud escalable | Google Cloud / Firebase Hosting |

### 2.2. Software

| Categoría | Herramienta / Versión | Descripción |
| --- | --- | --- |
| Sistema Operativo | Windows 11 Pro / Ubuntu 22.04 LTS | Estaciones de desarrollo |
| IDE / Editor | Visual Studio Code v1.93+ | Desarrollo frontend y backend |
| Control de versiones | Git + GitHub | Gestión de versiones y colaboración |
| Lenguajes | HTML5, CSS3, JavaScript (ES6), Node.js 20 | Desarrollo web |
| Frameworks | Express.js, Bootstrap 5 | Backend y frontend responsivo |
| Base de Datos | Firebase / IndexedDB | Persistencia y sincronización |
| Navegadores soportados | Google Chrome, Edge | Validación de compatibilidad |
| Pruebas | Rational TeamTest, Jest, Postman | Validación funcional e integración |
| Documentación | Microsoft Office 365, PlantUML | Gestión documental y diagramas |

## 3. Configuración del Entorno

- Repositorio principal en GitHub con ramas: main, develop y feature/\*.  
- Variables de entorno en .env: FIREBASE\_API\_KEY, FIREBASE\_PROJECT\_ID, STORAGE\_BUCKET, HOST\_URL.  
- Deploy automático mediante GitHub Actions hacia entorno de pruebas.  
- Registro de logs de errores y métricas de rendimiento en consola y Firebase Analytics.

## 4. Ambientes Definidos

| Tipo | Descripción | Propósito |
| --- | --- | --- |
| Desarrollo | Local / Firebase test | Programación y validaciones individuales |
| Pruebas | Firebase Hosting + BD de pruebas | Ejecución de casos definidos por QA |
| Producción | Firebase Hosting (dominio oficial) | Despliegue final validado y estable |

## 5. Roles y Responsabilidades

| Rol | Responsable | Función |
| --- | --- | --- |
| Líder Técnico | Diego Parra | Configuración del entorno, control de versiones, soporte técnico |
| QA/Tester | Rodrigo Cubillos | Pruebas de integración y validación de ambiente |
| Gerente de Proyecto | Adán Berríos | Seguimiento de despliegues y control de calidad |
| Desarrolladores | Equipo TI | Implementación de módulos y gestión de código |

## 6. Control de Versiones y Seguridad

- Uso de repositorio privado GitHub con permisos por rol.  
- Revisión obligatoria por pull request antes de fusionar ramas.  
- Backups automáticos semanales en Firebase Storage.  
- Validación HTTPS y autenticación por token.  
- Control de acceso por roles (Chofer, Supervisor, Mecánico/Admin).

## 7. Riesgos del Ambiente

| ID | Riesgo | Probabilidad | Impacto | Mitigación |
| --- | --- | --- | --- | --- |
| A1 | Fallas de conectividad | Media | Alta | Uso de modo offline con sincronización diferida |
| A2 | Configuración inconsistente entre entornos | Media | Media | Documentar versiones y scripts de instalación |
| A3 | Pérdida de datos de prueba | Baja | Alta | Backups automáticos y control de acceso |
| A4 | Fallos de despliegue en producción | Media | Alta | Pruebas completas en entorno staging antes de publicar |

## 8. Control de Cambios

Toda modificación al entorno deberá:  
1. Ser aprobada por el Líder Técnico.  
2. Registrarse en el Control de Cambios del Proyecto.  
3. Validarse en ambiente de pruebas antes de migrar a producción.