



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN NRC: 10522
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN
DESARROLLO DE APLICACIONES WEB

GRUPO 2
INTEGRANTES

Acosta Moreno Camilo Alejandro

Almeida Logroño Nicolás André

Benavides Mina Luis Mauricio

Díaz Peñafiel Angie Elizabeth

Moreno Cardozo Paola Natalia

Moreno Cardozo Wendy Piedad

Carlos Andrés Pillajo Bolagay

Fecha: 22 de agosto de 2023

Implementación de Front-End Web Development en la aplicación “Mundos de Papel”	3
Estructura del repositorio	3
Introducción	3
Objetivos	4
Objetivo General	4
Objetivos específicos	4
Marco teórico	5
Requerimientos	5
Documentación	5
Diagramas	5
Pruebas.....	6
Conclusiones	11
Recomendaciones	11
Referencias.....	11

Implementación de Front-End Web Development en la aplicación “Mundos de Papel”

Estructura del repositorio

https://github.com/Camilo2600/Team_2_Web_A1_A2_A3

Team_2_Web_A1_A2_A3 Public	
main 1 branch 0 tags	
NicolasAndreAlmeida Add files via upload	
01-Definition	Definición V02
02-Requerimientos	Add files via upload
03-Documentación	Documentación V03
04-Diagramas	Add files via upload
05-Pruebas	Add files via upload
06-Código	Add files via upload
07-Otros	Evaluación Actividad 1 y 2
08 Arquitectura	Add files via upload

Introducción

El presente proyecto tiene como objetivo desarrollar dos sitios web utilizando React y otras bibliotecas de Front-End. En el primer sitio web, se implementarán interfaces de usuario diseñadas previamente para consumir una API pública que devuelve datos de publicaciones científicas. Además, se trabajará en la implementación de una página que consumirá una URI desarrollada en la actividad anterior del proyecto. En el segundo sitio web, se diseñarán interfaces de usuario para el proyecto temático del equipo, consumiendo información de una URI mediante solicitudes GET. A lo largo de este proyecto, se hará uso de tecnologías de desarrollo web modernas para lograr una experiencia interactiva y funcional.

Objetivos

Objetivo General

El objetivo general de este proyecto es desarrollar dos sitios web utilizando React y tecnologías de Front-End que permitan consumir y mostrar información de una API pública, así como implementar interfaces de usuario que consuman datos de una URI propia desarrollada en la actividad anterior.

Objetivos Específicos

- Implementar interfaces de usuario para consumir datos de la API pública de publicaciones científicas proporcionada por <https://api.plos.org/>.
- Diseñar y desarrollar una página web que permita al usuario buscar y visualizar documentos científicos a través de la API pública.
- Crear una página que muestre los detalles de un artículo científico seleccionado, incluyendo su resumen, en una nueva pestaña del navegador.
- Desarrollar una página que muestre la información de la computadora cliente que accede a la API, formateando los datos en una página web.
- Diseñar interfaces de usuario para el proyecto temático del equipo utilizando React o un marco de Front-End similar.
- Implementar al menos una página web que consuma información de una URI desarrollada en la actividad anterior del proyecto.
- Aprender y aplicar el uso de sesiones en las capas de back-end y front-end, verificando su correcta implementación.
- Integrar las soluciones individuales de cada miembro del equipo en un único repositorio en GitHub bajo la carpeta "front-end".

- Crear un video que muestre la funcionalidad de las páginas web, incluyendo la interacción con la API pública y la URI propia, así como los datos en formato JSON y su relación con la base de datos MongoDB Atlas.

Marco Teórico

En este proyecto, se aplicarán conceptos de desarrollo web Front-End utilizando React, una biblioteca de JavaScript ampliamente utilizada para la construcción de interfaces de usuario interactivas. Además, se emplearán bibliotecas y marcos de Front-End para mejorar la eficiencia y experiencia del usuario en la creación de interfaces atractivas y funcionales.

El consumo de API (Application Programming Interface) es una práctica común en el desarrollo web moderno, permitiendo que las aplicaciones accedan y utilicen datos de otras fuentes a través de solicitudes HTTP. En este proyecto, se consumirá la API pública de <https://api.plos.org/> para obtener información sobre publicaciones científicas.

Además, se implementarán sesiones para gestionar la autenticación y el estado del usuario, permitiendo que la aplicación recuerde datos entre diferentes páginas y sesiones. Esto es fundamental para brindar una experiencia coherente y personalizada al usuario durante su interacción con las interfaces desarrolladas.

En resumen, este proyecto combina el uso de tecnologías de desarrollo web Front-End como React, consumo de API, diseño de interfaces de usuario, y la implementación de sesiones para crear dos sitios web funcionales y atractivos que cumplen con los objetivos establecidos.

