**BOOTCAMP DE FULL STACK ENGINEERING**

**DESAFÍO SEMANAL F2-N14**

1. **OBJETIVOS**

* Trabajar con componentes funcionales.
* Gestionar props con PropTypes.
* Implementar y utilizar Context API para gestión de estado global.
* Crear y utilizar hooks personalizados para lógica de negocio.
* Diseñar interfaces con estilos en SASS siguiendo la metodología BEM.

1. **CONSIGNA**

Desarrolla un **gestor de tareas** completo y funcional utilizando React. El proyecto debe tener una base sólida, implementando la **Context API** para el manejo del **estado global**, así como **hooks nativos** y **personalizados** para encapsular la lógica de negocio. Se requiere el uso de **useState** y **useEffect** para el control del estado y efectos secundarios, y de **useNavigate** y **useLocation** para la navegación programática y el control de rutas. También deberás usar componentes de **React Router** como **BrowserRouter**, **Routes** y **Route** para definir y gestionar las rutas de la aplicación. Además, es obligatorio mantener la **persistencia de datos** en el **localStorage** para conservar las tareas al cerrar el navegador o hacer un refresh de la página web.

Tip: Puedes revisar la estructura de carpetas y componentes de la clase Nº41, así como parte de la implementación provista en la clase Nº42. La arquitectura base ya está bien definida; tu enfoque debe centrarse en completar la funcionalidad respetando esa estructura, manteniendo un código limpio, modular y escalable.

1. **INSTRUCCIONES**
2. Implementar un hook useTasks:
   * **tasks**: Array con todas las tareas
   * **add**(): Función para crear una nueva tarea
   * **remove**(id): Función para eliminar una tarea por su ID
   * **toggleStatus**(id): Función para cambiar el estado completado/pendiente
   * **getCompletedCount**(): Función que retorna el número de tareas completadas
   * **getTotalCount**(): Función que retorna el total de tareas
3. Crear los siguientes componentes funcionales:
   * **Home**: página principal con estadísticas y navegación.
   * **Tasks**: página de listado de tareas con funcionalidad de agregar.
   * **TaskList**: componente que renderiza la lista de tareas.
   * **TaskListItem**: componente que renderiza cada tarea de la lista.
   * **PageRoutes**: configuración de rutas de la aplicación.
   * **AppContext**: configuración del contexto global.
   * **AppProvider**: configuración del proveedor global.
4. Funcionalidades de botones:
   * **Ver Todas las Tareas**: En Home, redirige al usuario a la página de tareas.
   * **Agregar Tarea:** En Tasks, crea una nueva tarea.
   * **Eliminar**: En TaskListItem, eliminar la tarea.
   * **Marcar Completada / Marcar Pendiente**: En TaskListItem, cambiar el estado de la tarea.
5. Emplear SASS y nomenclatura BEM
6. Utilizar los colores de las variables de Sass.
7. Replicar los estilos que se observan en la [demo](https://drive.google.com/file/d/1FCVdM83yvlpynn1W_r_GKuQcGY6fYNHv/view?usp=drive_link).
8. **ESTRUCTURA DE DATOS**

| {  id: number,  title: string,  description: string,  completed: boolean,  } |
| --- |