

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	31/01/2022

HITO GRUPAL 1 - 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 diferentes frameworks web a la hora de diseñar aplicaciones híbridas utilizando Ionic.
- 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

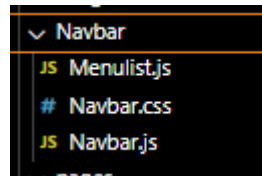
El hito consiste desarrollar una aplicación web con tu framework favorito sobre consumo de datos en API Rest y generar una app híbrida.

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

CUESTIÓN 1. Framework

- Crea un proyecto web con tu framework preferido (Angular / Vue / React) que contenga las características de routing, consumo de datos, gestión de cookies y seguridad de la app.

Routing



```
Navbar.js X
src > components > Navbar > JS Navbar.js > ...
1 import {MenuList} from "../MenuList";
2 import { useState } from "react";
3 import "../Navbar.css";
4 import { NavLink } from "react-router-dom";
5
6 const Navbar = () => {
7   const [clicked, setClicked] = useState(false);
8   const menuList = MenuList.map(({url, title}) => {
9     return(
10       <li key={title}>
11         <NavLink to={url} className={navData => (navData.isActive ? "active"
12           {title}
13         </NavLink>
14       </li>
15     );
16   });
17   const handleClick = () =>{
18     setClicked(!clicked);
19   }
20   return(
21     <nav>
22       <div className="logo">
23         HitoGrupal<font>React</font>
24       </div>
25       <div className="menu-icon" onClick={handleClick}>
26         <i className={clicked ? "fa fa-times" : "fa fa-bars"}></i>
27       </div>
28       <ul className={clicked ? "menu-list" : "menu-list close"}>
29         {menuList}
30       </ul>
31     </nav>
32   )
33 };
34
35 export default Navbar;
```

```

JS Menulist.js x
src > components > Navbar > JS Menulist.js > ...
1  export const MenuList = [
2      {
3          title: "Home",
4          url: "/"
5      },
6      {
7          title: "About",
8          url: "/about"
9      },
10     {
11         title: "Contacto",
12         url: "/contacto"
13     },
14     {
15         title: "Datos",
16         url: "/datos"
17     },
18     {
19         title: "DatosExt",
20         url: "/datosExt"
21     },
22 ]

```

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.

Consumo de datos

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

```

1  import Data from '../data/data.json';
2
3
4  const Datos = () => {
5    return(
6      <div className="page-heading">
7        <div className='jsondata'>
8          <table class="table text-white">
9            <thead>
10             <tr>
11               <th scope="col">#</th>
12               <th scope="col">Nombre</th>
13               <th scope="col">Apellido</th>
14               <th scope="col">Edad</th>
15               <th scope="col">Año Nacimiento</th>
16             </tr>
17           </thead>
18           <tbody>
19             <tr>
20               <th scope="row" key={data.id}>
21                 <td>{data.nombre}</td>
22                 <td>{data.apellido}</td>
23                 <td>{data.edad}</td>
24                 <td>{data.anio_nac}</td>
25               </tr>
26             </tbody>
27           </table>
28         </div>
29       </div>
30     );
31   };
32
33   export default Datos;

```

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.

```
App.js | JS Datos.js | JS DatosExt.js x
c > components > pages > JS DatosExt.js > ...
1
2 import React, {useEffect, useState} from 'react';
3 //import axios from 'axios';
4
5 const DatosExt = () => {
6
7   const [products, setProducts] = useState([]);
8
9   //componentDidMount
10  useEffect(()=>{
11    fetch("https://61e5a6bdc14c7a0017124dc9.mockapi.io/Tecnologies")
12      .then(res => res.json())
13      .then(res => {
14        | setProducts(res);
15      })
16      .catch(e=>{
17        | console.log(e);
18      })
19    });
20
21  return (
22    <>
23    <div id="prods" className="DatosExt">
24      <ul>
25        {products.map(product =>
26          <li class="products" key={product.id}>
27            | {product.id} / {product.nombre} / {product.creacion} / {product.operativ
28          </li>
29        )}
30      </ul>
31    </div>
32    </>
33  );
34  }
35
36  export default DatosExt;
```