Requerimientos

Matriz de Marco de Trabajo de HU													
ITEM	PROBLEMA	QUE (NECESIDAD)	PARA QUÉ (SOLUCIÓN)	PARA QUIEN (USUARIO)	COMO (DESCRIPCIÓN DE TAREAS)	HECHO POR (PROG. RESP.)	CUANTO TIEMPO (ESTIMADO EN HORAS)	FECHA DE ENTREGA	PRIORIDAD	STATUS	PRUEBA (COMO SE VERIFICA)	COMENTARIOS	NOMBRE DE HISTORIA
REQ001	Permitir el acceso de inicio de sesión al usuario	Ingresar al aplicativo web	El usuario pueda ingresar con usuario y contraseña	Administrador/ personal	Ingreso y validación: Usuario (nombre y apellido) y contraseña	Camilo Acosta	20	2023-08-19	Alta	Terminado	Mostrar al Usuario/Administrador: Sus datos son correctos/ Sus datos son incorrectos, vuelva a intentar	Se debe validar que el usuario y la contraseña sean validos.	Login
REQ002	Permitir añadir producto	Ingresar los datos del producto	Agregar un nuevo producto en la base de datos	Administrador/ personal	Ingresar: ID, Nombre, Descripción, Precio, Cantidad	Mauricio Benavides	20	2023-08-19	Alta	Terminado	Al presionar el boton de guardar se debe almacenar en la base de datos y mostrarse en listar productos.	Tambien se puede visualizar los botones para limpiar los campos y listar productos.	Ingresar producto
REQ003	Permitir añadir proveedor	Ingresar los datos del proveedor	Agregar un nuevo proveedor en la base de datos	Administrador/ personal	Ingresar: ID, Nombre, Direccion, Telefono, Tipo de producto, Total	Angie Diaz	20	2023-08-19	Alta	Terminado	Al presionar el boton de guardar se debe almacenar en la base de datos y mostrarse en listar proveedores.	Tambien se puede visualizar los botones para limpiar los campos y listar proveedores.	Ingresar proveedor
REQ004	Permitir añadir cliente	Ingresar los datos del cliente	Agregar un nuevo cliente en la base de datos	Administrador/ personal	Ingresar: ID, Nombre, Apellido, Email, Telefono	Nicolas Almeida	20	2023-08-19	Alta	Terminado	Al presionar el boton de guardar se debe almacenar en la base de datos y mostrarse en listar clientes.	Tambien se puede visualizar los botones para limpiar los campos y listar clientes.	Ingresar cliente
REQ005	Permitir visualizar la API Publica de PloOne	Listar los articulos de las API Publica	Filtrar los articulos disponibles en la API Publica	Administrador/ personal	Debe escribir el nombre del articulo que se quiere encontrar	Wendy Moreno	20	2023-08-19	Alta	Terminado	En la barra del buscador se escribe el nombre de cualquier de los articulos disponibles para empezar a filtrar		Listar articulos de API Publica

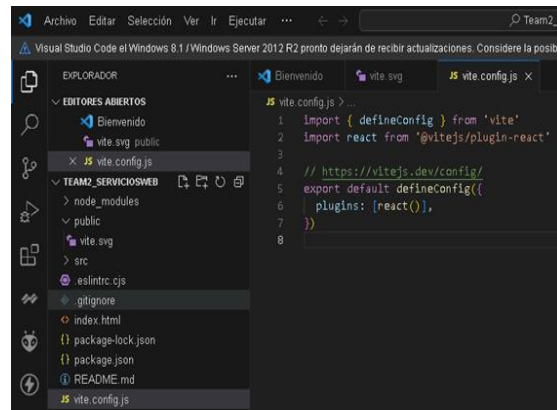
HISTORIA DE USUARIO (HU)

ITEM REQ001	USUARIO Administrador/personal	STATUS Terminado		
TIEMPO 20	PRIORIDAD Alta	PROG. RESP Camilo Acosta		
QUE Ingresar al aplicativo web	PARA QUE El usuario pueda ingresar con usuario y contraseña	COMO Ingreso y validación: Usuario (nombre y apellido) y contraseña		
NOMBRE HISTORIA Login				
PRUEBA Mostrar al Usuario/Administrador: Sus datos son correctos/ Sus datos son incorrectos, vuelva a intentar			COMENTARIOS Se debe validar que el usuario y la contraseña sean validos.	

https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/02-Requerimientos

Documentación

Código Fuente



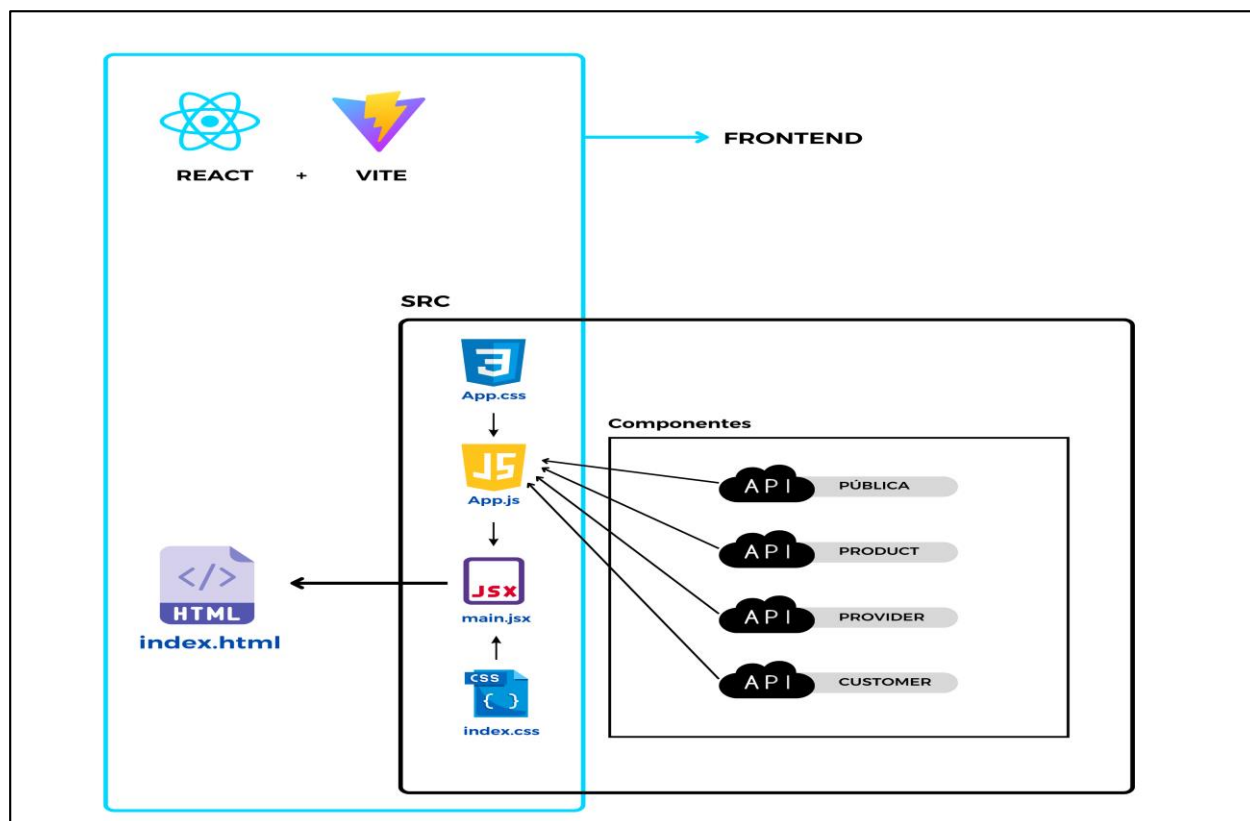
```

vite.config.js
1 import { defineConfig } from 'vite'
2 import react from '@vitejs/plugin-react'
3
4 // https://vitejs.dev/config/
5 export default defineConfig({
6   plugins: [react()],
7 })
8

