

***Pràctica Obligatoria per a DA02***

***AITOR BAUZÁ GÓMEZ - 25/26***

***Desenvolupament al Nùvol***

***Carlos Canals Gozávez - CIFP Pau Casesnoves***

# ÍNDEX

1. INTRODUCCIÓ.....	3
2. METODOLOGÍA.....	3
3. Pasos previs per poder dur a terme la tasca.....	4
3.1 Instal·lar Node.js.....	4
3.2 Plugins de VS Code necessaris.....	5
3.3 Creació del projecte React (Vite).....	6
3.4 Neteja inicial.....	6
3.5 Inicializam Git.....	7
4. Crear projecte en Firebase.....	8
4.1 Entrar en Firebase.....	8
4.2 Creació de la app web en Firebase.....	8
4.3 Configurar Firebase Auth.....	9
4.4 Crear Firestore.....	10
4.5 Instal·lar Firebase en React.....	10
4.6 Variables d'entorn.....	10
4.7 Configuració de Firebase en el projecte.....	11
4.8 Probar connexió amb Firebase.....	12
5. Autenticació d'usuaris.....	12
5.1 Instal·lar React Router.....	12
5.2 Creació de carpetes i arxius.....	13
5.3 Proves del funcionament.....	16
6. Gestió de projectes.....	17
6.1 Modificació i creació d'arxius.....	17
6.2 Proves del funcionament.....	18
7. Gestió de despeses.....	20
7.1 Modificació i creació d'arxius.....	20
7.2 Proves del funcionament.....	20
8. Gestió de participants.....	23
8.1 Modificació i creació d'arxius.....	23
8.2 Proves del funcionament.....	23
9. Afegir un panell de resum que mostri el total de despeses per projecte i la quantitat que cada participant deu o ha de rebre segons la divisió de les despeses.....	25
9.1 Modificació i creació d'arxius.....	25
9.2 Proves del funcionament.....	25
10. CONCLUSIÓ.....	27
11. BIBLIOGRAFIA.....	27
12. REPOSITORI GITHUB.....	27

## 1. INTRODUCCIÓ

Aquesta tasca es divideix en dos parts:

- Fer un frontal web amb React (en aquest cas JSX), emprant funcionalitats com React Router i hooks propis de React, a més de crear components amb la finalitat de dur a terme un frontal encarregat de mostrar, afegir, editar i eliminar projectes, participants i despeses dins dels projectes, a més d'un panell informatiu indicant les despeses per projecte.
- Construir un backend, amb tota la base que t'ofereix Firebase des de l'autenticació via email/contrasenya fins una base de dades NoSQL com es FireStore per emmagatzemar les dades, facilitant la gestió del back.

Mitjançant un *Api\_Key* el front es podrà connectar amb el back.

## 2. METODOLOGIA

Per dur a terme aquesta pràctica, he emprat:

- VS Code pel front amb React.
- Funcionalitats de React com React Router per gestionar l'enrutament.
- Firebase com a backend, amb totes les facilitats que t'ofereix com Firestore Auth o com FireStore, la seva base de dades NoSql que, amb clics, ets capaç de gestionar.
- La documentació d'aquest mòdul, proporcionada pel professor.
- Chat GPT alhora de corregir errades que no era capaç de veure a nivell de codi.
- Documentació oficial, citades en la bibliografia.

### 3. Pasos previs per poder dur a terme la tasca

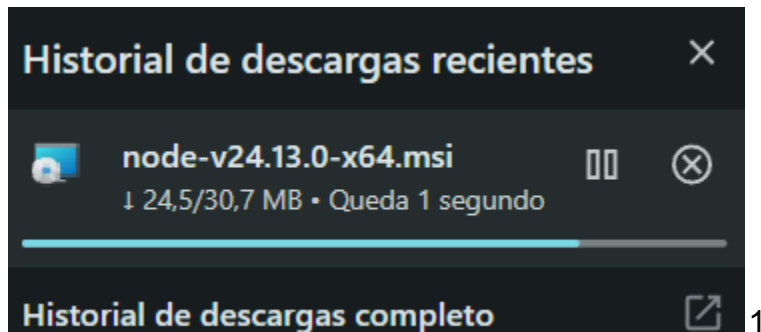
Hi ha passos que no estaran documentats perquè, en el meu cas, no seran necessaris, és a dir per exemple, jo ja tinc VS Code instal·lat, docs no el documentaré.

