

INFORME ENTREGA 1

DANIEL CAMILO BARRERA PEREZ-2205562
JAIDER DANIEL GONZALES ARIZA-2205563

MANUEL GUILLERMO FLOREZ BECERRA

UNIVERSIDAD INDUSTRIAL DE SANTANDER



BUCARAMANGA, SANTANDER
2023

INTRODUCCION

En el siguiente informe veremos los aspectos importantes de la web realizada, como las tecnologías utilizadas y el uso de los conocimientos expuestos en clase.

OBJETIVOS

- Implementar función con EMAILJS.
- Implementar funciones del BACKEND.

MARCO TEORICO

Internet: Red informática de nivel mundial que utiliza la línea telefónica para transmitir la información.

Página web: Una página web, página electrónica, página digital o ciber página es un documento digital complejo, que puede integrar y/o contener texto, sonido, vídeo, programas, enlaces, imágenes, hipervínculos y otros elementos, adaptado para la World Wide Web, y que puede ser accedida y visualizada mediante un navegador web.

HTML: HTML, siglas en inglés de HyperText Markup Language, hace referencia al lenguaje de marcado para la elaboración de páginas web.

CSS: CSS, en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

JavaScript: JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Base de datos: Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS).

GitHub: GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc.

Git: Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

Aerolínea: Una aerolínea, línea aérea o compañía aérea es una empresa que se dedica al transporte de pasajeros o carga —y, en algunos casos, animales— por medio de vehículos aéreos como aviones o helicópteros. El mundo de las líneas aéreas es complejo.

React es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. En el proyecto hay más de mil desarrolladores libres.

Vite es un servidor de desarrollo local escrito por Evan You y utilizado de forma predeterminada por las plantillas de proyecto de Vue. Tiene soporte para TypeScript y JSX.



Desarrollo

El proyecto se elaboró como NAVBAR -DIFERENTES CLASES DE COMPONENTES-
FOOTER El NavBar brinda un ligera búsqueda rápida de los vuelos que es la pagina principal nada más entrar también un apartado de información detallando mas detalles de los sitios que se pueden visitar y un apartado de ayuda que brinda un sistema que envía un correo con el mensaje que el usuario quiere enviar.

VUELOS

La página principal cuenta principalmente con un container de filtrado que se intentó hacer el apartado de filtrado más afondo, pero para eso necesitamos cierta información adicional de las aerolíneas disponibles. Siguiendo en la página principal después del filtrado cuenta con unas cajas que están distribuidas por id la cual facilita su implementación.(Este apartado de cajas la podemos ver en components/data/Data.js) Y por ultimo el footer que imprime y deja unos enlaces de las REDES SOCIALES de Jaider Gonzalez y los designers Jaider Gonzalez y Daniel Barrera

ENRUTAMIENTO

El enrutamiento lo podemos visualizar en la carpeta de Pages el cual tiene incluido 4 rutas exactas y una ruta establecida solamente para rutas que no estén establecidas en dicha página la cual están en jerarquía bien establecidas

AYUDA

En el apartado de ayuda se buscó un sistema de quejas enviadas a las diferentes empresas para así poder establecer factores para su solución. En dicho apartado se utilizó EMAILJS que nos brinda un servicio y un template que es el diseño de dicho correo a enviar que nos facilitó la sincronización de los componentes del correo electrónico para así poderlo enviar a la empresa que este caso será enviado a mi correo electrónico ya que yo soy la empresa.

Esta implementación la podemos ver en la ruta de home/help/Help.jsx

Los aspectos importantes del proyecto son:

- Se realiza una estructura típica de una página y/o aplicación web
- Se implementan tecnologías como react para mejorar la experiencia de la web y facilitar el trabajo a la hora de hacer tareas de la misma



- Se implementa el login para que el usuario no pueda acceder a muchas de las acciones a realizar en la web sin antes haberse registrado en la misma.
- Se diseña también un logo de la empresa creada.
- Se utilizan librerías de react como carousel para implementar la parte de un carrusel de imágenes en el “section”.
- Se hacen también partes importantes de la pagina como “aside” o el “footer” que contiene información general de la web.
- Se implementa la ruta principal (/vuelos) que contiene los complementos de hero y Price que imprimen información de los vuelos y un sistema de filtrado de solo imagen ya que el filtrado de vuelos ocupa un api de pago para los vuelos recientes y que están disponible.
- Para las demás rutas que son (/información y /ayuda) siguió un orden para su funcionamiento que se encuentra en la carpeta de pages de igual manera con ayuda del enrutado cualquier ruta que no este definida será dirigido a una ruta de error
- Para el sistema de ayuda se implemento un servicio gracias EMAILJS que nos brindaba un template y varias keys para el acceso a el Gmail y el envio de este.
- Para la parte del backend, implementamos nodejs con archivos como db-config.js para la conexión con una base de datos documental (mongoDB).