```

https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/06-C%C3%B3digo/Act-3

Arquitectura del Sistema



La arquitectura del sistema Front-End basado en React y Vite sigue un enfoque modular y bien estructurado para garantizar la escalabilidad, el mantenimiento y la claridad del código. Se organiza en diferentes componentes y archivos que representan las diferentes capas y responsabilidades de la aplicación. A continuación, se describe la arquitectura del sistema utilizando terminología de arquitectura de software:

1. Capa de Presentación:

En esta capa, los componentes de la interfaz de usuario se definen y representan visualmente. Se compone de archivos en el directorio "src" que incluyen componentes, estilos y archivos de configuración.

- **index.html:** Punto de entrada HTML de la aplicación, que incluye los elementos raíz de la interfaz y vincula los archivos CSS y JavaScript necesarios.

- **index.css:** Archivo de estilo global que define reglas de estilo aplicables a toda la aplicación.
- **App.js:** Componente principal de la aplicación que envuelve todos los demás componentes. Sirve como punto de entrada para el enrutamiento y la navegación entre diferentes páginas.
- **Componentes:** Directorio que contiene componentes reutilizables de la interfaz de usuario, como formularios, botones y elementos visuales.

2. Capa de Lógica de Aplicación:

Aquí es donde reside la lógica de la aplicación, incluyendo la gestión de datos y las interacciones con las API.

- **API pública, API producto, API cliente, API proveedor:** Componentes que encapsulan la lógica para interactuar con las diferentes API, abstrayendo los detalles de las solicitudes y respuestas HTTP.
- **Main.jsx:** Este archivo es responsable de la configuración principal de la aplicación, incluido el enrutamiento, la definición de rutas y la composición de los componentes para mostrar en función de la navegación del usuario.

3. Capa de Datos:

La capa de datos se encarga de la gestión y el almacenamiento de los datos utilizados en la aplicación.

- **Componentes API:** Estos componentes realizan llamadas a las API públicas y gestionan las respuestas, extrayendo y procesando los datos necesarios para mostrar en la interfaz de usuario.

4. Capa de Infraestructura:

En esta capa, se manejan aspectos de configuración y dependencias.

- **Vite (node_modules):** Directorio que almacena las dependencias y paquetes necesarios para la construcción y ejecución de la aplicación Vite.

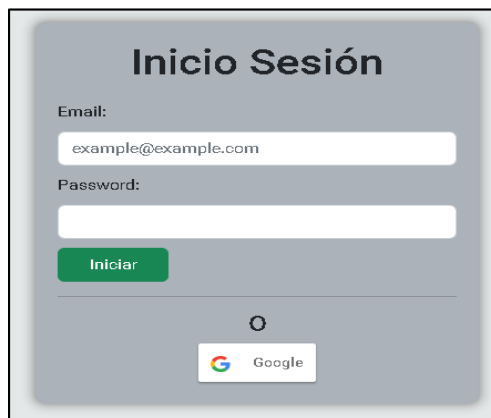
Resumen:

La arquitectura del sistema se basa en un enfoque modular y bien organizado. La capa de presentación maneja la visualización y la interacción del usuario, la capa de lógica de la aplicación gestiona las operaciones y la lógica detrás de la interfaz, y la capa de datos se encarga de la obtención y procesamiento de información. La estructura de archivos y componentes favorece la mantenibilidad, escalabilidad y claridad del código, permitiendo un desarrollo eficiente y la incorporación de nuevas funcionalidades de manera ordenada.

Diseño de las interfaces

Pestaña de Inicio de Sesión

- Diseñar una interfaz de usuario atractiva con campos de entrada para correo electrónico y contraseña.
- Implementar la lógica de autenticación para permitir que los usuarios inicien sesión en la plataforma.
- Proporcionar enlaces o botones para que los nuevos usuarios se registren en la plataforma.
- Utilizar rutas y navegación para cambiar entre las pestañas y mantener una experiencia fluida.



The image shows a mockup of a login interface. At the top, the title "Inicio Sesión" is displayed in a bold, dark font. Below the title, there are two input fields: one for "Email:" with the placeholder text "example@example.com", and another for "Password:". Below the password field is a green button with the text "Iniciar". At the bottom of the form, there is a small circular icon and a "Google" login button with the Google logo.

Nos registramos con la cuenta de Gmail para poder acceder a la página web y comprobar que el Front end funciona correctamente.