#### 3.1 Instal·lar Node.js

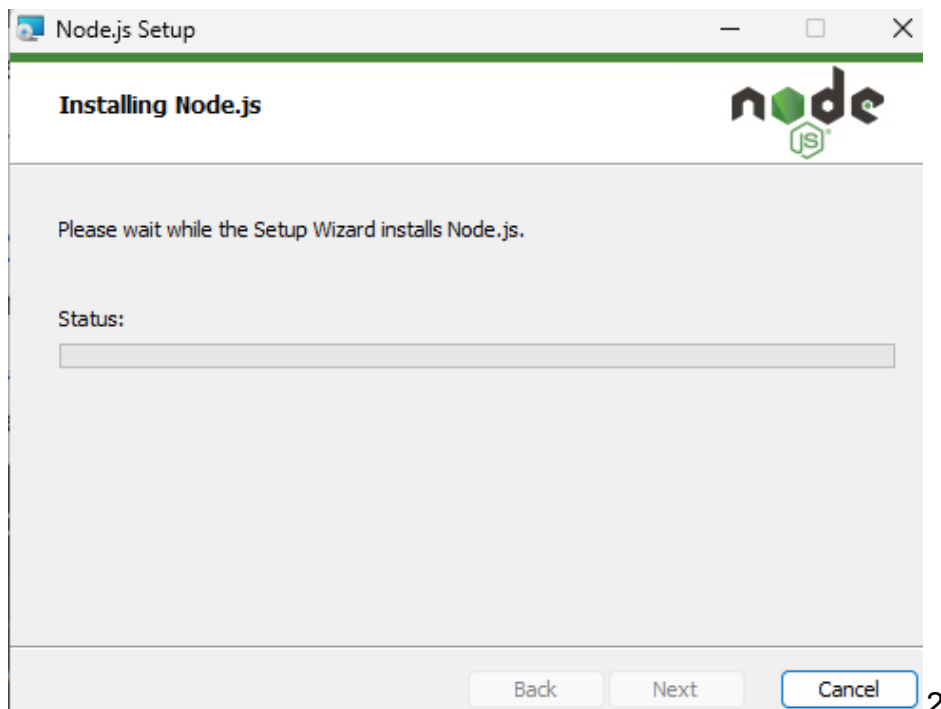
- Per instal·lar-lo, hauràs d'anar al següent enllaç:

<https://nodejs.org/>

Dins d'ell, selecciona la versió LTS, deixa tot per defecte i instal·la-t'ho.



*(procés de descàrrega de Node.js)*



*(procés d'instal·lació de Node.js)*

- Podem comprovar amb la següent comanda que Node s'ha instal·lat correctament:

`node -v`

`npm -v`

Si apareix algo paregut a això es que Node s'ha instal·lat OK:

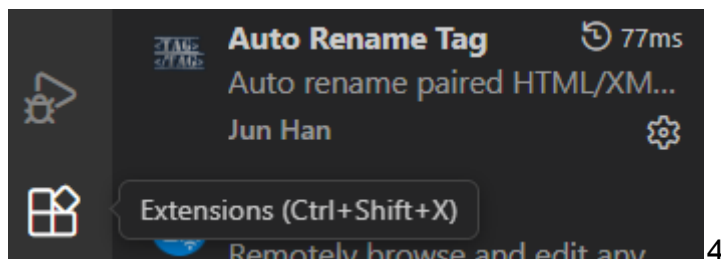
```
PS C:\Users\Usuario> node -v
v24.13.0
PS C:\Users\Usuario> npm -v
11.6.2
PS C:\Users\Usuario> |
```

3

### 3.2 Plugins de VS Code necessaris

- En aquest cas, treballem amb l'IDE Visual Studio Code, per a que el desenvolupament d'aquesta tasca sigui millor, ens descarregarem les següents extensions:

- *ESLint* -> Anàlisi de codi que detecta errors i malas pràctiques en JS i React
- *Prettier* -> Formatador automàtic de còdi que fa que tot el codi tingi el mateix estil visual, ordena automàticament “sangrías”, claus...
- *Auto Rename Tag* -> Canvia automàticament el tag de tancat quan es modifica un tag de obertura, és a dir, si jo modifico un *tag* `<div>` per un *tag* `<section>`, automàticament el *tag* de tancat es modifica de `</div>` a `</section>`.



