Text

Description automatically generated with medium confidence

## **Actividad4 - Tecnología front end en la construcción de una aplicación web II**

**Asignatura**

# Electiva disciplinar II - Desarrollo de aplicaciones web II

# Presenta

Saray Alexandra Cotes Cotes

# Docente

# Magda Fernández

Medellín – Colombia 2023

**Link del Repositorio con el Código de la aplicación:**

<https://github.com/Saraycotes92/InventariosOptimusFly>

**Introducción**

El desarrollo de aplicaciones web modernas requiere un conocimiento sólido de una variedad de tecnologías y conceptos. En este texto, vamos a explorar varios de estos conceptos, incluyendo REST con Swagger, ReactJS, Hooks (useState, useContext, useEffect, useReducer), Context API, Peticiones HTTP con Axios, Rutas y navegación, y el proceso de despliegue. Estos conceptos fundamentales son esenciales para cualquier desarrollador web moderno.

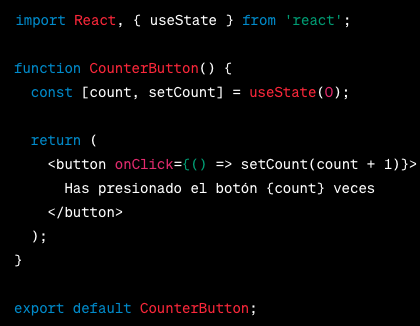
**Contenido**

**REST con Swagger**

REST, o Transferencia de Estado Representacional, es un estilo arquitectónico para diseñar redes de software. Se centra en el uso de recursos y cómo se pueden manipular utilizando protocolos estándar HTTP. Swagger, por otro lado, es una herramienta útil para diseñar y construir APIs REST de una manera fácil y estándar. Por ejemplo, en un proyecto de desarrollo de una API para una biblioteca, Swagger podría usarse para documentar cómo se puede acceder a los recursos del libro a través de la API.  
  
Ejemplo: Supongamos que tienes una API que gestiona usuarios. Podrías utilizar Swagger para documentar los endpoints de tu API, como GET /users para obtener una lista de usuarios, POST /users para crear un nuevo usuario, etc.

**ReactJS**

ReactJS es una biblioteca de JavaScript para construir interfaces de usuario. React permite a los desarrolladores crear grandes aplicaciones web que pueden cambiar los datos sin recargar la página. Un ejemplo de uso de React podría ser la construcción de una aplicación de lista de tareas, donde los elementos de la lista pueden ser agregados, eliminados o marcados como completados sin necesidad de recargar la página.

Ejemplo: Un componente React básico podría ser un botón con un contador. Cada vez que el botón es presionado, el contador aumenta.

**Hooks (useState, useContext, useEffect, useReducer)**

Los Hooks son funciones que te permiten "enganchar" el estado y el ciclo de vida de React desde componentes funcionales. useState te permite agregar estado a tu componente, useContext te permite acceder al contexto sin necesidad de anidamiento, useEffect te permite realizar efectos secundarios en tu componente y useReducer te permite manejar un estado más complejo con una interfaz similar a Redux.

Ejemplo: Podríamos usar useEffect para realizar una petición HTTP cuando un componente se monta.



**Context API**

La Context API de React permite compartir valores que se consideran globales para un árbol de componentes de React. Por ejemplo, podría utilizarse para compartir el idioma actual de la aplicación entre todos los componentes para que sepan qué idioma

Ejemplo: Podríamos utilizar la Context API para compartir el tema actual en toda la aplicación.

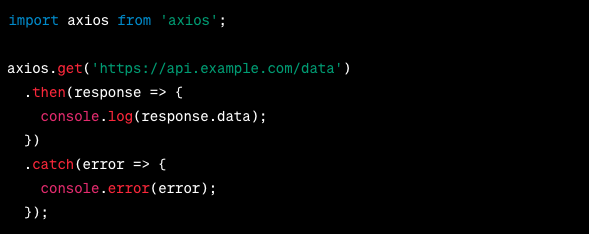
****

**Peticiones HTTP con Axios**

Axios es una biblioteca de JavaScript muy popular que se utiliza para hacer solicitudes HTTP desde el# Let's look up more recent and detailed examples for the remaining concepts. Starting with "HTTP requests with Axios examples".

search("HTTP requests with Axios examples")

Ejemplo: Podríamos usar Axios para hacer una petición GET a una API y obtener datos.



**Rutas y navegación:** as rutas y la navegación son una parte esencial de cualquier aplicación web. Permiten a los usuarios navegar entre diferentes partes de la aplicación y mantener el estado de la aplicación a medida que navegan.

Ejemplo: Podríamos usar la biblioteca de enrutamiento de React Router para configurar las rutas de nuestra aplicación

**Despliegue**

El despliegue es el proceso de poner en línea una aplicación web para que los usuarios puedan acceder a ella. Involucra empaquetar la aplicación, preparar el entorno de producción y poner en marcha la aplicación.

Ejemplo: Podrías utilizar un servicio como Netlify o Vercel para desplegar tu aplicación React. Estos servicios te permiten desplegar tu aplicación directamente desde un repositorio de GitHub. Solo tienes que conectar tu repositorio, configurar el comando de construcción (npm run build para una aplicación creada con Create React App) y el directorio de salida (build para una aplicación creada con Create React App), y el servicio se encargará del resto.

**Bibliografía**

* Pérez Ibarra, S.  G., Quispe, J. R., Mullicundo, F. F. y Lamas, D. A. (15-16 de abril de 2021). Herramientas y tecnologías para el desarrollo web desde el frontend y backend. [Objeto de conferencia]. XXIII Workshop de Investigadores en Ciencias de la Computación, Chilecito, La Rioja.
* Valdivia Caballero, J. J. (2021). Modelo de procesos para el desarrollo del front-end de aplicaciones web. [Tesis de maestría, Universidad Nacional Mayor de San Marcos]. (pp. 5-40).