Pestaña de Productos:

- Diseñar una página que muestre una lista de productos disponibles.
- Implementar un formulario para agregar nuevos productos con campos como nombre, descripción y precio.
- Permitir la edición y eliminación de productos existentes mediante operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

INICIO PRODUCTOS ▾ PROVEEDORES ▾ CLIENTES ▾ API PUBLICA

Registrar Producto

ID
1

Nombre
Estadística

Descripción
Tomo 2

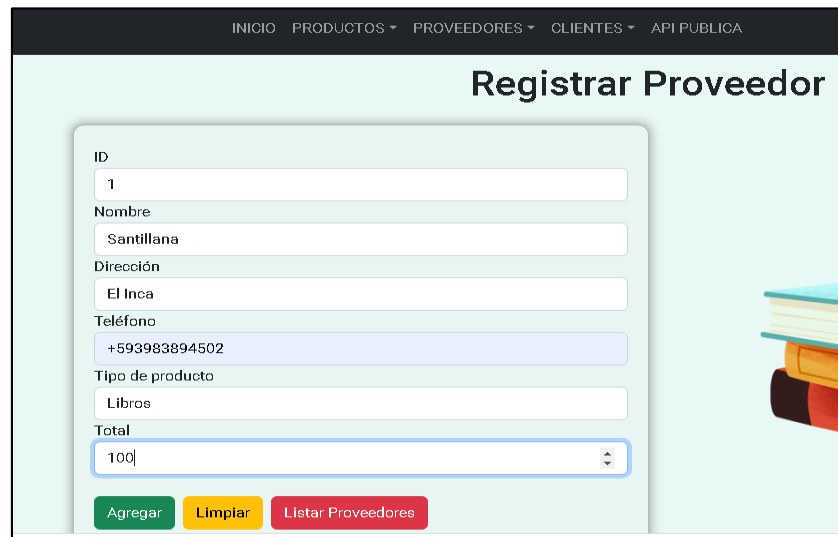
Precio
35.60

Cantidad
20

Agregar Limpiar Listar Productos

Pestaña de Proveedores:

- Crear una página similar a la de productos para mostrar una lista de proveedores.
- Implementar un formulario para agregar nuevos proveedores con campos como nombre, dirección y contacto.
- Habilitar la edición y eliminación de proveedores existentes utilizando operaciones CRUD.



The screenshot shows a web application interface with a dark navigation bar at the top containing links: INICIO, PRODUCTOS, PROVEEDORES, CLIENTES, and API PUBLICA. The main content area has a light blue background and is titled 'Registrar Proveedor'. It features a form with the following fields: ID (value: 1), Nombre (value: Santillana), Dirección (value: El Inca), Teléfono (value: +593983894502), Tipo de producto (value: Libros), and Total (value: 100). At the bottom of the form are three buttons: 'Agregar' (green), 'Limpiar' (yellow), and 'Listar Proveedores' (red). To the right of the form is a decorative illustration of a stack of books.

Pestaña de Clientes:

- Desarrollar una página similar a la de productos y proveedores para mostrar una lista de clientes.
- Implementar un formulario para agregar nuevos clientes con campos como nombre, correo electrónico y número de teléfono.
- Permitir la edición y eliminación de clientes existentes mediante operaciones CRUD.

INICIO PRODUCTOS PROVEEDORES CLIENTES API PUBLICA

Registrar Cliente

ID
1

Nombre
Angie

Apellido
Diaz

Email
angiediaz@gmail.com

Teléfono
3

Agregar Limpiar Listar Clientes

Pestaña de API Pública:

- Diseñar una página donde los usuarios puedan ingresar un término de búsqueda.
- Mostrar resultados de la búsqueda en una tabla con las columnas identificadas previamente (identificación, diario, eissn, etc.).
- Implementar enlaces en los títulos de los artículos para mostrar los resúmenes en una nueva página o pestaña.
- Utilizar llamadas a la API pública proporcionada para obtener los datos de las publicaciones científicas según el término de búsqueda ingresado.

