

## **Definición de Proyecto: Conecta Tehuacán**

### **Definición del Problema**

En la región de Tehuacán, Puebla, existe una brecha de comunicación entre las micro, pequeñas y medianas empresas (MiPyMEs) y la fuerza laboral local (jóvenes y adultos desempleados). Los métodos actuales son informales o dependen de redes sociales saturadas que no permiten un filtrado eficiente.

### **Justificación de la App Web:**

A diferencia de un sitio web estático, Conecta Tehuacán requiere una Aplicación Web (Web App) porque necesita interactividad bidireccional:

- Gestión de Datos: Los usuarios deben subir y editar CVs y vacantes.
- Dinamicidad: Las búsquedas deben filtrarse en tiempo real por sector y ubicación.
- Seguridad: Requiere un sistema de autenticación para proteger los datos personales.

### **Usuarios Objetivo**

- Reclutadores (Empresas/Negocios): Desde microempresas familiares hasta compañías grandes en Tehuacán que buscan talento local.
- Candidatos: Jóvenes buscando su primer empleo y adultos desempleados que necesitan una plataforma accesible y centralizada.

### **Alcance del Proyecto**

Dado que el tiempo es limitado, el alcance se centra en un Producto Mínimo Viable (MVP) con un backend funcional al 70%.

Pestañas y Módulos de la Aplicación:

- Home: Buscador principal de vacantes con filtros básicos.
- Registro/Login: Acceso diferenciado para empresas y candidatos.
- Panel del Candidato: Creación de perfil y carga de información profesional.
- Panel de Empresa: Formulario para publicar, editar y dar de baja vacantes.
- Vista de Vacante: Detalle de la oferta y botón de postulación.

## Arquitectura Técnica

Para garantizar que la plataforma sea escalable, segura y capaz de manejar la interacción bidireccional requerida (postulaciones en tiempo real, filtros dinámicos, edición de perfiles/vacantes), se ha seleccionado el siguiente stack tecnológico moderno y probado en 2026:

### Frontend

- React → Biblioteca declarativa y eficiente para construir interfaces de usuario basadas en componentes reutilizables.
- Vite → Herramienta de construcción ultra-rápida para un entorno de desarrollo ágil (HMR instantáneo y builds optimizados).
- Next.js → Framework complementario para optimizar el rendimiento mediante renderizado del lado del servidor (SSR), rutas dinámicas y mejoras en SEO (útil para que las vacantes sean indexadas y encontradas orgánicamente).
- Gestión de Estados → Uso de bibliotecas maduras como Redux o Zustand (según complejidad) para manejar la complejidad de los datos en el cliente (filtros, perfiles, postulaciones en tiempo real).
- Interactividad → Actualización de la interfaz en tiempo real en respuesta a las acciones del usuario (búsquedas dinámicas, notificaciones de postulación, edición inmediata de CV/vacantes).

### Backend

- Node.js → Entorno de ejecución de JavaScript orientado a eventos, diseñado específicamente para aplicaciones de red escalables y con alto volumen de conexiones concurrentes (ideal para múltiples usuarios buscando/postulando al mismo tiempo).
- Express / NestJS → Frameworks para la creación rápida de rutas API (Express para simplicidad y velocidad en MVP) o para implementar una arquitectura modular, estructurada y tipada con TypeScript (NestJS para mayor mantenibilidad a mediano plazo).
- Lógica de Negocio → Gestión completa de autenticación (JWT + refresh tokens), seguridad (validación de datos, rate limiting), APIs RESTful y comunicación fluida con la base de datos.

## **Base de Datos y Persistencia**

- PostgreSQL (preferido) / MySQL → Sistemas relacionales robustos para el manejo de datos estructurados (usuarios, empresas, vacantes, postulaciones, historial). PostgreSQL destaca por su soporte nativo a búsquedas complejas, JSON y extensiones de geolocalización.
- Prisma / TypeORM → ORMs modernos para abstraer la base de datos, facilitar migraciones automáticas, tipado fuerte y mejorar la mantenibilidad del código (Prisma recomendado por su excelente DX y autogeneración de tipos).
- Geolocalización → Implementación de filtros de cercanía y ubicación específicos para la región de Tehuacán (usando PostGIS en PostgreSQL o consultas con coordenadas + distancia).

## **Infraestructura, Hosting y Servicios en la Nube**

- Contenedores y Despliegue → Docker para estandarizar entornos (desarrollo = producción). Despliegue del backend en servidores VPS o Kubernetes (según crecimiento). Alternativas PaaS rápidas para MVP: Render, Fly.io, Railway o Heroku (para rapidez en las primeras fases).

- Almacenamiento de Archivos → Servicios en la nube como AWS S3, Google Cloud Storage o DigitalOcean Spaces para guardar de forma segura y escalable los CVs en PDF, fotos de perfil y otros archivos adjuntos.
- Notificaciones → Integración con SendGrid o Mailgun para envío de correos electrónicos transaccionales (confirmación de registro, nueva postulación, vacante publicada). En fases posteriores: notificaciones push (Firebase Cloud Messaging o OneSignal).
- Este stack combina velocidad de desarrollo (Vite + React), rendimiento y SEO (Next.js cuando sea necesario), escalabilidad horizontal (Node.js + Docker), seguridad y mantenibilidad (TypeScript + ORM + buenas prácticas) y costos razonables para un proyecto regional que puede crecer a nivel estatal o nacional en el futuro.

## **Equipo, Roles y Responsabilidades**

Para garantizar la calidad de software y una ejecución alineada a los estándares de la industria, se ha definido una estructura de roles basada en el control de calidad y la especialización técnica:

### Estructura de Trabajo

- 3523110229 — TL (Tech Lead / Arquitectura) Responsable de la visión técnica global y el diseño de la arquitectura del sistema. Supervisa la interoperabilidad entre el frontend y el backend, asegurando que el flujo de datos sea eficiente y seguro.
- 3523110279 — FE (Frontend Developer) Encargado de la construcción de la interfaz de usuario utilizando React y Vite. Su enfoque es la modularidad mediante componentes reutilizables y la optimización del lado del cliente para una experiencia de usuario fluida.
- 3523110374 — BE (Backend Developer) Implementa la lógica de negocio y las APIs RESTful en Node.js/Express. Gestiona la persistencia en la base de datos PostgreSQL, la autenticación y la lógica de filtrado dinámico.

- 3523110007 — DO (DevOps / CI-CD) Responsable de la portabilidad del sistema mediante la contenedorización con Docker. Configura y mantiene el pipeline de integración continua (CI/CD) para automatizar el despliegue y las pruebas.
- 3523110321 — QA (Quality Assurance / Testing) Garante de la estabilidad del sistema. Lidera el proceso de Code Review y validación de Pull Requests, asegurando que cada integración cumpla con los estándares de seguridad y rendimiento antes de llegar a la rama principal.

### **Justificación**

El proyecto es viable para el 28 de enero bajo un enfoque de MVP. Al centrar el esfuerzo en el núcleo de la base de datos (vacantes y usuarios) y una interfaz limpia en React, se garantiza un prototipo funcional que resuelve el problema real de desempleo en Tehuacán.