Cookies

Establecemos las cookies en las que verificamos que el usuario acepta la política de cookies, guardando la verificación mediante un Session Storage al darle al botón pertinente.

```
const cookieStorage = {
  getItem: (key) => {
    const cookies = document.cookie
      .split(';')
      .map(cookie => cookie.split('='))
      .reduce((acc, [key, value]) => ({...acc, [key.trim()]: value}), {});
    return cookies[key];
  },
  setItem: (key, value) => {
    document.cookie = `${key}=${value}`
  },
};

const storageType = cookieStorage;
const consentPropertyName = "jdc_consent";

const shouldShowPopup = () => !storageType.getItem(consentPropertyName);
const saveToStorage = () => storageType.setItem(consentPropertyName, true);

window.onload = () => {

  const consentPopup = document.getElementById('consent-popup');
  const acceptBtn = document.getElementById('accept');

  const acceptFn = event => {
    saveToStorage(storageType);
    consentPopup.classList.add('hidden');
  }

  acceptBtn.addEventListener('click', acceptFn);

  if(shouldShowPopup(storageType)){
    /*const consent = confirm('¿Está de acuerdo con los términos y condiciones de la
    if(consent){
      saveToStorage();
    }*/
    setTimeout(() => {
      consentPopup.classList.remove('hidden');
    }, 2000);
  }
}
```

```
<div id="consent-popup" class="hidden">
  <p>Usando este sitio aceptas nuestros <a href="#"> Términos y Condiciones</a>.
  Por favor <a id="accept" href="#">Acepte</a> esto antes de usar nuestro siti
</p>
</div>
```

Seguridad (Auth)

Establecemos la seguridad en la aplicación gracias al sistema de registro e inicio de sesión de usuarios al sistema para poder utilizar la aplicación.

Login

```

import { makeStyles } from '@material-ui/core';
import { signInWithEmailAndPassword } from 'firebase/auth';
import React from 'react';
import { Form, Button } from 'react-bootstrap';
import { auth } from '../firebase';

export const Login = () => {
  const clases = usarEstilos();
  return (
    <div className={clases.container}>
      <br/></br>
      <h2>Iniciar sesión</h2>
      <div>
        <Form.Group className="mb-3" controlId="formBasicEmail">
          <Form.Label>Correo electrónico</Form.Label>
          <Form.Control id="correo" type="email" placeholder="Introduce corr
            <Form.Text className="text-muted">
              Su información es confidencial. Ningún tercero tendrá acceso a
            </Form.Text>
        </Form.Group>

        <Form.Group className="mb-3" controlId="formBasicPassword">
          <Form.Label>Contraseña</Form.Label>
          <Form.Control id="passwd" type="password" placeholder="Contraseña"
        </Form.Group>

        <Button variant="primary" onClick={login}>
          Iniciar sesión
        </Button>
      </div><br/></br>
      <a href="/registro">¿No tiene cuenta? Regístrese</a>
    </div>
  )
}

function login() {
  signInWithEmailAndPassword(auth, document.getElementById('correo').value, docu
    .then(() => {
      sessionStorage.setItem('user', document.getElementById('correo').value);
      alert('Bienvenido ' + document.getElementById('correo').value);
      window.location.href = '/';
    })
    .catch(() => {
      alert('Error al iniciar sesión');
    })
  );
}

const usarEstilos = makeStyles(() => ({
  container: {
    maxWidth: '80vh',
    maxHeight: '80vh',
    margin: '0 auto',
    padding: '15px'
  }
}));
  
```

Registro

```
App.js JS Registro.js X
> components > JS Registro.js > [?] usarEstilos > [?] makeStyles() callback
1 import { makeStyles } from '@material-ui/core';
2 import { createUserWithEmailAndPassword } from 'firebase/auth';
3 import { doc, setDoc } from 'firebase/firestore';
4 import React from 'react';
5 import { Form, Button } from 'react-bootstrap';
6 import { auth, db } from '../firebase';
7
8 export const Registro = () => {
9   const clases = usarEstilos();
10  return (
11    <div className={clases.container}>
12      <br/><br/>
13      <h2>Registrarse</h2>
14      <div>
15        <Form.Group className="mb-3" controlId="formBasicEmail">
16          <Form.Label>Correo electrónico</Form.Label>
17          <Form.Control id="correo" type="email" placeholder="Introduce correo electrónico" />
18          <Form.Text className="text-muted">
19            Su información es personal y nadie tiene acceso a ella.
20          </Form.Text>
21        </Form.Group>
22
23        <Form.Group className="mb-3" controlId="formBasicPassword">
24          <Form.Label>Contraseña</Form.Label>
25          <Form.Control id="pass" type="password" placeholder="Contraseña" />
26        </Form.Group>
27
28        <Button variant="primary" onClick={login}>
29          Registrarse
30        </Button>
31      </div><br/><br/>
32      <a href="/login">¿Ya tienes cuenta? Inicia sesión</a>
33    </div>
34  )
35
36  async function login() {
37    const cardb = doc(db, "usuarios", document.getElementById('correo').value);
38
39    await setDoc(cardb, {
40      registro: 'true'
41    })
42      .then((e) => {
43        createUserWithEmailAndPassword(auth, document.getElementById('correo').value, document.getElementById('pass').value)
44          .then(() => {
45            console.log('esto si ');
46            sessionStorage.setItem('user', document.getElementById('correo').value);
47            console.log('GUARDO');
48            alert('Registrado correctamente');
49            window.location.href = '/';
50          })
51          .catch((err) => {
52            console.log(err);
53            alert('Error al registrar');
54          });
55      })
56  }
57
58  return (
59    <div>
60      <h1>Registro</h1>
61    </div>
62  )
63 }
```


