



# Desarrollo FrontEnd con React



## Objetivos

Esta práctica tiene como objetivo aplicar conceptos fundamentales de React, así como el consumo de APIs REST, gestión de estado y diseño de interfaces de usuario.



## Herramientas / Preparación

- Para alcanzar el objetivo indicado, se debe consultar las teorías y referencias bibliográficas presentadas en las clases teóricas.



## Referencias bibliográficas

<https://es.react.dev/>



## Organización

Grupal de 2 a 3 personas.



## Fecha de entrega / presentación

Fecha tope de entrega: **06/11/2024**



## Ejercicio

- 1) Desarrollar una aplicación web de noticias usando React que se conecte a una API REST de gestión de noticias. La aplicación debe permitir a los usuarios visualizar, buscar, filtrar y gestionar artículos de noticias.

### Objetivos Específicos



1. **Consumir API REST:** Conectar la aplicación a una API de noticias (se da en clase) que proporcione endpoints para obtener, crear, actualizar y eliminar noticias.
2. **Interfaz de Usuario Dinámica:** Diseñar una UI que permita a los usuarios interactuar intuitivamente con el contenido.
3. **Gestión de Estado:** Utilizar hooks de React (como `useState`, `useEffect` y `useContext` o alternativas de gestión global) para manejar datos y acciones en la interfaz.
4. **React Router:** Crear rutas de navegación para organizar el contenido de la aplicación.

### Requerimientos Técnicos

1. **Lista de Noticias:** Mostrar una lista de artículos de noticias, con información básica (título, fecha, copete).
2. **Detalle de Noticia:** Al hacer clic en un artículo, se debe mostrar su contenido completo en una página aparte.
3. **Buscar y Filtrar:** Implementar un sistema de búsqueda por palabras clave y filtros (por ejemplo, por categoría o fecha).
4. **Creación y Edición:** Incluir formularios para crear nuevos artículos y editar los existentes.
5. **Eliminación:** Permitir la eliminación de artículos de la lista, con una confirmación antes de eliminar.
6. **Notificación de Errores y Éxitos:** Informar al usuario sobre el éxito o error de cada acción (por ejemplo, creación o eliminación de artículos).

### Componentes Sugeridos

- **NewsList:** Lista de noticias disponibles.
- **NewsItem:** Representación de un artículo en la lista.
- **NewsDetail:** Vista detallada de un artículo.
- **SearchBar:** Barra de búsqueda.
- **FilterOptions:** Opciones de filtrado.
- **NewsForm:** Formulario para crear/editar noticias.

### Endpoints Sugeridos para la API REST

- **GET /news:** Obtener todas las noticias.
- **GET /news/**



: Obtener detalles de una noticia específica.

- **POST /news:** Crear una nueva noticia.
- **PUT /news/**

: Actualizar una noticia existente.

- **DELETE /news/**

: Eliminar una noticia.

### Consideraciones Extra

- **Responsividad:** Asegúrate de que la interfaz sea responsiva y se vea bien en diferentes dispositivos.
- **Estilos y Temas:** Usa una biblioteca de componentes como Material-UI o Bootstrap, o diseña tus propios estilos.
- **Pruebas:** Implementa pruebas unitarias básicas para los componentes principales y pruebas de integración para el flujo principal.

### Entrega

- Repositorio en GitHub, con instrucciones claras para la instalación y despliegue.