Desarrollo de un Front-End utilizando React

Objetivos

Las tres actividades que se llevarán a cabo en la asignatura “*Desarrollo Web Full Stack*” tienen como objetivo:

* **Desarrollar un Front-End haciendo uso de React**.
* **Desarrollar un Back-End que exponga APIs REST** mediante microservicios con Java, Spring, Eureka y Zuul.
* **Integrar ambas partes y desplegar el Back-End mediante contenedores Docker**. De forma opcional, se podrá desplegar toda la aplicación en Heroku.

**Esta primera actividad consiste en la realización del Front-End de la aplicación web**. Por ello, esta actividad permite poner en práctica lo estudiado en los cinco primeros temas:

* Creación de páginas con HTML.
* Aplicación de estilos con CSS
* Manejo de JavaScript.
* Uso de la librería React
  + Orientación a componentes.
  + Hooks.
  + Router.

Pautas de elaboración

Esta actividad abarca la creación del Front-End de una aplicación web haciendo uso de los lenguajes estudiados hasta el momento: HTML, CSS y JavaScript. Además, haremos uso de la librería React que se ha estudiado en el tema 5.

El Front-End desarrollado deberá hacer uso de los principales conceptos que se han estudiado en los primeros cinco temas:

* **Uso de la librería React** para construir un frontal que utilice componentes funcionales (**mínimo 10**), haciendo uso en algún momento de JSX (por ende, HTML).
* **Uso de los hooks estudiados**, *useState* y *useEffect*.
* **Implementación de un *custom hook*** a elección.
* **Uso de *React Router*** para implementar diferentes rutas en la aplicación (**como mínimo 4**).
* **Aplicar estilos** mediante hojas de estilo CSS.

**La temática de la aplicación es libre**. **No obstante**, para ser considerada válida, la idea de la aplicación a desarrollar **deberá satisfacer los siguientes requisitos** (*su implementación no forma parte de esta actividad, pero es necesario que la idea sea compatible con ellos*):

* **Debe requerir de algún tipo de base de datos,** que contendrá elementos característicos de la aplicación que se desea construir.
* **Debe requerir un buscador** que se usará para interactuar con la base de datos.
* **Debe poder realizar una tarea principal**, aquella para la que está concebida la aplicación.

Algunos arquetipos (que se recomienda utilizar como idea base de la aplicación) son los siguientes:

* **Tienda online**: Una aplicación que ofrecerá a los usuarios una gran cantidad de productos (cada producto tendrá atributos básicos como nombre, empresa asociada, descripción corta, descripción larga, imagen del producto, precio…) para poder comprarlos.
* **Biblioteca online**: Una aplicación que se servirá de una colección de libros (que tendrán como atributos el nombre, autor, año de publicación, códigos ISBN de 10 y 13 dígitos, imagen de portada, sinopsis, críticas…) para que los usuarios puedan alquilarlos por un tiempo determinado.
* **Plataforma de Streaming online**: Una aplicación similar a otras muy conocidas hoy en día que mostrará a los usuarios un gran catálogo de películas o documentales (con atributos como el nombre, director, año de publicación, sinopsis, críticas, duración, imagen de la película…). En lugar de reproducir la película o documental, se reproducirá el tráiler de YouTube. Los usuarios podrán alquilar estas películas durante un tiempo determinado.
* **Aplicación de inventario**: Una aplicación que permitirá controlar el inventario de una tienda, taller o almacén permitiendo listar y buscar todos aquellos elementos contenidos en una base de datos. También permitirá realizar pedidos a un distribuidor externo para pedir stock de los productos que se considere.

Como puedes observar, todos los ejemplos cumplen con los requisitos que se han indicado. Es muy recomendable que elijas uno de estos cuatro tipos de aplicación. No obstante, puedes realizar cualquier otro tipo de aplicación siempre que cumpla con dichos criterios.

Para no complicar demasiado el desarrollo, **no será obligatorio incluir en la aplicación ningún mecanismo de *login***. Asumiremos que la aplicación se usa de forma anónima y todas aquellas vistas que estén destinadas a gestión o administración serán accesibles a través de una ruta especial concreta. No obstante, se valorará positivamente incluir esta característica.

Dado que no trabajaremos con ninguna base de datos en esta actividad, para poder mostrar datos en las pantallas utilizaremos datos de prueba (*mocks*). No es necesario que el frontal, en este punto, realice ningún tipo de petición HTTP para obtener datos del Back-End.

La entrega consistirá en un archivo comprimido que contendrá:

* Memoria de la actividad.
* Archivo comprimido con el código de la aplicación.

Extensión y formato de la memoria

La memoria tendrá una extensión máxima de 20 páginas, y deberá incluir los siguientes apartados (en este orden):

1. **Portada**
2. **Índice**
3. **Introducción**: Se hará una introducción al contenido de la memoria, indicando en primer lugar qué tipo de aplicación se ha decidido construir y por qué motivos.
4. **Componentes de React**: Se listarán y explicarán brevemente todos los componentes de React que forman parte de la solución desarrollada
5. **Hooks utilizados**: Se listarán todos los hooks utilizados en la solución desarrollada, indicando en qué componente se han utilizado. Además, se incluirán aquí los hooks personalizados que se hayan creado, indicando por qué son necesarios y el valor que aportan a la solución.
6. **Vistas**: Se mostrarán capturas de pantalla del Front-End desarrollado, describiendo brevemente su contenido y haciendo énfasis en los estilos CSS que se hayan utilizado.
7. **Conclusiones**
8. **Referencias bibliográficas:** Si procede, en formato APA.

Rúbrica

|  |  |  |  |
| --- | --- | --- | --- |
| Título de la actividad | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | Implementación de componentes funcionales en React (mínimo 10) haciendo uso de JSX. | 3 | 30% |
| Criterio 2 | Uso de los *hooks* principales de React: *useState* y *useEffect*. | 1 | 10% |
| Criterio 3 | Implementación de un *custom hook* que sea utilizado en algún componente funcional. | 2 | 20% |
| Criterio 4 | Uso correcto de React Router para crear diferentes rutas dentro de la aplicación. | 2 | 20% |
| Criterio 5 | Modificación del estilo de la aplicación mediante el uso de CSS | 1 | 10% |
| Criterio 6 | Elaboración de la memoria según la especificación, cuidando la presentación y la corrección ortográfica de la misma | 1 | 10% |
|  |  | **10** | **100 %** |