GRADO FORMATIVO	DAW
CURSO (1º o 2º)	2º
ASIGNATURA	Desarrollo web en entorno cliente
NOMBRE Y APELLIDOS DEL ALUMNO	CARLOS PARRO PÉREZ
DNI	48159548D
FECHA	13/01/2022

HITO INDIVIDUAL 2 - SEGUNDO TRIMESTRE 2021

Objetivos:

- Asimilar los contenidos de la asignatura Desarrollo web en entorno cliente desarrollados en el segundo trimestre 2021
- Aprender los fundamentos más relevantes de React como uno de los principales frameworks frontend del mercado y su aplicación práctica en el desarrollo web.
- Diseño y aplicación de conceptos de desarrollo como componentes, routing, servicios y consumo de datos

Materiales:

- Editores de código como Visual Studio code o similar
- Git / Github
- Google Firebase

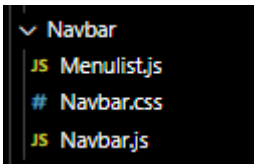
Enunciado: **LEE BIEN EL ENUNCIADO ANTES DE PASAR AL DESARROLLO**

El hito consiste desarrollar una aplicación web con React sobre consumo de datos en API Rest.

Desarrollo: **LEE BIEN LAS CUESTIONES PLANTEADAS ANTES DE RESOLVERLAS. LEE BIEN LO QUE SE PIDE COMO RESPUESTA**

CUESTIÓN 1. *React y diseño*

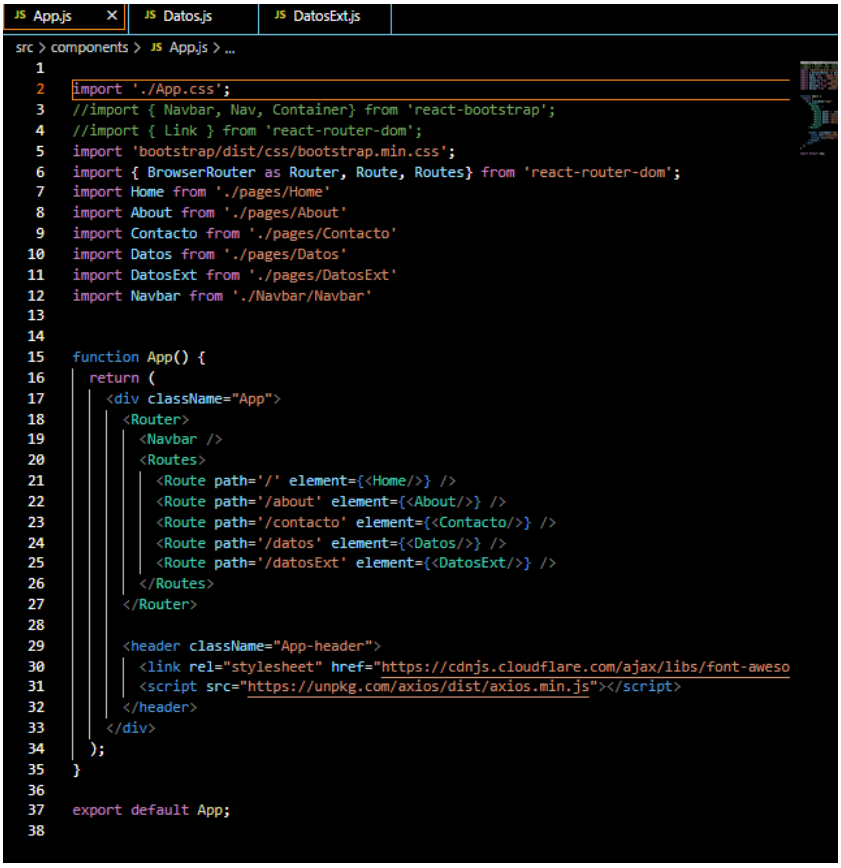
- *Crea un proyecto React en tu equipo local en donde muestres al menos 4 secciones, como por ejemplo, home, about, contacto, datos*




```jsx
import {MenuList} from "./Menulist";
import { useState } from "react";
import "./Navbar.css";
import { NavLink } from "react-router-dom";

const Navbar = () => {
    const [clicked, setClicked] = useState(false);
    const menuList = MenuList.map(({url, title}) => {
        return(
            <li key={title}>
                <NavLink to={url} className={(navData) => (navData.isActive ? "active"
                    {title}
                </NavLink>
            </li>
        );
    });
    const handleClick = () =>{
        setClicked(!clicked);
    }
    return(
        <nav>
            <div className="logo">
                Hito<font>React</font>
            </div>
            <div className="menu-icon" onClick={handleClick}>
                <i className={clicked ? "fa fa-times" : "fa fa-bars"}></i>
            </div>
            <ul className={clicked ? "menu-list" : "menu-list close"}>
                {menuList}
            </ul>
        </nav>
    )
};

export default Navbar;
```



