

Trabajo Práctico grupal de presentación obligatoria.

El objetivo de la parte 4 del trabajo práctico n° 3 es transformar la aplicación en una Single Page Application (SPA) real utilizando **React Router** para gestionar la navegación entre las secciones pendientes. También incorporar el uso de biblioteca de componentes para modernizar la interfaz gráfica de usuario (UI).

1. Implementación de Rutas con React Router

Trabajar sobre la navegación del componente `<Nav />` que hasta el momento permanecía sin implementación. Para ello, deberán instalar e integrar **react-router-dom** para coordinar las vistas.

- **Configuración del Enrutador:** Configurar el componente principal para soportar el enrutamiento dinámico.
- **Declaración de Rutas:** El área principal (`<main>`) de la aplicación debe renderizar los componentes de forma condicional basándose en la URL del navegador:
 - `/` o `/dashboard`: Renderizará el componente `<Dashboard />`.
 - `/proyectos`: Renderizará el componente `<ListaProyectos />`.
 - `/proyectos/:id`: Renderizará el detalle de un proyecto específico (`<DetalleProyecto />`). El botón "Ver Detalle" de las tarjetas ahora debe cambiar para utilizar los enlaces de React Router.
 - `/perfil`: Renderizará el componente `<PerfilUsuario />`.
- **Refactorización del Menú (`<Nav />`):** Reemplazar todas las etiquetas tradicionales `<a>` por los componentes `<Link>` o `<NavLink>` de React Router para permitir que la navegación ocurra de manera instantánea y sin recargar el navegador.

2. Organización de Archivos y Estructura del Proyecto

Organizar en carpetas el contenido de `src/` de la siguiente forma:

- **Carpeta `src/views/`:** Aquí se colocarán los componentes que representan una "pantalla" o página completa de la aplicación asociada a una ruta de navegación.
- **Carpeta `src/components/`:** Aquí se guardarán los componentes secundarios, bloques de construcción o piezas de la interfaz que se inyectan dentro de las vistas.

Alta de Nuevas Vistas Requeridas:

Para que el sistema de rutas sea funcional, deberán crear dos nuevas pantallas en `src/views/` utilizando los componentes contenedores de Material UI (Box, Container, Typography, Paper, etc.):

1. **<Dashboard /> (src/views/Dashboard.jsx):** Será la pantalla de bienvenida (`/dashboard`). Debe incluir un título, descripción general y, al menos, un par de tarjetas de Material UI (Card) que muestren métricas simuladas (por ejemplo: *"Total de proyectos: 12"*, *"Proyectos en curso: 5"*).
2. **<PerfilUsuario /> (src/views/PerfilUsuario.jsx):** Pantalla del perfil del usuario (`/perfil`). Debe simular los datos del alumno o docente (Nombre, Rol, Institución) organizados elegantemente dentro de un componente Paper o una lista de Material UI.

3. Adopción de un Framework de Componentes (Material UI o React Bootstrap)

En esta sección deberán incorporar componentes de una biblioteca externa.

Pueden elegir **Material UI (MUI) o React Bootstrap**. Deberán refactorizar los siguientes elementos utilizando componentes de la biblioteca elegida:

- **Estructura Base:** El sistema de rejilla (Grid/Containers/Rows) para la distribución general de la página.
- **Formularios y Controles:** Los inputs, selectores y botones del componente `<FormularioProyecto />` deben heredar la estética del framework seleccionado.
- **Componentes de Presentación:** Las tarjetas de proyectos (`<ProyectoCard />`) deben implementarse utilizando el componente Card de la biblioteca.
 - Los mensajes o estados (como la alerta del `<RegistroActividad />`) deben transformarse en componentes de tipo Alert o Snackbar.

Apartado de Complejidad: Parámetros de Ruta Dinámicos (useParams)

En la fase anterior, el componente `<ListaProyectos />` decidía internamente si mostrar el detalle guardando el proyecto seleccionado en un estado local. **Esto debe cambiar.**

- El componente `<DetalleProyecto />` ahora debe ser una "página independiente" vinculada a la ruta `/proyectos/:id`. Al cargar esta vista, el componente deberá usar el hook `useParams` de React Router para capturar el id directamente desde la barra de direcciones del navegador, solicitar al `proyectoService` el objeto correspondiente y renderizar sus datos en pantalla.

Esto permitirá que si un usuario refresca la página (F5) estando en el detalle de un proyecto, la aplicación no se rompa ni vuelva al inicio, sino que retenga la visualización del proyecto seleccionado gracias a la URL.