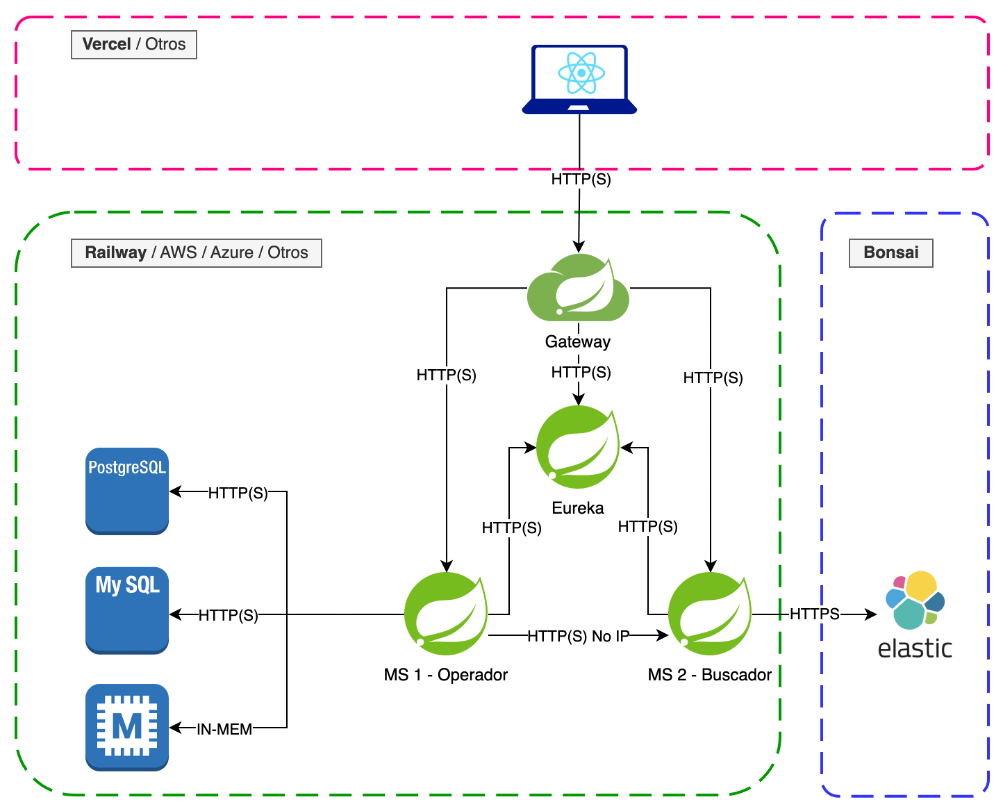
Actividad 1: Desarrollo de un front-end utilizando React

Objetivos

Las tres actividades que se llevarán a cabo en la asignatura Desarrollo Web Integral tienen como objetivo:

* Desarrollar un front-end haciendo uso de HTML5, CSS3, JavaScript y React.
* Desarrollar un back-end haciendo uso de una arquitectura orientada a microservicios, donde cada microservicio expondrá una API REST haciendo uso de Java y Spring.
* Integrar ambas partes y desplegar front-end y back-end tanto en local como de forma pública.
* Hacer uso de bases de datos relacionales y del motor de búsqueda Elasticsearch.



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 HTML5.
* Aplicación de estilos con CSS3 y metodología BEM.
* Manejo de JavaScript (ECMAScript V6).
* Uso de la librería React.
  + Orientación a componentes.
  + *Hooks.*
  + React Router V6.

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: React.

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)** y que haga uso de 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. Se permite utilizar Bootstrap, Tailwind, SCSS… Pero **se debe usar la metodología BEM**.

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,** el cual se usará para interactuar con la base de datos.
* Debe poder realizar **al menos dos tareas principales,** aquellas para las que está concebida la aplicación.