- *Aplica Bootstrap al proyecto con un diseño usable*

Para instalar bootstrap al proyecto React utilizo el siguiente comandoa través del contenedor de paquetes Node.js.

```
npm install bootstrap
```

- *Aplica routing para pasar de una sección (página) a otra.*

Para aplicar el Routing he creado un componente que incluye tres archivos, un archivo JavaScript el cual contiene la configuración del NavBar aplicándole el estilo a través de la importación del archivo css, y por último, contiene un archivo MenuList JavaScript que contiene la configuración de los links de nuestra web.

***RESPUESTA CUESTION 1**: Proyecto React*

CUESTIÓN 2. *Consumo de datos*

- *Muestra una serie de datos utilizando json desde una propiedad local*

Llamamos a la API local mediante un mapeo de datos mostrando los datos en orden alfabético.

App.js | Datos.js | DatosExt.js

src > components > pages > Datos.js > Datos > map() callback

```
import Data from '../data/data.json';

const Datos = () => {
    return(
        <div className="page-heading">
            <div className='jsondata'>
                <table class="table text-white">
                    <thead>
                        <tr>
                        <th scope="col">#</th>
                        <th scope="col">Nombre</th>
                        <th scope="col">Apellido</th>
                        <th scope="col">Edad</th>
                        <th scope="col">Año Nacimiento</th>
                        </tr>
                    </thead>
                {Data.sort((a, b) => a.nombre.localeCompare(b.nombre)).map(data =>{
                    return(
                        <>

                        <tbody>
                            <tr>
                            <th scope="row" key={data.id}/>
                                <td>{data.nombre}</td>
                                <td>{data.apellido1}</td>
                                <td>{data.edad}</td>
                                <td>{data.anio_nac}</td>
                            </tr>
                        </tbody>

                        </>
                    )
                })}
                </table>
            </div>
        </div>
    )
};

export default Datos;
```

- *Muestra una serie de datos utilizando json desde una petición remota*

  Mostramos datos a través de la conexión con una API externa la cual, a través de un fetch y la función useEffect (la cual nos garantiza que vamos a recoger los datos a través de un array de datos). Luego realizamos el mapeo de datos si el fetch ha resultado satisfactorio.

- *Muestra un ejemplo con un service a tu elección*

  En este ejemplo, vemos como se crea una clase que será nuestra Service la cual nos servirá para alojar las funciones que utilizaremos para conectarnos con nuestra fuente de datos y recoger los mismos en nuestra aplicación.

```
class ItemService {
 constructor() {
  this.items = [
   {link:1, name:"test1", summary:"Summary Test 1", year:"2001", country:"us", price:"1000", description:"Desc 1"},
   {link:2, name:"test2", summary:"Summary Test 2", year:"2002", country:"uk", price:"2000", description:"Desc 2"},
   {link:3, name:"test3", summary:"Summary Test 3", year:"2003", country:"cz", price:"3000", description:"Desc 3"},
  ];
 }
 async retrieveItems() {
   return Promise.resolve(this.items);
 }
 async getItem(itemLink) {
  for(var i = 0; i < this.items.length; i++) {
   if ( this.items[i].link === itemLink) {
    return Promise.resolve(this.items[i]);
   }
  }
  return null;
 }
 async createItem(item) {
  console.log("ItemService.createItem():");
  console.log(item);
  return Promise.resolve(item);
 }
 async deleteItem(itemId) {
  console.log("ItemService.deleteItem():");
  console.log("item ID:" + itemId);
 }
 async updateItem(item) {
  console.log("ItemService.updateItem():");
  console.log(item);
 }
}
export default ItemService;
```

