

Fechas Importantes:

Lanzamiento: 22/09/25 > Entrega: 20/10/25

Formato de entrega:

IMPORTANTE: crear otro repositorio para resguardar el TP1, TP2 y TP3 por separado

Deberán publicar en la pestaña “Entrega TP2 Links”:

<https://docs.google.com/spreadsheets/d/1Lauy1BETMIXa2wBHyyUTSNqSmfQSafToLZjNoo55ny4/edit?usp=sharing>

Los siguientes links:

- 1- Link al repositorio, donde el docente revisará la carpeta de archivos y el read.me
- 2- Link a la web publicada en Vercel.

>>Objetivo General

Habiendo desarrollado en el primer TP un sitio web estático utilizando **HTML, CSS y JavaScript**, el propósito de este nuevo trabajo es migrar ese proyecto a **React**, transformándolo en una **SPA (Single Page Application)**. Esto implica reorganizar la estructura del sitio mediante el uso de **componentes reutilizables**, lo que permitirá mejorar la modularidad, la escalabilidad y el mantenimiento del código.

A su vez, se busca que el equipo incorpore prácticas propias del desarrollo moderno en React, tales como la gestión de rutas para la navegación interna, el uso de props y estados para dinamizar contenidos, y la integración de estilos consistentes.

>Requerimientos Generales

Repositorio y Publicación:

- Crear un repositorio en GitHub y subir el proyecto completo.
- Publicar el proyecto en la nube utilizando Vercel

Privacidad:

- Como el proyecto se publica de forma pública, no es obligatorio usar nombre y apellido completos ni una foto personal real.
- Se recomienda utilizar avatares, imágenes generadas con IA, ilustraciones o incluso fotos de mascotas como presentación individual en lugar de fotos reales de cada uno.

Estructura y Contenidos Obligatorios del Proyecto

1. **Nuevo** → Sidebar (barra lateral fija)
 - Contendrá el logo del grupo
 - Debajo, un menú vertical con enlaces a las diferentes secciones del sitio.
 - El sidebar debe ser visible y funcional en todos los dispositivos.
2. Sección Portada
 - Repite requerimientos TP1
3. Sección Bitácora
 - Repite requerimientos TP1
4. Sección Individual por cada integrante
 - Repite requerimientos TP1
5. **Nuevo** → **Sección con Datos desde Archivo JSON**
 - Deben crear un archivo JSON con al menos 20 objetos (puede ser sobre películas, libros, videojuegos, animales, etc.).
 - La aplicación debe leer este archivo y mostrar los datos en un formato de tarjetas o listado dinámico.
 - Esto demuestra la capacidad de manejar datos locales y renderizarlos dinámicamente.
6. **Nuevo** → **Sección con Datos desde API Pública**
 - Deben seleccionar una API pública (por ejemplo, de clima, noticias, películas, etc.).
 - La app debe consumir esta API y mostrar los resultados en tarjetas o listas.
 - Esto permite practicar la integración con servicios externos y manejo de datos asíncronos.
7. **Nuevo** → **Sección Árbol de Renderizado y Organización de Carpetas**

Cada equipo deberá elaborar **dos diagramas separados** que representen aspectos clave de su proyecto en React:

Árbol de renderizado (jerarquía de componentes):

Mostrar cómo se organiza la aplicación a nivel de componentes.

- Indicar qué componentes renderiza **App.jsx** y cuáles son sus hijos y subcomponentes (por ejemplo, Navbar, Home, CardIntegrante, Footer).
- Reflejar, en caso de que se utilicen, las rutas de la SPA (/bitacora, /integrantes/:id).
- Para comprender mejor este concepto, revisen la documentación oficial de React:

 [Understanding your UI as a Tree](#)

Diagrama de organización de carpetas:

- Mostrar la estructura de archivos y carpetas del proyecto.
- Incluir la ubicación de **App.jsx**, componentes, estilos CSS, imágenes y cualquier

otro recurso relevante.

-Debe reflejar buenas prácticas de organización y claridad para entender dónde se encuentra cada elemento dentro del proyecto.

Recomendación para ambos diagramas:

-Realizar gráficos en **draw.io (diagrams.net)** u otra herramienta similar. Se sugiere elegir la herramienta con la que el equipo se sienta más cómodo.

-Exportar cada diagrama en formato **PNG o JPG** para incluirlos en la entrega.

-Pensar en la **adaptabilidad** y **legibilidad** de los gráficos: deben ser claros y fáciles de entender, permitiendo comprender tanto la arquitectura de componentes como la organización del proyecto a simple vista.

Requisitos Técnicos y Funcionales Obligatorios

- React Router: Implementar rutas para que cada sección se cargue sin recargar la página, manteniendo la experiencia SPA.
- Responsividad: El diseño debe adaptarse correctamente a 3 puntos de quiebre (por ejemplo, móvil, tablet y escritorio).
- Componentización: Crear componentes reutilizables para tarjetas, menús, etc.
- Deploy en Vercel: El proyecto debe estar publicado y accesible públicamente.
- Repositorio en GitHub: Control de versiones y colaboración mediante commits claros y organizados.

Evaluación y Entrega

- Se evaluará la correcta implementación de todas las secciones y funcionalidades.
- La calidad del código, organización y uso de React Router.
- La adaptabilidad del diseño en distintos dispositivos.
- La legibilidad de los gráficos de Árbol de Renderizado y Organización de Carpetas
- La claridad y brevedad en la bitácora sobre el trabajo en equipo.
- La actualización del Archivo README.md.
- La publicación correcta en GitHub y Vercel.

>>

</TP2>