← Artículo 🔍

ARTÍCULOS

OPCIONES

GENERAR URL

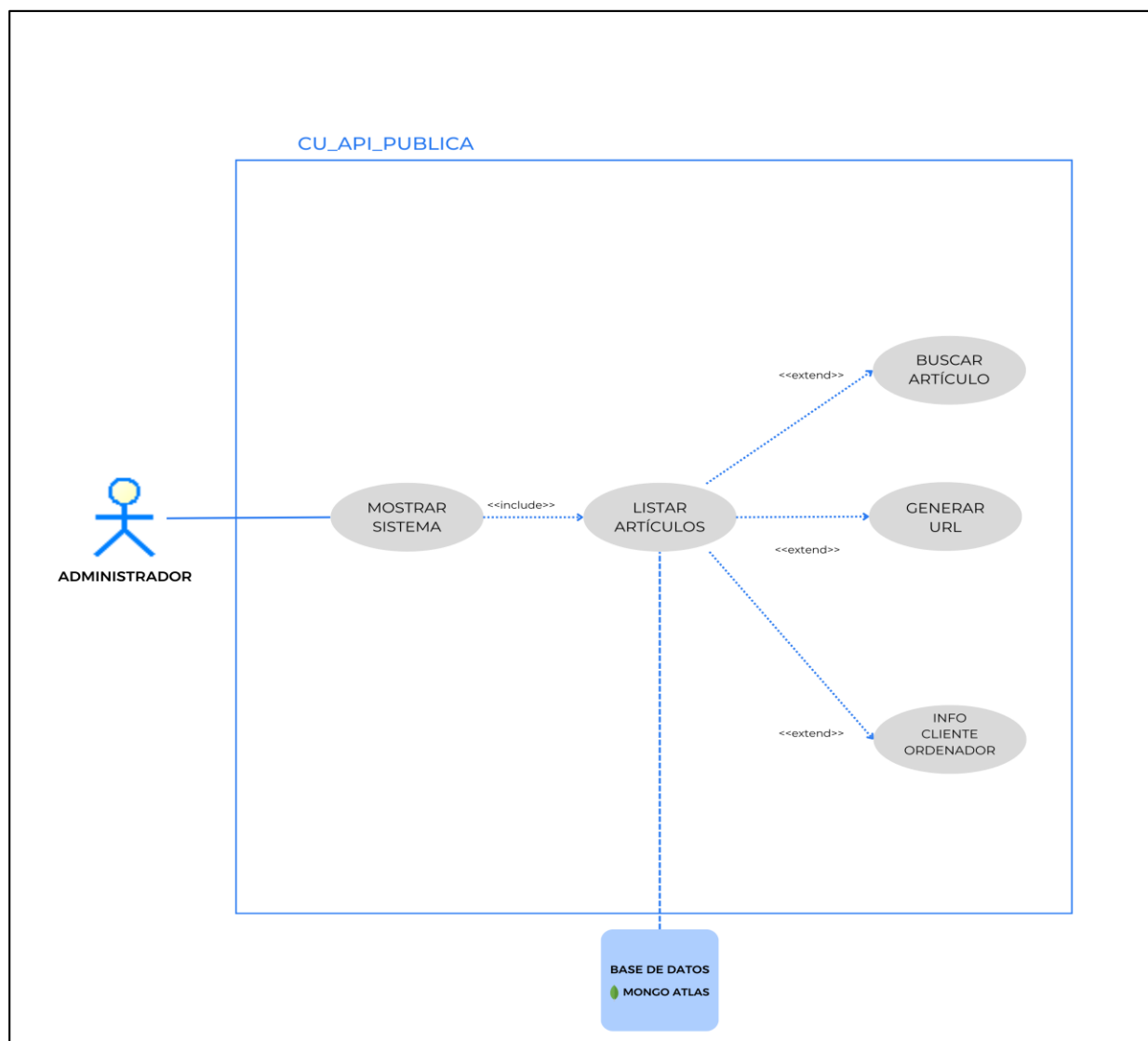
INFORMACION CLIENTE

ID	TÍTULO	AUTORES
10.1371/journal.pone.0287677	Content and form of original research articles in general major medical J	Nicole Heßler, Andreas Ziegler
10.1371/journal.pone.0284462	Developing a comprehensive model of home-based formal care for elder	Khorshid Mobasser, Ahmad Kc
10.1371/journal.pone.0269882	Effect of internet-delivered cognitive behavioral therapy on insomnia in	Yan Chen, Xiang Zhou, Jie Liu, R
10.1371/journal.pone.0263663	Interventions to improve early retention of patients in antiretroviral ther	Samuel Muhula, John Gachohi,
10.1371/journal.pone.0269821	Telehealth in outpatient care for children and adolescents with chronic	Larissa Karoline Dias da Silva
10.1371/journal.pbio.3001260	What's not in the news headlines or titles of Alzheimer disease articles?	Maroia Triunfol, Fabio C. Gouve

