

Assessment Técnico

Desarrollo Full Stack

Contexto General

El objetivo de esta prueba es evaluar las competencias de desarrollo full stack mediante la construcción de una aplicación de gestión de tareas (To-Do) con un nivel de complejidad mayor al clásico, incorporando unidades organizacionales, roles, permisos, y un flujo de aprobación.

La aplicación simula un entorno corporativo donde múltiples unidades trabajan sobre un conjunto común de tareas, pero con reglas claras de visibilidad, edición y aprobación.

Unidades Organizacionales

- Recursos Humanos (RRHH)
- Finanzas
- Business Intelligence (BI)

Roles y Jerarquías

Cada unidad debe contar con al menos dos niveles de usuario:

- Usuario estándar
- Usuario supervisor

Permisos generales

Acción	Usuario estándar	Usuario supervisor
Ver todas las tareas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Crear tareas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Editar tareas de su unidad	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Eliminar tareas de su unidad	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Completar tareas de su unidad	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Aprobar cambios	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Modelo de Tareas (To-Do)

Cada tarea debe contener como mínimo los siguientes campos:

- ID
- Nombre de la tarea
- Descripción
- Estado (Ejemplo: pendiente, en progreso, completada)
- Prioridad (Ejemplo: baja, media, alta)
- Fecha de creación

Flujo de Aprobación

Tabla principal de tareas

- Contiene únicamente tareas y cambios aprobados.
- Todas las tareas en esta tabla son visibles para todos los usuarios.

Tabla temporal / pendiente de aprobación

- Cuando un usuario estándar crea, edita, completa o solicita la eliminación de una tarea:
 - El cambio no impacta directamente en la tabla principal.
 - Se guarda en una tabla temporal de solicitudes.
- Solo los encargados de la misma unidad pueden:
 - Ver las solicitudes pendientes.
 - Aprobar o rechazar los cambios.
- Una vez aprobado:
 - El cambio se refleja en la tabla principal.

Diseño y Experiencia de Usuario

- La aplicación debe contar con una interfaz clara, consistente y usable, acorde a un entorno corporativo.
- El diseño visual de la aplicación es libre, y se evaluará el criterio estético, la coherencia entre pantallas y la experiencia de usuario.

Login

- Debe ajustarse al diseño proporcionado, respetando la imagen y la paleta de colores entregadas.
- Se evaluará la fidelidad visual al diseño entregado.

Registro

- Diseño libre
- Debe permitir la selección de unidad y rol.

Gestión de tareas

- Diseño libre
- Debe permitir la visualización clara de las tareas, su estado, prioridad y acciones disponibles según el rol del usuario.
- Se valorará una experiencia de uso clara y ordenada.

Requisitos Técnicos Obligatorios

- Desarrollo del backend utilizando Node.js con Express (se sugiere el uso de TypeScript).
- Desarrollo del frontend utilizando React.js.
- Implementación de servicios RESTful para la comunicación entre frontend y backend.
- Implementación de un mecanismo de autenticación segura (JWT, sesiones u otro equivalente).
- Gestión de roles y control de permisos a nivel de backend.
- Implementación de validaciones tanto en el frontend como en el backend.
- Persistencia de datos utilizando una base de datos relacional o no relacional
- Simulación y/o documentación de pruebas de los endpoints mediante Postman o Swagger.

Entregables

- Repositorio con el código fuente.
- Instrucciones claras para levantar el proyecto.
- Breve documento explicando:
 - Modelo de datos y justificación
 - Principales decisiones técnicas tomadas durante el desarrollo.
 - Breve descripción del flujo de permisos y aprobaciones.

Se valorará la inclusión de funcionalidades, estructuras o mejoras adicionales que el/la candidato/a considere relevantes, siempre que estén justificadas y alineadas con el dominio del problema.