4

(exemple de plugin “Auto Rename Tag” instal·lat)

### 3.3 Creació del projecte React (Vite)

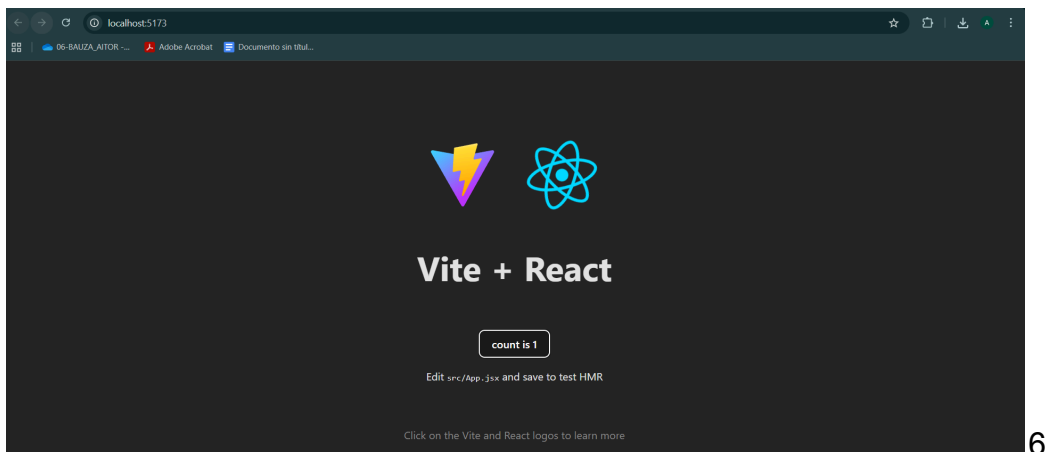
En la terminal de VS Code, crearem el projecte amb la següent comanda:

`npm create vite@latest gestor-despesas`

```
PS C:\Users\Usuario> npm create vite@latest gestor-despesas
Select a framework:
  o Vanilla
  o Vue
  • React
```

*(procés de creació del projecte, en aquest cas serà de React amb JS)*

- Una vegada creat, si anam al següent enllaç <http://localhost:5173/> i si tot ha anat OK, podrem veure la pàgina de Vite:



### 3.4 Neteja inicial

- Borram la carpeta `src/assets` (només inclou la foto del icono de React) i modifiquem el `App.jsx` així:

```
App.jsx
src > App.jsx > ...
1  function App() {
2    return (
3      <h1>Gestor de despeses (Aitor Bauzà Gómez)</h1>
4    )
5  }
6
7  export default App
```

- Dins el projecte feim `npm run dev` i tornem a arracar-ho:

```
PS C:\Users\Usuario\gestor-despesas> npm run dev
VITE v7.3.1 ready in 1044 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

8

- Vem com la pantalla inicial ha canviat:



### 3.5 Inicializam Git

- Encara no farem cap pujada al repositori, pero deixam el commit inicial fet:

```
PS C:\Users\Usuario\gestor-despesas> git init
>> git add .
>> git commit -m "Inicialització del projecte React amb Vite"
```

10

## 4. Crear projecte en Firebase

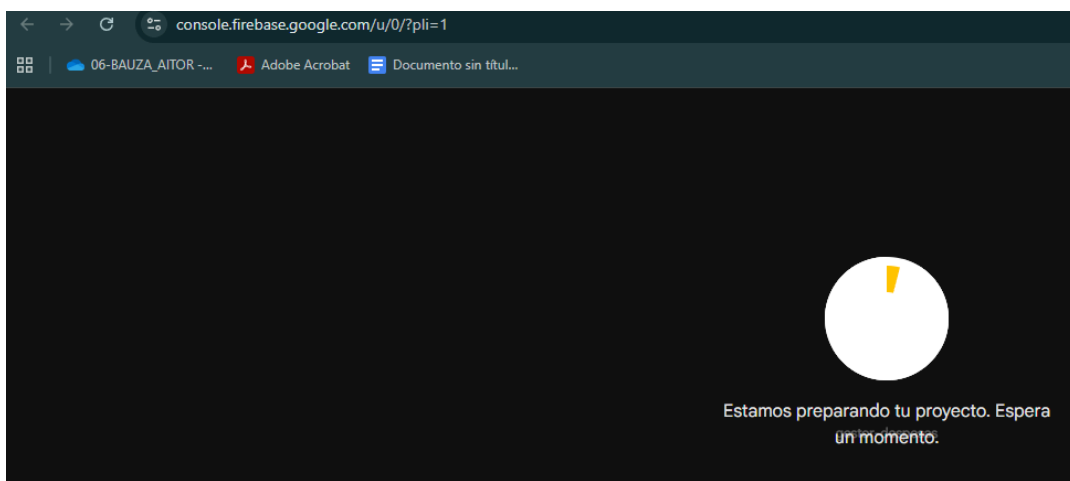
### 4.1 Entrar en Firebase

- Si anem al següent enllaç podeu crear el projecte:

<https://console.firebase.google.com/>

- Dins d'ell cream un projecte i li possem el nom que vulgueu, en el meu cas: gestor-despesas

- En el meu cas he deshabilitat “Google Analytics”, ja que no és necessari.

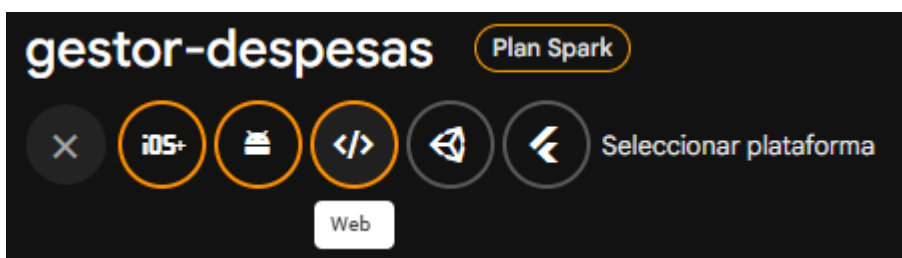


11

*(procés de creació del projecte)*

### 4.2 Creació de la app web en Firebase

- Cream dins del projecte la webapp gestor-despesas-web



12

*(sense Firebase Hosting per ara)*



- Ens donaran les configuracions necessaries per poder inicializar *Firebase*:

**2 Agrega el SDK de Firebase**

☒ Usar npm ☐ Usar una etiqueta <script>

Si ya usas [npm](#) y un agrupador de módulos como [Webpack](#) o [Rollup](#), puedes ejecutar el siguiente comando para instalar la versión más reciente del SDK ([más información](#)):