En nuestro archivo App.js llamamos a nuestro Service y hacemos uso de las funciones alojadas en ella para realizar las acciones pertinentes en nuestra aplicación.

```
import React, { Component } from 'react';
import './App.css';
import ItemDetails from './item-details';
import NewItem from './new-item';
import EditItem from './edit-item';
import ItemService from './shared/mock-item-service';
class App extends Component {
 constructor(props) {
  super(props);
  this.itemService = new ItemService();
  this.onSelect = this.onSelect.bind(this);
    this.onNewItem = this.onNewItem.bind(this);
    this.onEditItem = this.onEditItem.bind(this);
    this.onCancel = this.onCancel.bind(this);
    this.onCancelEdit = this.onCancelEdit.bind(this);
    this.onCreateItem = this.onCreateItem.bind(this);
    this.onUpdateItem = this.onUpdateItem.bind(this);
    this.onDeleteItem = this.onDeleteItem.bind(this);
    this.state = {
     showDetails: false,
     editItem: false,
     selectedItem: null,
     newItem: null
    }
  }
  componentDidMount() {
    this.getItems();
  }
  render() {
   const items = this.state.items;
   if(!items) return null;
   const showDetails = this.state.showDetails;
   const selectedItem = this.state.selectedItem;
   const newItem = this.state.newItem;
   const editItem = this.state.editItem;
   const listItems = items.map((item) =>
    <li key={item.link} onClick={() => this.onSelect(item.link)}>
      <span className="item-name">{item.name}</span> |  {item.summary}
    </li>    );
```

```
  return (
    <div className="App">
      <ul className="items">
       {listItems}
      </ul>
      <br/>
      <button type="button" name="button" onClick={() => this.onNewItem()}>New Item</button>
      <br/>
       {newItem && <NewItem onSubmit={this.onCreateItem} onCancel={this.onCancel}></NewItem>}
       {showDetails && selectedItem && <ItemDetails item={selectedItem} onEdit={this.onEditItem}  onDelete={this.onDeleteItem} ></ItemDetails>}
       {editItem && selectedItem && <EditItem onSubmit={this.onUpdateItem} onCancel={this.onCancelEdit} item={selectedItem} ></EditItem>}
    </div>
  );
 }
 getItems() {
  this.itemService.retrieveItems().then(items => {
      this.setState({items: items});
    }
  );
 }
 onSelect(itemLink) {
  this.clearState();
  this.itemService.getItem(itemLink).then(item => {
    this.setState({
      showDetails: true,
      selectedItem: item
     });
    }
  );
 }

 onCancel() {
  this.clearState();
 }
 onNewItem() {
  this.clearState();
  this.setState({
    newItem: true
  });
```

```
}

onEditItem() {
  this.setState({
    showDetails: false,
    editItem: true,
    newItem: null
  });
}

onCancelEdit() {
  this.setState({
    showDetails: true,
    editItem: false,
    newItem: null
  });
}
onUpdateItem(item) {
  this.clearState();
  this.itemService.updateItem(item).then(item => {
      this.getItems();
    }
  );
}
onCreateItem(newItem) {
  this.clearState();
  this.itemService.createItem(newItem).then(item => {
      this.getItems();
    }
  );
}
onDeleteItem(itemLink) {
  this.clearState();
  this.itemService.deleteItem(itemLink).then(res => {
      this.getItems();
    }
  );
}

clearState() {
  this.setState({
    showDetails: false,
    selectedItem: null,
    editItem: false,
```

```
      newItem: null
    });
  }
}
export default App;
```

***RESPUESTA CUESTION 2.1.**: Proyecto React*

CUESTIÓN 3. *Publicación y despliegue*

- *Publica tu proyecto React en Google Firebase*

Para subir nuestro proyecto de React vamos a seguir los siguientes pasos:

1. Creamos el proyecto en firebase
2. Abrimos terminal de node.js
3. Nos loggueamos a firebase mediante firebase login
4. Inicializamos el proyecto a subir a hosting de firebase mediante firebase init
    a. Single page up yes
    b. Carpeta dist
    c. Github no
5. Compilamos el archivo que subiremos a producción con npm run build
6. Borramos la carpeta dist
7. En el archivo firebase.json de nuestro proyecto cambiamos el atributo public de hosting a "build"
8. Y en la terminal de Node.js ejecutamos el deploy de nuestra aplicación.

***RESPUESTA CUESTION 3.1.**: Enlace del proyecto React en Google Firebase*

Firma del alumno:

Rúbrica:

PUNTUACIÓN MÍNIMA PARA CONSIDERAR LA PRUEBA SUPERADA: 5 PUNTOS

CUESTION 1 **React y diseño**	El proyecto de React no funciona	El proyecto de React funciona correctamente	El proyecto funciona y además incorpora elementos visuales y de diseño con un atractivo impactante que permite disponer de un sitio accesible y usable
	0 puntos	**2 puntos**	**3 puntos**
CUESTION 2 **Consumo de datos**	El proyecto no consume los datos correctamente	El proyecto consume los datos correctamente	El proyecto consume los datos y los muestra de una manera atractiva y funcional
	0 puntos	**2 puntos**	**3 puntos**
CUESTION 3 **Firebase**	El proyecto no funciona correctamente en Firebase	El proyecto funciona correctamente en Firebase	El proyecto funciona en Firebase y su publicación y optimización en el hosting son adecuadas
	0 puntos	**2 puntos**	**3 puntos**
DOCUMENTACION APORTADA	Documento PDF con explicación de los pasos realizados		
	1 punto		