Algunos **arquetipos** (que se recomiendan 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, etc.) para poder comprarlos.
  + **Funcionalidad mínima necesaria**: compra de uno o varios productos. Devolución de pedidos.
  + **Base de datos**:contiene información de productos, compras y devoluciones.
  + **Buscador**: sobre los productos de la tienda (nombre, categoría, fabricante y otros atributos específicos del tipo de producto a vender).
* **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, etc.) para que los usuarios puedan alquilarlos por un tiempo determinado.
  + **Funcionalidad mínima necesaria**: alquiler de libros. Extensión de plazo de alquiler.
  + **Base de datos**: contiene información de libros y alquileres.
  + **Buscador**: sobre los diferentes libros de la biblioteca (nombre, código ISBN o equivalente, categoría, autor, idioma y descripción).
* **Plataforma de *streaming* online:** una aplicación similar a otras muy conocidas hoy en día. 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, etc.). En lugar de reproducir la película o documental, se reproducirá el tráiler de YouTube. Los usuarios podrán alquilar estas películas durante un tiempo determinado.
  + **Funcionalidad mínima necesaria**: alquiler de películas. Compra de películas.
  + **Base de datos**: contiene información de películas, alquileres y compras.
  + **Buscador**: sobre las películas disponibles (nombre, sinopsis, categoría, año, director, idioma y actores).
* **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.
  + **Funcionalidad mínima necesaria**: petición de material a proveedores. Creación de pedidos para ser enviados a usuarios.
  + **Base de datos**: contiene información de productos, pedidos a proveedores y envíos a clientes.
  + **Buscador**: sobre los productos de la aplicación (nombre, categoría, fabricante y otros atributos específicos del tipo de producto).

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. En caso de elegir una idea diferente, **debes asegurarte de que cumple con los requisitos indicados. De no ser así, podrías suspender la actividad, por lo que se recomienda que valides con tu docente que la idea es válida**.

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 (en caso de haberlas) serán accesibles a través de una ruta diferente. No obstante, se valorará positivamente incluir esta característica.

Dado que no trabajaremos con ninguna base de datos ni back-end 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 peticiones por HTTP para obtener datos del back-end. Si lo deseas, puedes usar APIs de datos estáticos de prueba (públicas en Internet).

**Formato:**

La entrega consistirá **un único archivo ZIP** con los siguientes archivos:

* Vídeo memoria **obligatoria** en formato MP4 de la actividad.
* Archivo TXT con la **URL del proyecto desplegado** en Vercel (u otros).
* **Proyecto con el código** de la aplicación. **Sin node\_modules**.

¿Qué es una vídeo memoria?

La vídeo memoria es una grabación donde explicarás con detalle el software que has desarrollado. Si estás muy orgulloso del software que has construido, podrías incluso publicar este vídeo de forma pública. La vídeo memoria tendrá una **duración aproximada de 5 minutos** y deberá visitar los siguientes aspectos de tu actividad:

1. **Introducción:** se hará una introducción a la aplicación. Se indicará qué tipo de aplicación se ha decidido construir y por qué motivos.
2. **Componentes de React:** se listarán y explicarán brevemente todos los componentes de React que forman parte de la solución desarrollada. Hablarás aquí también del enrutador que has utilizado.
3. **Hooks utilizados:** se listarán todos los hooks utilizados en la solución desarrollada y en qué componentes 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.
4. **Vistas:** mostrarás cómo el proyecto puede arrancarse en local. Una vez desplegado en local, se mostrarán las diferentes vistas del front-end desarrollado, junto con una descripción breve de su contenido, que haga énfasis en los estilos CSS que se utilizaron (los más destacables, no es necesario hablar de todo el código CSS).
5. **Despliegue:** se mostrará el proyecto desplegado en Vercel correctamente.

Rúbrica

|  |  |  |  |
| --- | --- | --- | --- |
| Criterio | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | Implementación de componentes funcionales en React (mínimo 10) haciendo uso de JSX. | 2 | 20 % |
| 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. | 1 | 10 % |
| Criterio 5 | Modificación del estilo de la aplicación mediante el uso de CSS. Uso de metodología BEM. | 1 | 10 % |
| Criterio 6 | Implementación de las vistas necesarias para cubrir la funcionalidad mínima de la aplicación. | 1,5 | 15 % |
| Criterio 7 | **Elaboración de la vídeo memoria** **obligatoria** según las especificaciones, ajustándose a la duración aproximada y cubriendo los aspectos solicitados. | 1 | 10 % |
| Criterio 8 | Despliegue en Vercel | 0,5 | 5 % |
|  |  | **10** | **100 %** |