



Test	Developer Backend
Área	Desarrollo
Fecha	Julio 2024 Diseñada Karla Versión 1.0

Test para Desarrollador Full Stack (React + Node.js)

Objetivo: Desarrollar una aplicación web para realizar evaluaciones 360 grados de empleados remotos en una empresa de desarrollo de aplicaciones.

Tecnologías requeridas:

- Frontend: React
- Backend: Node.js con Express.js
- Base de datos: MongoDB (puedes usar Mongoose como ODM)

Requerimientos:

Backend (Node.js + Express):

1. Implementar una API RESTful con los siguientes endpoints:
 - POST /api/auth/register: Registro de usuarios
 - POST /api/auth/login: Inicio de sesión (devuelve JWT)
 - GET /api/employees: Obtener lista de empleados
 - POST /api/evaluations: Crear una nueva evaluación
 - GET /api/evaluations/:id: Obtener detalles de una evaluación
 - PUT /api/evaluations/:id: Actualizar una evaluación
 - GET /api/evaluations/employee/:id: Obtener evaluaciones de un empleado
 - POST /api/feedback: Enviar feedback para una evaluación
 - GET /api/reports/employee/:id: Generar reporte de evaluación para un empleado
2. Implementar autenticación JWT y manejo de roles (Admin, Manager, Employee)
3. Crear modelos de datos para Usuario, Empleado, Evaluación y Feedback
4. Implementar lógica para calcular resultados de evaluaciones
5. Utilizar middleware para manejar errores y validar datos de entrada

Frontend (React):

1. Crear una interfaz de usuario responsiva con los siguientes componentes:
 - Página de registro e inicio de sesión
 - Dashboard para diferentes roles (Admin, Manager, Employee)
 - Formulario para crear y editar evaluaciones
 - Interfaz para completar evaluaciones (autoevaluación y evaluación de colegas)
 - Visualización de resultados de evaluaciones (gráficos y tablas)
 - Página de perfil de empleado con historial de evaluaciones
2. Implementar gestión de estado utilizando Redux o Context API
3. Utilizar React Router para la navegación



Test	Developer Backend
Área	Desarrollo
Fecha	Julio 2024 Diseñada Karla Versión 1.0

4. Crear formularios dinámicos para las evaluaciones (diferentes categorías y preguntas)
5. Implementar visualizaciones de datos (gráficos de radar, barras, etc.) para los resultados

Funcionalidades adicionales: (Opcional)

1. Implementar un sistema de notificaciones (email o in-app) para recordatorios de evaluaciones pendientes
2. Añadir la opción de exportar resultados en formato PDF o CSV
3. Implementar un chat o sistema de comentarios para discusiones sobre evaluaciones

Diseño propuesto:

1. Utilizar un diseño profesional y moderno, adecuado para una herramienta de RRHH
2. Implementar una paleta de colores que transmita confianza y profesionalismo (sugerir al menos 3 colores principales)
3. Usar iconos intuitivos para las diferentes acciones y secciones
4. Asegurar que la interfaz sea responsiva y funcione bien en dispositivos móviles
5. Implementar un diseño accesible (considerar contraste, tamaños de fuente, etc.)

Entrega:

- Repositorio Git con el código fuente completo (frontend y backend en carpetas separadas)
- Archivo README.md con:
 - Instrucciones para configurar y ejecutar el proyecto
 - Explicación de la estructura del proyecto y decisiones de diseño
 - Cómo ejecutar los tests (si se implementaron)
- Archivo .env de ejemplo con las variables de entorno necesarias
- Diagrama de la estructura de la base de datos
- Deploy de la aplicación en alguna plataforma como Heroku, Netlify o Vercel (Opcional)

Criterios de evaluación:

1. Funcionalidad (30%):
 - Implementación correcta de todas las características requeridas
 - Lógica adecuada para el cálculo y presentación de resultados de evaluaciones
 - Manejo de roles y permisos
2. Calidad del código (20%):
 - Estructura y organización del código
 - Uso de patrones de diseño apropiados
 - Legibilidad y mantenibilidad



Test	Developer Backend
Área	Desarrollo
Fecha	Julio 2024 Diseñada Karla Versión 1.0

3. Frontend (20%):

- Uso efectivo de componentes de React y hooks
- Implementación correcta de la gestión del estado
- Diseño responsivo y experiencia de usuario
- Visualización efectiva de datos de evaluaciones

4. Backend (20%):

- Diseño apropiado de la API
- Implementación correcta de autenticación y seguridad
- Eficiencia en las consultas a la base de datos
- Manejo adecuado de la lógica de negocio

5. Diseño y UX (10%):

- Atractivo visual y coherencia del diseño
- Facilidad de uso e intuitividad de la interfaz
- Consideraciones de accesibilidad

6. Características adicionales y creatividad (10%):

- Implementación de funcionalidades extras relevantes para el sistema de evaluación 360
- Soluciones creativas a los desafíos específicos de las evaluaciones remotas

7. Documentación y uso de Git (5%):

- Claridad en la documentación proporcionada
- Uso adecuado de Git (commits, branches, etc.)