



PROYECTO INTERMODULAR

SPRINT 8

Contexto del sprint

En este sprint se implementa en **React** la navegación definida en el prototipo de Figma, construyendo una aplicación navegable que respete los **flujos, pantallas y estados** (carga, error, vacío, éxito) previamente diseñados.

El objetivo es transformar el prototipo navegable en una experiencia funcional en código: **rutas, layouts, navegación entre módulos, y pantallas especiales** (no autorizado, 404, etc.), manteniendo coherencia con la arquitectura UX diseñada.

(Estructura inspirada en el sprint de diseño previo.)

El desarrollo incluirá, como mínimo:

- Definición de **arquitectura de rutas** (mapa de rutas).
- Implementación de navegación con **React Router** (o equivalente exigido en el módulo).
- **Layouts** (público/privado) y navegación común (header/sidebar/breadcrumbs si aplica).
- **Rutas protegidas** (autenticación/roles si existen en el flujo).
- Implementación de **todas las pantallas navegables** definidas en Figma (aunque el contenido sea “mock” si aún no hay lógica).
- Estados por pantalla: **cargando, error, vacío, éxito**.
- Página **404** y pantalla **No autorizado**.
- Documentación clara: rutas, decisiones, correspondencia Figma → ruta.

Objetivos del sprint

Al finalizar este sprint, el alumnado deberá ser capaz de:

1. Traducir un prototipo de Figma a una **arquitectura de navegación real** en React.
2. Definir e implementar un **mapa de rutas** completo y coherente con los user flows.
3. Construir una estructura de aplicación con **layouts reutilizables** (público/privado).
4. Implementar navegación funcional entre pantallas: login → dashboard → módulos → logout (según el flujo del prototipo).
5. Gestionar rutas especiales: **404, no autorizado, pantallas vacías/error/carga/éxito**.
6. Verificar cobertura: que **todas las pantallas navegables del Figma** existan y se alcancen por navegación/ruta.
7. Documentar la navegación (rutas, layouts, decisiones técnicas) de forma clara y ordenada.

Actividades del sprint

Actividad 1 – Arquitectura de rutas y layouts (React)

Objetivo: Definir y construir la base estructural de navegación.

Alcance mínimo:

- Definir un **mapa de rutas** equivalente al mapa de pantallas del Figma.
- Implementar **React Router** con:
 - Rutas públicas (ej.: login/registro)
 - Rutas privadas (ej.: dashboard/módulos)
- Crear **layouts** (ej.: **PublicLayout**, **PrivateLayout**) y navegación común.
- Añadir rutas especiales:
 - **404 / NotFound**
 - **No autorizado / Forbidden** (si procede en el prototipo)

Actividad 2 – Implementación del flujo navegable completo

Objetivo: Reproducir en código la experiencia navegable del prototipo.

Alcance mínimo:

- Flujo navegable completo: **login** → **dashboard** → **módulos** → **logout** (adaptado al prototipo de cada alumno).
- Navegación entre pantallas principales y secundarias (menú, enlaces, botones).
- Implementar placeholders cuando aún no exista lógica: pantallas “vivas” aunque muestren datos de ejemplo.

Actividad 3 – Estados de pantalla y robustez de navegación

Objetivo: Cubrir variaciones y casos reales de UX (no todo es el camino feliz).

Alcance mínimo:

- Para las pantallas clave, incluir componentes/variantes de estado:
 - **Cargando** (loading)
 - **Error**
 - **Vacío** (empty)
 - **Éxito** (success)
- Evitar “rutas muertas”: cualquier pantalla del Figma debe tener:
 - Ruta accesible y/o navegación que llegue a ella
 - Comportamiento razonable si falta información

Actividad 4 – Documentación de navegación (Figma → React)

Objetivo: Explicar qué se implementó y cómo se corresponde con el prototipo.

Alcance mínimo:

- Tabla o listado: **Ruta → Pantalla → Descripción breve → Enlace en Figma (si aplica)**.
- Explicación de decisiones: layouts, rutas protegidas, estructura de carpetas (alto nivel).
- Evidencia de cobertura: checklist de pantallas del Figma “implementadas y navegables”.

Entregables del sprint

1. **Mapa de rutas** (documentado) equivalente al mapa de pantallas de Figma.
2. Aplicación React con **navegación funcional completa** según el flujo del prototipo.
3. Implementación de pantallas especiales: **404, no autorizado, estados (loading/error/empty/success)**.
4. **Documentación completa** (Figma → rutas + decisiones técnicas).

Criterios de evaluación

Descripción	Pts
Cobertura completa de pantallas y flujos definidos en Figma (todas navegables)	4
Correcta arquitectura de rutas y layouts (claridad, orden, coherencia)	4
Robustez: 404/no autorizado + estados de pantalla (loading/error/empty/success)	2