

React: CRUD con Firebase

Objetivo

El objetivo de esta práctica es que el alumno desarrolle una aplicación en React (con JavaScript) utilizando Firebase como repositorio de datos. La aplicación permitirá gestionar un tema elegido por el alumno, con la posibilidad de añadir, listar y buscar temas.

Requisitos Mínimos

- La aplicación debe usar React y Firebase para la persistencia de datos.
- Debe contar con una navegación fluida entre las diferentes secciones y utilizar el Router de React.
- Implementar manejo de errores y redirección a Inicio en caso de URL no encontrada.
- Permitir operaciones de lectura y creación de temas.
- Usar estilos adecuados.
- Mantener un diario de desarrollo y una documentación básica del proyecto.
- Todo el código, el diario y la documentación deben estar publicados, desde el primer momento, en un repositorio en GitHub.

Estructura Básica del proyecto

Carpetas y archivos

Components: Carpeta para los componentes generales.

- **Navbar.js**
 - Barra de navegación presente en toda la aplicación.
 - Incluye un título enlazado a Inicio.
 - Contiene la barra de búsqueda y un botón para agregar un nuevo tema.
- **List.js**
 - Muestra un listado de temas obtenidos de Firebase.
 - Cada tema tiene un enlace a su detalle (Tema.js).
- **Searchbar.js**
 - Barra de búsqueda basada en parámetros tipo query.
 - Muestra resultados en tiempo real en la página de búsqueda.

Pages: Carpeta para los componentes enrutados. Cada componente estará, a su vez, dentro de una carpeta propia. Se utilice o no, cada componente estará acompañado de su "css".

- **inicio/Inicio.js**
 - Muestra todos los temas almacenados en Firebase.

- **tema/Tema.js**
 - Muestra la información detallada de un tema.
 - Incluye botones para editar y eliminar el tema.
- **formulario/Formulario.js**
 - Contiene un formulario para agregar un nuevo tema.
 - Incluye un campo "multi-ítem" con un botón para agregar elementos.
 - Al enviar el formulario, redirige a Inicio.
- **busqueda/Busqueda.js**
 - Muestra los resultados de la búsqueda de temas en tiempo real.

Desarrollo Propuesto

1. **Configuración del entorno:**
 - Crear un proyecto en React.
 - Configurar Firebase y conectar la base de datos.
 - Instalar dependencias necesarias.
 - Publicar el repositorio en GitHub.
2. **Implementación de componentes generales:**
 - Crear Navbar, List y Searchbar.
3. **Creación de las páginas enrutadas:**
 - Implementar Inicio.js para mostrar los temas.
 - Desarrollar Tema.js para mostrar información detallada.
 - Crear Formulario.js para añadir temas.
 - Programar Busqueda.js para mostrar resultados de búsqueda.
4. **Operaciones CRUD:**
 - Implementar la creación, edición y eliminación de temas.
5. **Manejo de errores y redirecciones:**
 - Gestionar errores y redirigir a Inicio si la URL no existe.
6. **Estilizado y mejoras:**
 - Agregar diseño atractivo y elementos visuales.

Diario de desarrollo

El alumno deberá llevar un diario de desarrollo del proyecto, donde documentará:

- Tareas realizadas en cada sesión de trabajo.
- Problemas encontrados y soluciones aplicadas.
- Mejoras o funcionalidades adicionales implementadas.

El diario puede mantenerse en un documento de texto o cualquier otro formato accesible y debe subirse al repositorio de GitHub junto con el código. Debe actualizarse diariamente.

Documentación del proyecto

Al finalizar la práctica, cada alumno deberá entregar una documentación básica del proyecto que incluya:

- **Introducción:** Breve descripción del proyecto y su propósito.
- **Instalación y configuración:** Pasos para ejecutar el proyecto en un entorno local.
- **Estructura del código:** Explicación breve de los archivos y carpetas principales.
- **Funciones implementadas:** Descripción de las principales características y funcionalidades.
- **Mejoras y futuras implementaciones:** Sugerencias para mejoras futuras.

Esta documentación debe estar incluida en el repositorio de GitHub.

Valor añadido

Se valorará positivamente:

- Permitir operaciones CRUD completas (a leer y crear se debería añadir actualizar y eliminar) sobre los temas.
- Agregar elementos visuales (imágenes, iconos, etc.).
- Cualquier otra mejora adicional.

Fechas Importantes

- **Inicio:** 13/02/2025
- **Final y Entrega:** 06/03/2025

Evaluación

Se valorará:

- Implementación correcta de React y Firebase.
- Uso adecuado de JavaScript.
- Eficiencia en la captura de datos.
- Eficiencia en el tratamiento de datos.
- Uso correcto del enrutamiento.
- Estilizado y diseño atractivo y funcional.
- Manejo adecuado de errores y redirecciones.
- Uso de una barra de búsqueda eficiente.
- Funcionalidad de CRUD.
- Uso de elementos visuales.
- Inclusión de mejoras adicionales.
- Calidad del diario de desarrollo y documentación del proyecto.
- Correcto uso de GitHub para almacenar código, documentación y avances.

Propuesta de temas

El alumno ha de elegir el tema de su aplicación, que ha de ser exclusivo.

Algunas sugerencias:

- **Hotel:** Permite gestionar diferentes tipos de habitaciones y servicios.
- **Aviones:** Base de datos de modelos de aviones con detalles.
- **Libros:** Biblioteca de libros con información detallada.
- **Videojuegos:** Listado de juegos con género, plataformas y detalles.
- **Películas:** Base de datos con títulos, géneros, directores y reseñas.
- **Ciudades:** Información sobre ciudades del mundo, población, cultura y atractivos turísticos.
- **Eventos:** Gestión de eventos con fechas, ubicaciones y descripciones.
- **Mascotas:** Registro de mascotas con datos como raza, edad y cuidados especiales.
- **Instrumentos musicales:** Listado con características, historia y tipos de instrumentos.
- **Plantas:** Base de datos de plantas con información sobre cuidados y tipos de suelo.
- **Coches:** Registro de automóviles con especificaciones técnicas y comparaciones.
- **Superhéroes:** Información sobre héroes, habilidades y universos ficticios.
- **Deportes:** Detalles sobre distintos deportes, reglas y equipos destacados.
- **Personajes históricos:** Biografías y logros de personajes importantes en la historia.

Cada tema debe incluir información relevante y aplicar las funcionalidades descritas en la práctica.