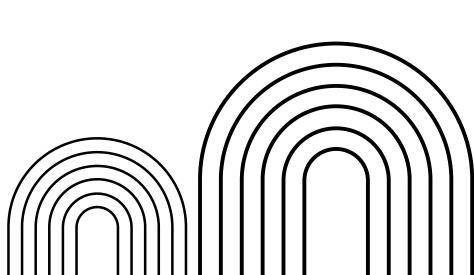


JOURNEY

Herramientas y Librerías para un Desarrollo Más Eficiente



OBJETIVOS DEL NIVEL 2

- Aprender a integrar librerías de terceros en una aplicación React para manejar estado, datos, y Ul.
- Entender buenas prácticas de desarrollo en React para mejorar la escalabilidad y el mantenimiento de la aplicación.
- Aplicar el diseño de interfaces accesibles con herramientas modernas como Chakra UI y Material UI.
- Optimizar el rendimiento utilizando técnicas avanzadas y librerías de optimización.
- Incorporar la validación y pruebas para garantizar la calidad y fiabilidad de la aplicación.

REACT JOURNEY - NIVEL 2

Manejo de estado global

- **Tema principal:** manejo del estado global (más allá de *useState* y el *useReducer*)
- Librerias:
 - **Zustand:** maneja el estado global de una manera sencilla y escalable. Compararlo con otras soluciones como Redux para ver por qué es más liviano y directo.
 - <u>Recoil:</u> Otra opción para gestionar el estado global, ideal para aplicaciones grandes.
 - <u>Context API</u>: Refrescar el concepto del Context API de React para que el equipo entienda cuándo es conveniente usarlo o no.

Fetching de datos

- **Tema principal:** Cómo manejar las peticiones HTTP y el estado relacionado con la carga de datos de forma eficiente.
- Librerias:
 - <u>React Query:</u> Explicar cómo esta librería hace que el manejo de estados de carga, error y éxito de las peticiones sea mucho más sencillo, además de sus beneficios en cuanto a caching y sincronización de datos.
 - Axios: Para hacer las solicitudes HTTP de manera más flexible.
 - **SWR:** Alternativa a React Query con enfoque en la revalidación automática de datos.

Diseño de interfaces de usuario UI

• **Tema principal:** Crear interfaces de usuario rápidas, accesibles y reutilizables.

• Librerias:

- Chakra UI: Enseñar cómo usar componentes listos para usar y personalizables. Explicar la importancia de la accesibilidad y la consistencia visual.
- Material UI: Una opción más robusta que Chakra, ideal para aplicaciones más complejas.
- **Tailwind CSS:** Enfoque en el uso de clases utilitarias para crear diseños rápidos y personalizados sin escribir tanto CSS.

REACT JOURNEY - NIVEL 2

Manejo de Formularios

- **Tema principal:** Manejo eficiente de formularios complejos y validación.
- Librerias:
 - <u>Formik:</u> Mostrar cómo simplifica el manejo de formularios y validaciones.
 - React Hook Form: Alternativa ligera y basada en hooks para el manejo de formularios.
 - **Yup:** Usado junto con Formik o React Hook Form para validaciones más complejas.

Ruteo y Navegación

- **Tema principal:** Implementación de rutas en la aplicación.
- Librerias:
 - **React Router:** Enseñar cómo crear rutas en la aplicación, con redirección, rutas anidadas, y parámetros dinámicos.
 - <u>Reach Router:</u> Alternativa a React Router que está más centrada en la accesibilidad y la simplicidad.

Animaciones

- **Tema principal:** Crear animaciones fluidas y eficientes en React.
- Librerias:
 - <u>Framer Motion:</u> Explorar cómo esta librería facilita la creación de animaciones y transiciones suaves.
 - <u>React Spring:</u> Otra opción para animaciones complejas con físicas, ideal para interacciones más dinámicas.

Testing en React

- **Tema principal:** Testeo de componentes y lógica en React.
- Librerias:
 - Jest: Introducción a Jest para escribir pruebas unitarias.
 - React Testing Library: Cómo hacer pruebas de integración y de interacción con los componentes de React.
 - **Cypress:** Para pruebas de integración más complejas y end-to-end.

REACT JOURNEY - NIVEL 2

Optiimización y Performance

- Tema principal: Optimización de la performance de aplicaciones React.
- Librerias:
 - **<u>React Lazy y Suspense:</u>** Cargar componentes bajo demanda para mejorar el rendimiento.
 - <u>React.memo y useMemo:</u> Técnicas para evitar renderizados innecesarios.

Seguridad y Autenticación

- Tema principal: Cómo manejar la autenticación y autorización en aplicaciones React
- Librerias:
 - **JWT (JSON Web Tokens):** Implementación de autenticación basada en tokens.
 - <u>Auth0</u>: Usar servicios externos para simplificar la gestión de autenticación.
 - <u>Firebase Authentication:</u> Integración de Firebase para manejar usuarios y autenticación.

Proyecto: "Dashboard de Tareas Colaborativas"

Equipo este proyecto involucra varios conceptos importantes de la temática anterior.

Nota: Crear un repositorio público desde sus perfiles y desplegarlo en sitios de hosting gratuitos como <u>vercel</u> y <u>github pages</u>.

Descripción del Proyecto:

Crear un dashboard de tareas colaborativas donde los usuarios puedan:

- Ver una lista de tareas en tiempo real.
- Crear, editar y eliminar tareas.
- Filtrar tareas por categoría o prioridad.
- Marcar tareas como completadas.
- Gestionar el estado de las tareas con almacenamiento local (localStorage o sessionStorage)

Detalles del Proyecto:

Pantalla principal (Dashboard)

• Lista de Tareas:

- Mostrar todas las tareas de forma clara y ordenada.
- o Cada tarea debe tener la opción de editar, eliminar, y marcar como completada.
- Los usuarios pueden filtrar las tareas por categoría (ej. "Trabajo", "Personal") o prioridad (Alta, Media, Baja).

Barra lateral:

- Opción de ver tareas por diferentes categorías.
- Filtros para mostrar solo tareas pendientes o completadas.

• Crear/Editar Tarea:

- Un formulario donde los usuarios puedan agregar nuevas tareas y editarlas.
- Los campos incluyen título, descripción, fecha de vencimiento, categoría, y prioridad.

Funcionalidades Adicionales:

- 1. **Autenticación**: Los usuarios pueden iniciar sesión usando un servicio de autenticación como Firebase o JWT.
- 2. **Notificaciones**: Cuando una tarea es completada o editada, el sistema debe notificar a los usuarios .
- 3. **Estadísticas**: Una página adicional que muestre el porcentaje de tareas completadas, pendientes, etc., de cada usuario.