Prueba Técnica para Desarrollador Junior - React JS

A continuación, encontrarás una serie de retos. Debes elegir solo **una (1)** de las siguientes opciones y desarrollar la aplicación solicitada. Esta prueba está diseñada para evaluar tus conocimientos en React JS, tu capacidad para estructurar un proyecto de forma organizada y tu habilidad para abordar un problema real de una empresa.

Lista de Chequeo para Evaluación de la Prueba

La calificación de esta prueba se divide en dos partes con el mismo peso:

1. Revisión Técnica del Proyecto (50 puntos)

Criterio	**Puntaje Máximo**	**Puntaje Obtenido**
Uso correcto de useState	5 pts	
Uso de props entre componentes	5 pts	
Implementación de rutas con React Router DOM (incluyendo rutas protegidas)	10 pts	
Manejo de eventos (onClick, onChange, etc.)	5 pts	
Manipulación de datos con métodos de arreglo (map, filter, etc.)	5 pts	
Estructura de carpetas organizada y lógica	5 pts	
Persistencia con localStorage (datos y token)	5 pts	
Uso de datos quemados o conexión con JSON-server	5 pts	
Aplicación de semántica HTML adecuada	2.5 pts	
Inclusión y calidad del README (documentación, instrucciones, estructura)	2.5 pts	

Subtotal Revisión Técnica: ____ / 50 pts

---

2. Revisión o Entrevista Presencial (50 puntos)

Criterio	Puntaje Máximo	Puntaje Obtenido
Capacidad para explicar decisiones técnicas	15 pts	
Argumentación sobre estructura del proyecto	10 pts	
Claridad, comunicación y lógica al abordar problemas	10 pts	
Enfoque de diseño y experiencia de usuario (UX/UI)	15 pts	

Subtotal Entrevista: ____ / 50 pts

Puntaje Total Final

TOTAL: ____ / 100 pts

Observaciones del evaluador:

OPCIÓN 1: Sistema Interno de Registro de Visitas

Contexto: Una empresa de mediano tamaño recibe visitas de proveedores, clientes y aliados en sus oficinas. Actualmente, el registro de estas visitas se hace en papel, lo cual genera problemas de organización y acceso a la información. Se necesita una solución digital sencilla para mejorar el control de ingreso.

Requerimientos funcionales:

- Formulario para registrar una visita (nombre del visitante, motivo, hora de entrada).
- Posibilidad de marcar la salida.
- Listado de visitas registradas, filtrado por fecha.
- Vista pública informativa y panel privado para el administrador.

OPCIÓN 2: Panel de Productos para Empresa de Alimentos Artesanales

Contexto: Una pequeña empresa familiar elabora productos como mermeladas, salsas y conservas. Necesitan una herramienta sencilla para gestionar su inventario de productos, ya

que actualmente llevan el control en cuadernos o notas de celular, lo que genera confusiones frecuentes.

Requerimientos funcionales:

- Ver productos por categoría.
- Agregar nuevos productos (nombre, descripción, precio, disponibilidad).
- Marcar productos como agotados.
- Vista pública para clientes y panel privado para el administrador.

OPCIÓN 3: Registro de Citas para Peluquería

Contexto: Una peluquería urbana con varios barberos desea digitalizar la asignación de citas. Actualmente se manejan a través de mensajes por WhatsApp y una agenda física, lo que ha generado problemas de duplicidad o citas mal registradas.

Requerimientos funcionales:

- Formulario para registrar una cita (cliente, fecha, hora, barbero, servicio).
- Listado de citas del día.
- Filtrado por fecha o por barbero.
- Vista pública informativa y panel privado para el administrador.

OPCIÓN 4: Gestor de Tareas Internas

Contexto: Una microempresa de diseño gráfico busca organizar las tareas del equipo mediante una herramienta interna que permita visualizar el estado de cada actividad. Actualmente, la comunicación se da por mensajes y hojas de cálculo poco organizadas.

Requerimientos funcionales:

- Registro de tareas por usuario.
- Marcar tareas como pendiente, en progreso o completada.
- Filtrado por estado.
- Vista separada para cada usuario con acceso mediante autenticación.

OPCIÓN 5: Módulo de Pedidos - JolyGuacamoly

Contexto: JolyGuacamoly es una empresa emergente que produce guacamole natural y desea recibir pedidos a través de su página web. El dueño quiere evitar depender de aplicaciones externas o mensajes manuales. Necesita una herramienta donde los clientes puedan realizar pedidos y él los pueda gestionar.

Requerimientos funcionales:

- Vista pública de productos.
- Formulario de pedido (cliente, producto, cantidad, comentario).
- Panel privado donde el administrador ve los pedidos y actualiza su estado.

OPCIÓN 6: Registro de Asistencia a Capacitaciones

Contexto: El área de talento humano de una empresa realiza capacitaciones mensuales, pero no tiene una forma ordenada de saber quiénes asistieron ni qué temas se vieron. Se requiere una aplicación para registrar estas asistencias y consultar históricos.

Requerimientos funcionales:

- Formulario para registrar la asistencia (nombre, documento, sesión, tema).
- Vista con el listado de asistentes por sesión o tema.
- Vista administrativa privada y una vista pública informativa.

Consideraciones Generales (Obligatorias en todas las opciones):

- El proyecto debe estar realizado con **React JS**.
- Se debe usar **React Router DOM** para las rutas (incluyendo rutas protegidas).
- Uso de **useState** para manejar valores ingresados por teclado.
- Aplicación de **props** correctamente entre componentes.
- Debe existir una estructura de carpetas organizada.
- Uso de **datos quemados** (arreglos en el proyecto) o mediante **JSON-server**.
- Se debe usar **localStorage** para simular un inicio de sesión con token.
- Persistencia de datos a través de **localStorage** o una fuente externa.

- Debe utilizar **semántica HTML adecuada**.
- Se espera uso correcto de **objetos y arreglos**, incluyendo métodos como filter, map, find, etc.
- Deben aplicarse **eventos** relevantes (onClick, onChange, onSubmit, etc.).
- Debe incluirse un **README** documentando:
    - Qué hace la aplicación.
    - Cómo se ejecuta.
    - Tecnologías usadas.
    - Estructura del proyecto.
    - Capturas de pantalla (opcional pero valorado).
- El diseño debe tener una intención clara de experiencia de usuario (UX) y una presentación visual coherente (UI). No se calificarán soluciones "funcionales pero descuidadas" en su diseño.

Límites de implementación

No está permitido:

- Usar librerías adicionales de componentes como Material UI, Bootstrap, etc.
- Usar useContext, Redux o hooks personalizados.
- Hacer autenticación real, solo simulada con localStorage o json-server
- Usar bases de datos externas o backends reales (solo json-server)

Entrega:

- Entregar el proyecto en un repositorio de GitHub.
- Incluir el README solicitado.
- Compartir el enlace del repositorio.