Rows per page: 100 1-100 of 1000

Diagramas

CU_API_Pública



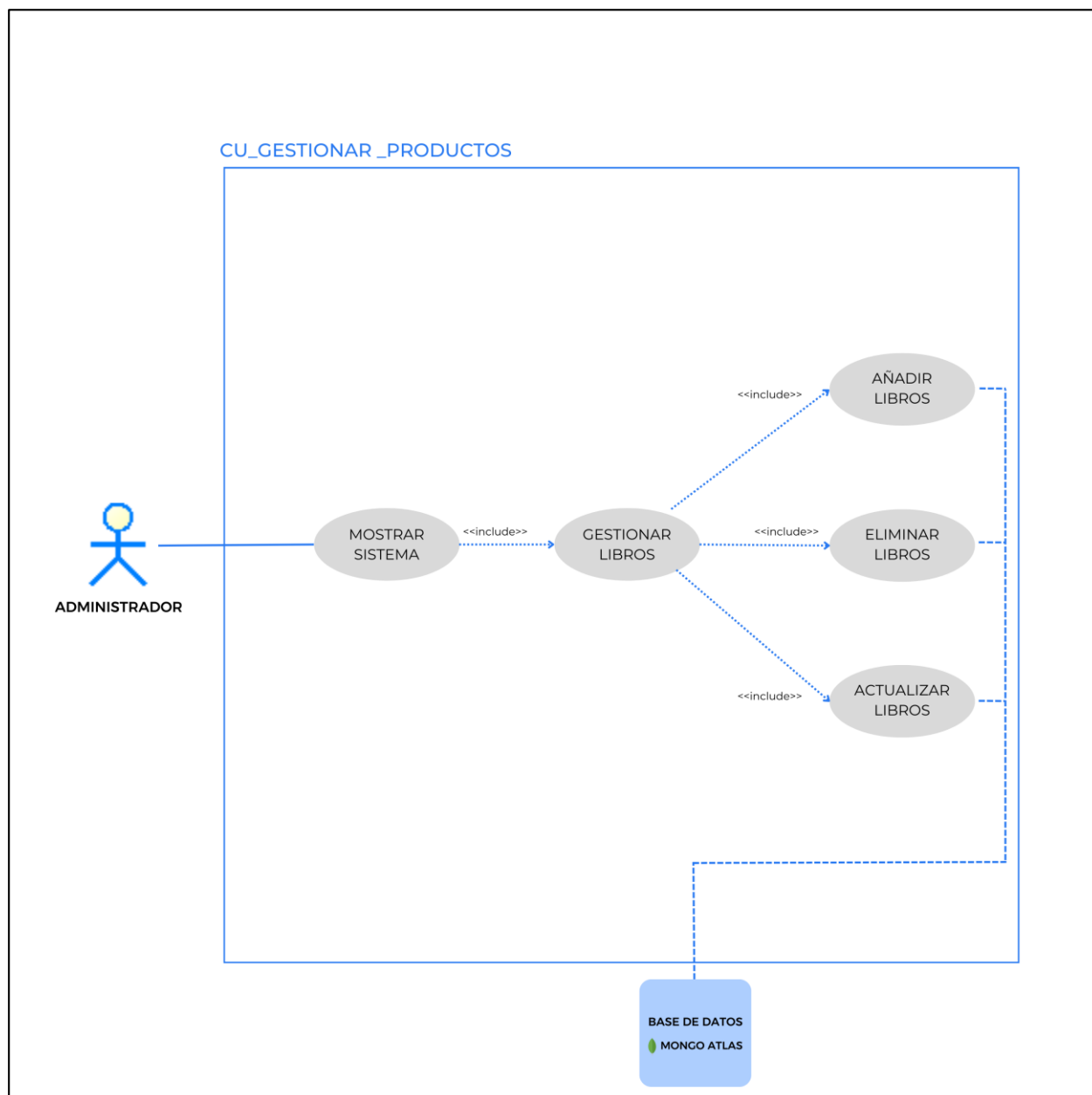
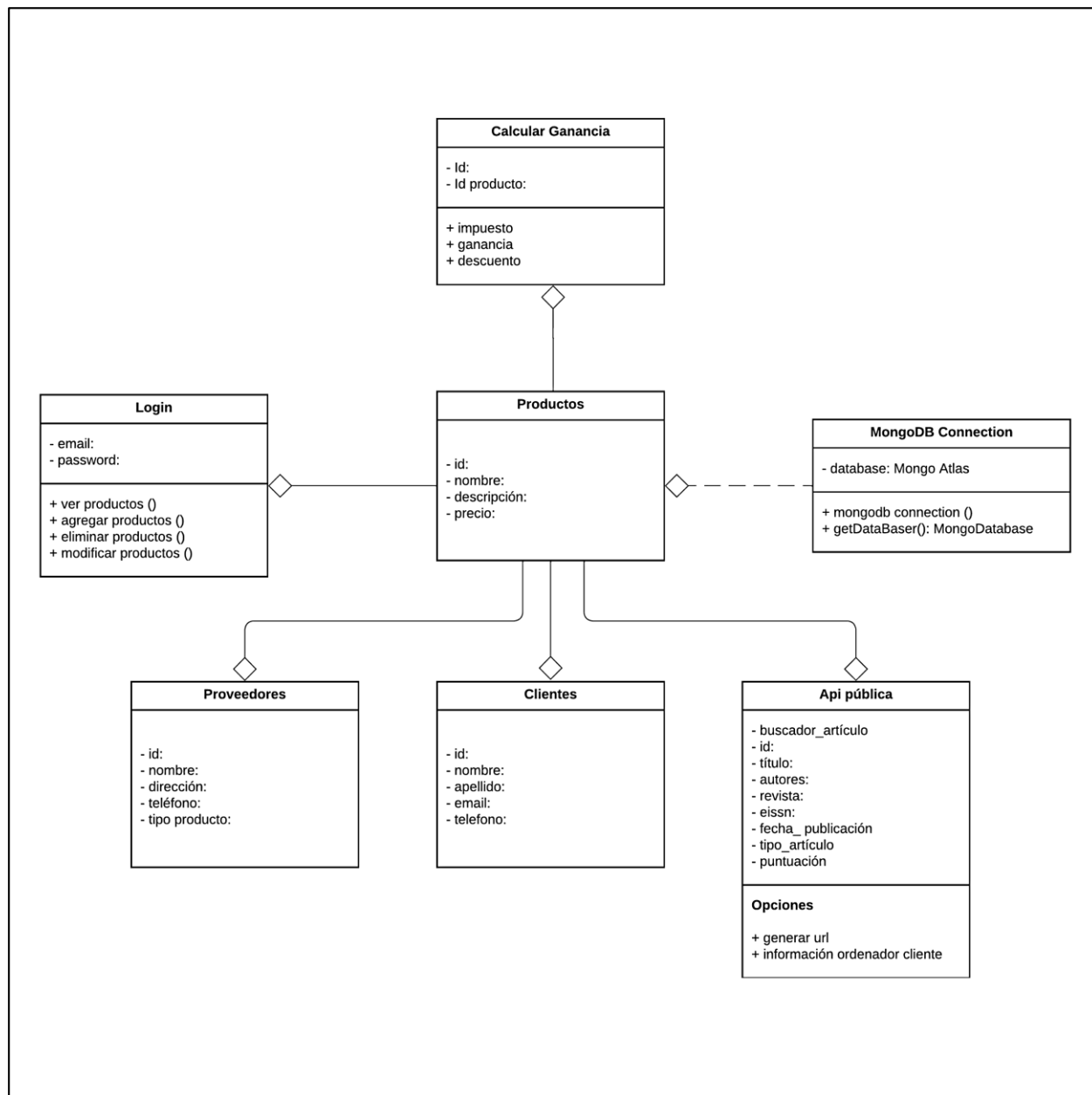
CU_Gestionar_Productos

Diagrama de Clases



Pruebas

Base de datos MongoDB:

