

# Propuesta de Diseño Arquitectónico (TDD)

*Proyecto: Electra FieldConnect*

Estado: Finalizado para Implementación

## I. Introducción y Propósito

Este documento define la arquitectura de software para el MVP de Electra FieldConnect. El sistema busca resolver la ineficiencia en las visitas técnicas en campo mediante un portal de autogestión para clientes residenciales en Cali. La arquitectura se ha diseñado bajo principios de alta cohesión y bajo acoplamiento, priorizando la velocidad de respuesta y la integridad de los datos.

## II. Decisiones de Diseño Estratégico

### Backend: .NET 9 (Isolated Worker Model)

Al separar el proceso de la función del host de Azure, ganamos control total sobre el ciclo de vida de la aplicación. Esto nos permite usar Minimal APIs para reducir la latencia y middleware personalizado para un manejo de errores elegante y uniforme.

### Capa de Datos: Dapper + SQLite (Modo WAL)

Necesitamos velocidad. Dapper nos permite hablar directamente con SQL sin la sobrecarga de un ORM pesado. Al activar el modo Write-Ahead Logging (WAL) en SQLite, permitimos que varios usuarios consulten sus NICs simultáneamente mientras otro está agendando, evitando bloqueos de base de datos.

### Frontend: React 19 + Tailwind CSS

La experiencia de usuario debe ser instantánea. React 19 nos permite manejar estados de carga de forma nativa. Tailwind asegura un diseño liviano que carga rápido incluso en redes móviles de baja calidad.