



Prueba Técnica

Posición: Desarrollador Full Stack

Reto Técnico: Desarrollador Full-Stack

Proyecto: DocuTrack

Bienvenido/a a nuestro proceso de selección. Queremos que veas esta prueba no como un examen, sino como tu primer proyecto con nosotros. Es una oportunidad para que nos muestres tu talento, tu forma de pensar y cómo das vida a una idea a través del código.

Nuestro objetivo es entender cómo aboras un problema de principio a fin. No buscamos la perfección, sino la funcionalidad, la organización y la pasión por construir.

Plazo: Tienes 7 días para completarlo a partir de la recepción de este correo.

1. El Proyecto: ¿Qué vamos a construir?

Construirás un prototipo funcional de **"DocuTrack"**, una aplicación web diseñada para simplificar la burocracia y tramitología. La idea es que cualquier persona pueda solicitar un certificado oficial (como el de nacimiento o de estudios), seguir su progreso y descargarlo, todo desde un mismo lugar.

2. Tu Caja de Herramientas (Stack Tecnológico)

Siéntete libre de elegir las tecnologías con las que te sientas más productivo/a.

- **Frontend (Elige uno):**
 - Next.js
 - React.js
- **Backend (Elige uno):**
 - FastAPI (Python)
 - Laravel (PHP)
 - SpringBoot (Java)
 - Go (con Gin)
 - Express.js / Nest.js (Node.js)
- **Base de Datos (Requerido):**
 - PostgreSQL
- **Despliegue Opcional: ¡es un gran plus!**
 - Si te animas, puedes desplegar tu proyecto en Vercel, Render, AWS, Google Cloud, etc.

3. Funcionalidades Esenciales (El Corazón del Proyecto)

Para facilitar el desarrollo, piensa en el proyecto como pequeñas "historias de usuario":

Para el Usuario Ciudadano: **USER**

1. Registro y Acceso:

- **Como usuario**, quiero poder crear una cuenta y hacer login para mantener mis trámites seguros. (Un sistema simple de email y contraseña es perfecto).

2. Solicitud del Certificado:

- **Como usuario**, quiero un formulario claro para solicitar mi certificado, donde pueda rellenar mis datos y adjuntar los documentos necesarios (PDF o JPG).

3. Seguimiento en Tiempo Real:

- **Como usuario**, quiero una pantalla donde pueda ver el estado actual de mi solicitud (Ej: "Recibido" -> "En Validación" -> "Emitido") para saber en qué etapa se encuentra.

4. Descarga Segura:

- **Como usuario**, quiero que, una vez mi trámite esté "Emitido", aparezca un botón para descargar mi certificado en formato PDF.

Para el Administrador (Backoffice): **ADMIN**

1. Panel de Control:

- **Como administrador**, quiero ver una lista de todas las solicitudes para tener una visión general del trabajo pendiente.

2. Gestión de Trámites:

- **Como administrador**, quiero poder hacer clic en una solicitud para revisar los datos del solicitante, descargar sus documentos y tomar una decisión.

3. Validación y Acciones:

- **Como administrador**, quiero tener botones para "Aprobar", "Rechazar" o "Pedir Corrección" en un trámite, y que esto actualice el estado que ve el usuario.

4. Requisitos Técnicos Clave (Las Reglas del Juego)

- **Seguridad con JWT:** Protege las rutas de tu API. Nadie sin autenticación debería poder acceder a la información.
- **Sistema de Roles:** Define al menos dos roles: **USER** y **ADMIN**. Asegúrate de que un **USER** no pueda acceder a las funciones del **ADMIN**.
- **Modelo de Datos:** Piensa bien cómo estructurarás tus tablas en PostgreSQL. Una buena base de datos es la clave de un buen proyecto.
- **Generación de PDF:** El certificado debe generarse dinámicamente con los datos del solicitante. No te compliques con el diseño; un documento simple con el texto correcto es suficiente.

5. ¿Cómo Organizar tu Trabajo? (Nuestros Consejos)

- **Planifica antes de codificar:** Dedica una hora o más, a pensar en el modelo de datos y las rutas de la API. Un pequeño diagrama puede ahorrarte horas de trabajo.
- **Usa `git` como un profesional:** Haz `commits` pequeños y frecuentes. Un historial de `git` claro nos cuenta la historia de cómo construiste el proyecto paso a paso.
- **Enfócate en el MVP (Producto Mínimo Viable):** Es mejor entregar algo simple y funcional que algo complejo e incompleto. La interfaz no necesita ser espectacular; la funcionalidad es la prioridad.

6. ¿Qué Debes Entregar?

- **El enlace a tu repositorio de código (GitHub, GitLab, etc.)**
 - **Esto debe incluir: La Planificación del proyecto.** Asegúrate de que el archivo `README.md` de tu repositorio sea tu carta de presentación. Debe incluir:
 1. Las tecnologías que elegiste y por qué.
 2. Instrucciones claras sobre cómo un desarrollador puede clonar tu proyecto, instalar las dependencias y ejecutarlo en su máquina.
 3. (Opcional) Cualquier comentario sobre decisiones de diseño o funcionalidades que dejaste fuera a propósito.

Entrega tu repositorio a través del siguiente enlace:
<https://forms.gle/9JqX6dhefe5UcQFN8>

¡Mucha suerte! Estamos muy interesados en ver tu solución.