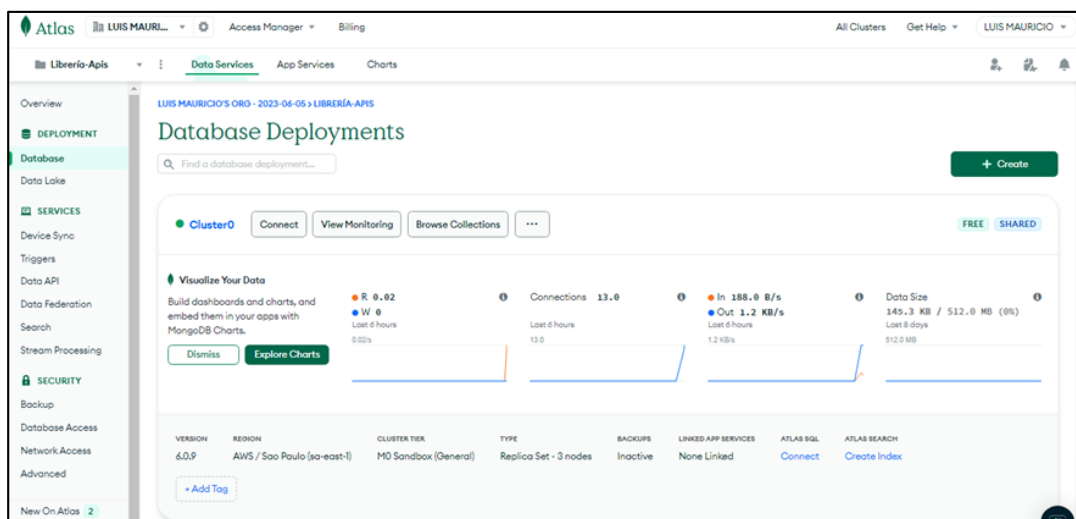


Fig. Base de datos MongoDB

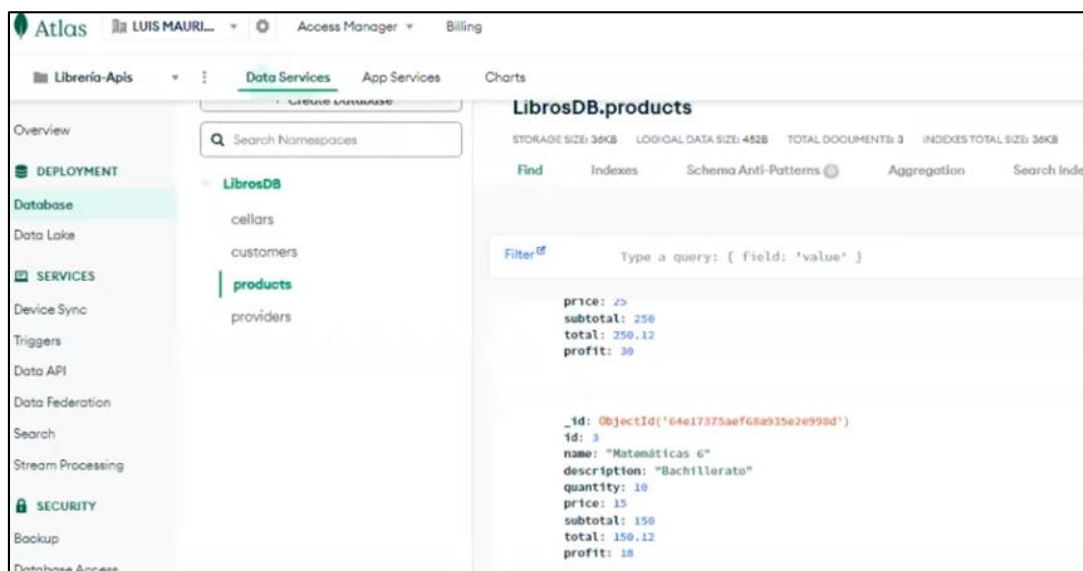


Fig. Base de datos MongoDB 2

Código del Login en react

```

Realizado por Camilo Acosta
import { GoogleLogin } from "react-google-login";
import { gapi } from "gapi-script";
import { useEffect } from "react";
import { useNavigate } from "react-router-dom";
import style from "../css/login.module.css";

const Login = () => {
  const clientId =
    "251742290273-2marja185r0v9n3u0ost29e361ms1243.apps.googleusercontent.com";

  const navigate = useNavigate();
  useEffect(() => {
    const start = () => {
      gapi.auth2.init({ clientId: clientId });
    };
    gapi.load("client:auth2", start);
  }, []);

  const onSuccess = (response) => {
    localStorage.setItem("user", JSON.stringify(response.profileObj));
    console.log(JSON.parse(localStorage.getItem("user")));
    navigate("/");
  };

  const onFailure = (response) => {
    console.log(response);
  };
};

```

Fig. Login Código React

Funcionalidad del login Front End:

Fig. Login Front End

Listar productos interfaz

Lista de Productos					
NOMBRE	DESCRIPCIÓN	PRECIO	CANTIDAD	GANANCIAS	OPCIONES
Matemáticas 5	Bachillerato	18	20	360.12	Editar Eliminar
Electrónica	Circuitos	25	10	250.12	Editar Eliminar
Matemáticas 6	Bachillerato	15	10	150.12	Editar Eliminar
Física 2	Secundaria	18	15	270.12	Editar Eliminar

[Agregar](#)