FALTANTE

- El filtrado de vuelos fue imposible su realización debido a las APIS que manejan dichas aerolíneas.
- Se logro conectar a una base de datos pero fue imposible hacer una especie de CRUD gracias a el estilo de proyecto que presentamos con modal y Bootstrap, sin embargo en una versión anterior a ésta si pudimos implementarlo gracias a que lo tenemos con react puro.

CONCLUSIONES

- La implementación de las funciones expuestas en el documento no se logró obtener debido a salidas continuas del equipo de trabajo.
- Se adquirió conocimientos, experiencias en el ámbito profesional gracias al trabajo hecho el AIRSCAPE que buscaba una buena experiencia a las personas en sus reservas de vuelos brindándoles la suficiente información para que su viaje fuera el mejor.

ANEXOS



AIRSCAPE[Vuelos](#)[Información](#)[Ayuda](#)[Login](#)[Register](#)

!Destino, País de nunca Jamás;

El hombre debe elevarse por encima de la Tierra, hasta la cima de la atmósfera y más allá, porque sólo así comprenderá completamente el mundo en el que vive

Origen

Bucaramanga

Destino-Ciudad

Salida

dd/mm/aaaa

Precio

Filtrado

Buscar

Búsqueda

ESP LAA

10:16 21/07



Visual Studio Code interface showing two files: `RutInformation.jsx` and `Pages.jsx`.

File 1: `RutInformation.jsx`

```

1 import React from 'react'
2 import Information from '../information/Information'
3
4
5 const RutInformation = () => {
6   return (
7     <Information/>
8   )
9 }
10
11
12 export default RutInformation

```

File 2: `Pages.jsx`

```

1 import { BrowserRouter as Router, Switch, Route } from 'react-router-dom'
2 import Footer from '../main/footer/footer.jsx'
3 import RutError from '../home/RutError.jsx'
4 import RutInformation from '../home/RutInformation.jsx'
5 import RutHelp from '../home/RutHelp.jsx'
6 import RutRegister from '../home/RutRegister.jsx'
7
8 const Pages = () => {
9   return (
10     <Router>
11       <NavBar/>
12       <Switch>
13         <Route exact path="/" component={Home}/>
14         <Route exact path="/vuelos" component={Home}/>
15         <Route exact path="/informacion" component={RutInformation}/>
16         <Route exact path="/ayuda" component={RutHelp}/>
17         <Route exact path="/register" component={RutRegister}/>
18         <Route exact path="*" component={RutError}/>
19       </Switch>
20       <Footer/>
21     </Router>
22   )
23 }
24
25 export default Pages

```

The terminal shows the command: `PS C:\Users\daniel\Desktop\backend_airscape\ProjectoAIRSCAPE>`



db-config.js - ProyectoAIRSCAPE - Visual Studio Code

```

1  const mongoose = require('mongoose');
2
3  const connectDB = async () => {
4    try {
5      await mongoose.connect('mongodb+srv://Barredaca23:Nico8dani17@cluster0.nsbd9jh.mongodb.net/airscape', {
6        useNewUrlParser: true,
7        useUnifiedTopology: true,
8      });
9    } catch (error) {
10     console.log('conexión exitosa a MongoDB');
11   }
12   console.error('Error al conectar a MongoDB', error);
13   process.exit(1);
14 }
15
16 module.exports = connectDB;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\daniel\Desktop\backend_airscape\ProyectoAIRSCAPE>

index.js - ProyectoAIRSCAPE - Visual Studio Code

```

1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import App from './App';
4  import 'bootstrap/dist/css/bootstrap.min.css'
5  import 'bootstrap/dist/js/bootstrap.bundle'
6
7  const root = ReactDOM.createRoot(document.getElementById('root'));
8  root.render(
9    <React.StrictMode>
10     <App />
11   </React.StrictMode>
12 );
13
14

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\daniel\Desktop\backend_airscape\ProyectoAIRSCAPE>



```
1
2 import './App.css'
3 import Pages from './components/pages/Pages';
4 function App() {
5   return (
6     <<Pages/></>
7   );
8 }
9
10 export default App;
11
```