App.js

```

return (
  <div>
    {
      sessionStorage.length !== 0 ? (
        <div className="App">
          <Router>
            <Navbar />
            <Routes>
              <Route path="/" element={<Home/>} />
              <Route path="/about" element={<About/>} />
              <Route path="/contacto" element={<Contacto/>} />
              <Route path="/datos" element={<Datos/>} />
              <Route path="/datosExt" element={<DatosExt/>} />
            </Routes>
          </Router>

          <div id="consent-popup" class="hidden">
            <p>Usando este sitio aceptas nuestros <a href="#"> Términos y Condiciones</a>  

            Por favor <a id="accept" href="#">Acepte</a> esto antes de usar nuestro  

          </p>
          </div>

          <header className="App-header">
            <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-a</
            <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
          </header>
        </div>) : (
          <div className="App">
            <Router>
              <Navbar />
              <Routes>
                <Route path="/" element={<Registro/>} />
                <Route path="/registro" element={<Registro/>} />
                <Route path="/login" element={<Login/>} />
              </Routes>
            </Router>

            <div id="consent-popup" class="hidden">
              <p>Usando este sitio aceptas nuestros <a href="#"> Términos y Condiciones</
              Por favor <a id="accept" href="#">Acepte</a> esto antes de usar nuestro s</
            </p>
            </div>

            <header className="App-header">
              <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-aw</
              <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
            </header>
          </div>)
        }
      </div>
    );
  export default App;

```

- Aplica Bootstrap al proyecto con un diseño usable

```
package.json X
() package.json > ...
1 {
2   "name": "hito_grupal_carlos_celia",
3   "version": "0.1.0",
4   "private": true,
5   "dependencies": {
6     "@capacitor/camera": "^1.3.1",
7     "@capacitor/filesystem": "^1.1.0",
8     "@capacitor/storage": "^1.2.5",
9     "@ionic/react": "^6.0.10",
10    "@testing-library/jest-dom": "^5.16.2",
11    "@testing-library/react": "^12.1.3",
12    "@testing-library/user-event": "^13.5.0",
13    "@types/react-router": "^5.1.18",
14    "@types/react-router-dom": "^5.3.3",
15    "axios": "^0.26.0",
16    "bootstrap": "^5.1.3",
17    "jquery": "^3.6.0",
18    "popper.js": "^1.16.1",
19    "react": "^17.0.2",
20    "react-dom": "^17.0.2",
21    "react-router": "^6.2.2",
22    "react-router-dom": "^6.2.2",
23    "react-scripts": "5.0.0",
24    "web-vitals": "^2.1.4"
25  },
}
```

```
import "../App.css";
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>
          <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>
              </th>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  );
};

export default Datos;
```

RESPUESTA CUESTION 1: Proyecto framework

Hemos convertido nuestra App en una APK de instalación gracias a la ayuda de Ionic.

1. Primeramente, hemos instalado el cliente de Ionic mediante el siguiente comando en Node.js.

```
npm install -g @ionic/cli
```

2. Seguidamente, hemos instalado Capacitor, que es en pocas palabras, un framework que permite "transformar" una WebApp en una App Mobile o Desktop mediante el uso de WebView. Es el framework por excelencia de Ionic.

```
npm install @capacitor/core @capacitor/cli
```

3. Iniciamos Capacitor en nuestro proyecto.

```
npx cap init
```

Y rellenamos los datos pertinentes.

4. Instalamos Capacitor Android CLI

```
npm install @capacitor/android
```

5. Compilamos nuestros archivos

```
ionic init  
ionic build
```

6. Añadimos los ficheros de configuración de Android

```
npx cap add android
```

Se creará una carpeta llamada android. Lo abrimos a través de Android Studio e iniciamos en nuestro simulador.

7. Generamos el apk, pestaña de Android Studio Build > Generate Signed Bundle / APK
Para generar una reléase apk versión, necesitamos tener una key store, creándola o usando una existente.

8. Elegimos reléase y chequeamos V2 (Full APK Signature)

9. En app > realease se generará la APK

CUESTIÓN 2. *Publicación y despliegue*

- *Publica tu proyecto en Firebase y explica los pasos.*

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 loggreamos 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 2.1.: *Enlace del proyecto en Google Firebase*

Firma del alumno:

Rúbrica:

PUNTUACIÓN MÍNIMA PARA CONSIDERAR LA PRUEBA SUPERADA: 5 PUNTOS

CUESTION 1 Framework	El proyecto no funciona	El proyecto 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	3 puntos	4 puntos
CUESTION 2 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	3 puntos	4 puntos
DOCUMENTACION APORTADA	Documento PDF con explicación de los pasos realizados		
	2 punto		