Fig. Deploy de API en el servicio Google Cloud

URI de productos

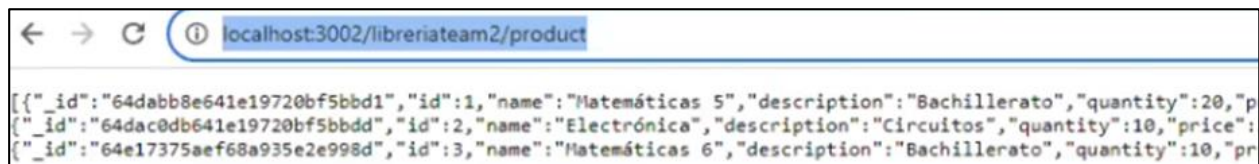


Fig. URI Funcionamiento

Funcionamiento registrar proveedor interfaz gráfica

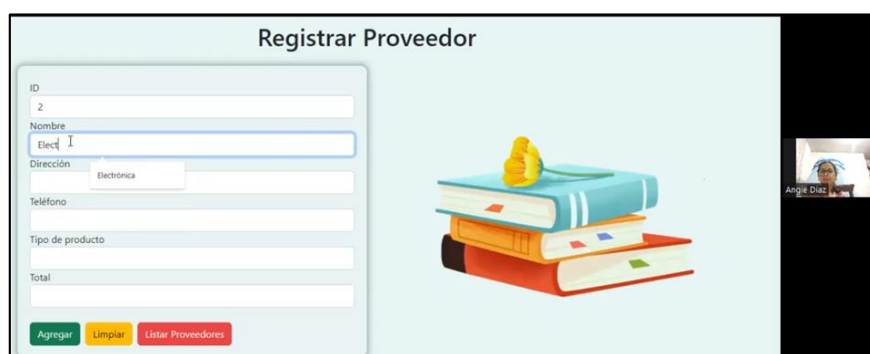


Fig. Funcionamiento de registro de proveedor

Funcionamiento de interfaz Listar Clientes



Fig. Listar Clientes

Funcionamiento del buscador de Artículos mediante API pública



Fig. Funcionamiento de API Pública

Funcionamiento del registro de las especificaciones del computador

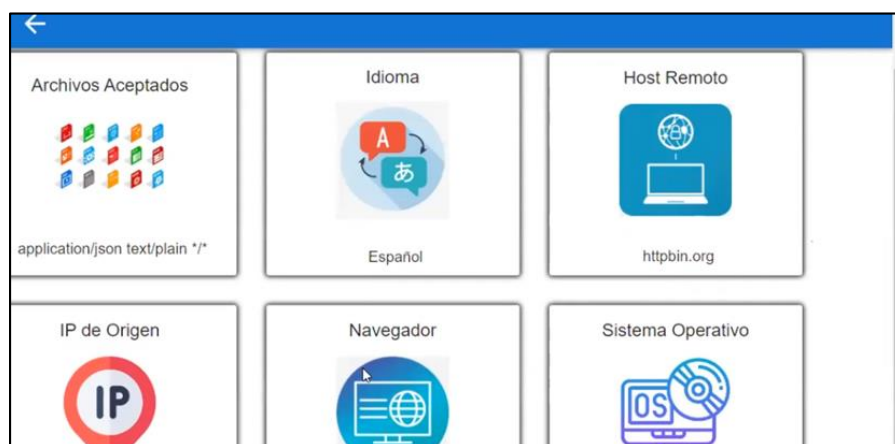


Fig. Funcionamiento de especificaciones del computador usuario

https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/05-Pruebas

Código

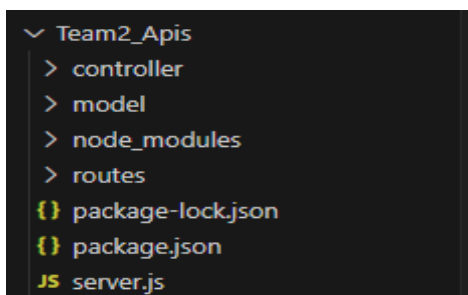


Fig. Carpeta de direccionamiento de las APIs

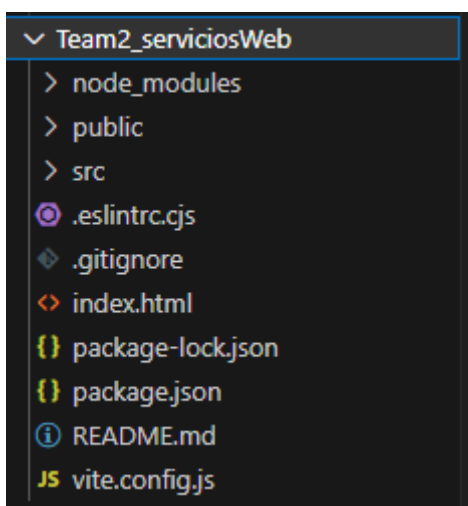


Fig. Carpeta de direccionamiento del Servicio Web

https://github.com/Camilo2600/Team_2_Web_A1_y_A2/tree/main/06-C%C3%B3digo/Act-3

Conclusiones

En el proyecto "Mundos de Papel", se ha logrado implementar con éxito un Front-End web utilizando React y otras tecnologías de desarrollo. A través de las diferentes pestañas y funcionalidades, se ha demostrado la capacidad de crear interfaces de usuario interactivas y funcionales que permiten a los usuarios realizar acciones como iniciar sesión, administrar productos, proveedores y clientes, y consumir información de una API pública. Además, se ha integrado el concepto de sesiones para gestionar la autenticación y el estado del usuario, proporcionando una experiencia personalizada y coherente.

Las pruebas realizadas demuestran el funcionamiento correcto de las diferentes características implementadas, como la autenticación, el CRUD para productos, proveedores y clientes, y la interacción con la API pública. Los diagramas y la documentación proporcionada facilitan la comprensión del flujo de la aplicación y sus componentes, permitiendo una mejor gestión del código y de las futuras actualizaciones.

Recomendaciones

Para garantizar el éxito continuo de la aplicación "Mundos de Papel", se sugiere mantenerse actualizado con las últimas tendencias en desarrollo Front-End, optimizar el rendimiento a medida que la aplicación evoluciona mediante la optimización de llamadas API y la gestión eficiente de recursos, además de priorizar la seguridad mediante la implementación de validación de entrada, protección contra amenazas como ataques XSS y CSRF, y el uso de HTTPS. Asimismo, es crucial asegurar un diseño responsivo para una experiencia óptima en diferentes dispositivos y tamaños de pantalla, fomentar la colaboración constante entre los miembros del equipo para resolver problemas eficazmente, mantener documentación actualizada para facilitar futuros desarrollos y la incorporación de nuevos miembros, llevar a cabo pruebas exhaustivas en diversos navegadores y dispositivos para asegurar un rendimiento uniforme y, finalmente, buscar el feedback de los usuarios para mejorar la experiencia y abordar cualquier problema o sugerencia que surja.

Link del Video

<https://youtu.be/bVy97INTwGg>

Referencias

- Chavarria Campillo, D. S. (2023). Software de retos de programación.
- Paje Alcántara, S. A. (2023). Desarrollo de un REST API y Frontend para una red social.
- Yelmo, J. C., & Fernández-Corugedo, J. (2010). APIS: una herramienta de soporte al aprendizaje colaborativo de la ingeniería del software. In *Congreso Iberoamericano de Informática Educativa, Santiago de Chile*.
- Rojas Barraza, C. J. (2022). Implementación del proceso de pruebas para API REST aplicado a un proyecto de software de una entidad bancaria.
- Martin, C., David, J., Poveda Solano, J. F., & Santiago, J. G. P. Desarrollo del front end del sistema de gestión de clínicas veterinarias aplicando paradigma de Línea de Producto de Software (LPS) enfocado en un desarrollo co-localizado.