```
$ npm install firebase
```

Luego, inicializa Firebase y comienza a usar los SDK de los productos que quieres utilizar.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyDzQ0y6ZaW7NjdN7roGFecdu13l4ntAw",
  authDomain: "gestor-despesas-58b19.firebaseio.com",
  projectId: "gestor-despesas-58b19",
  storageBucket: "gestor-despesas-58b19.firebaseio.com",
  messagingSenderId: "607152210355",
  appId: "1:607152210355:web:2ec5135db50c3a030dcbc6"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

13

### 4.3 Configurar Firebase Auth

Anam al *menú lateral* -> *Compilación* -> *Authentication*

- Dins d'ell triam un mètode d'inici de sessió, en aquest cas, via *Email/Password*:

gestor-despesas ▾

## Authentication

Usuarios Método de acceso Plantillas Uso Configuración Extensiones

Proveedores de acceso

Agrega tu primer método de acceso y comienza a utilizar Firebase Auth

Proveedores nativos	Proveedores adicionales	Proveedores personalizados
Correo electrónico/contraseña	Google	OpenID Connect
Teléfono	Facebook	SAML
Anónimo	Play Games	
	Game Center	
	Apple	
	GitHub	
	Microsoft	
	Twitter	
	Yahoo	

14

Proveedores de acceso

Agregar proveedor nuevo

Proveedor	Estado
Correo electrónico/contraseña	Habilitada

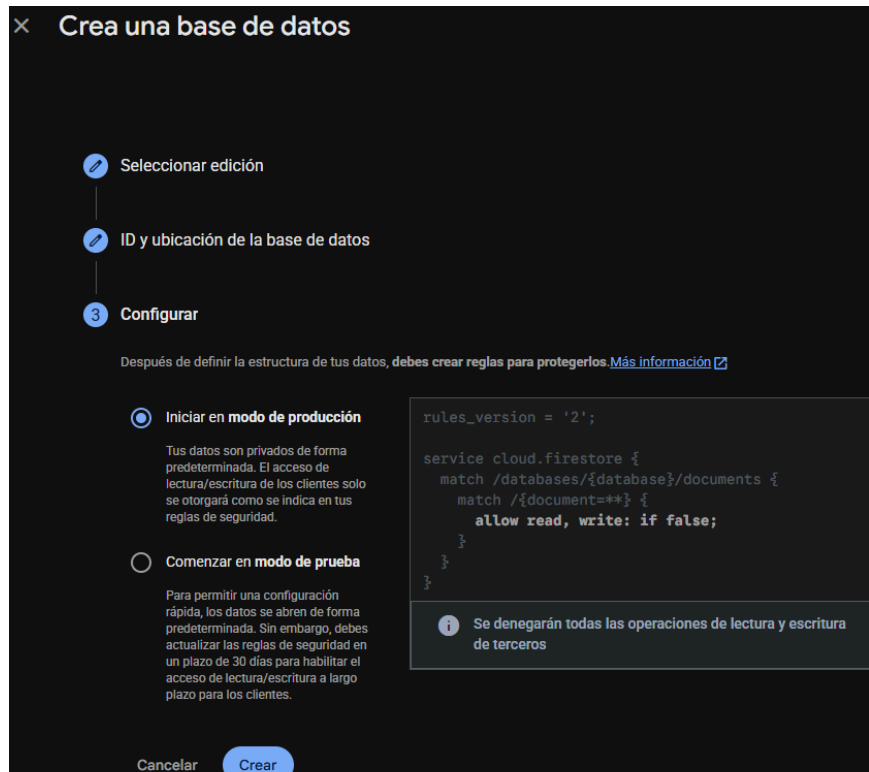
15

(ja tenim el mètode d'inici de sessió amb email/password)

## 4.4 Crear Firestore

És la base de dades que té Firebase.

- Anam al menú lateral -> Compilación -> Firestore Database:



16

(cream la BBDD, definim la regió més propera i el mode de producció)

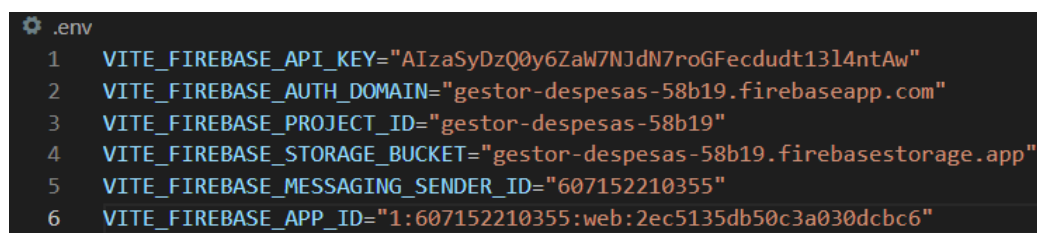
## 4.5 Instal·lar Firebase en React

- Anam a l'arrel del projecte i l'instal·lam amb la següent comanda:

`npm install firebase`

## 4.6 Variables d'entorn

- En l'arrel del projecte cream l'arxiu .env amb les variables de configuració que ens ha proporcionat Firebase al crear l'aplicació:



17

(Vite exigeix que comencen per "VITE\_")

- També cream l'arxiu `.env.example`, aquest arxiu es pujarà a *GitHub* i servirà com a plantilla per a que qualsevol altre desenvolupador sàpiga quines variables ha d'emplenar.

Les dades mai s'han de pujar, per això aquest arxiu, per no tenir que pujar el `.env`.  
*Amb gitignore farem que no es pugi.*

- `.env.example`:

```
1 VITE_FIREBASE_API_KEY=  
2 VITE_FIREBASE_AUTH_DOMAIN=  
3 VITE_FIREBASE_PROJECT_ID=  
4 VITE_FIREBASE_STORAGE_BUCKET=  
5 VITE_FIREBASE_MESSAGING_SENDER_ID=  
6 VITE_FIREBASE_APP_ID=
```

18

#### 4.7 Configuració de Firebase en el projecte

- Cream una nova carpeta `src/firebase` i, dins d'ella cream l'arxiu de configuració `"firebase.js"` amb els valors del `.env`:

```
src > firebase > JS firebase.js > ...  
1 import { initializeApp } from "firebase/app";  
2 import { getAuth } from "firebase/auth";  
3 import { getFirestore } from "firebase/firestore";  
4  
5 const firebaseConfig = {  
6   apiKey: import.meta.env.VITE_FIREBASE_API_KEY,  
7   authDomain: import.meta.env.VITE_FIREBASE_AUTH_DOMAIN,  
8   projectId: import.meta.env.VITE_FIREBASE_PROJECT_ID,  
9   storageBucket: import.meta.env.VITE_FIREBASE_STORAGE_BUCKET,  
10  messagingSenderId: import.meta.env.VITE_FIREBASE_MESSAGING_SENDER_ID,  
11  appId: import.meta.env.VITE_FIREBASE_APP_ID,  
12 };  
13  
14 const app = initializeApp(firebaseConfig);  
15  
16 export const auth = getAuth(app);  
17 export const db = getFirestore(app);
```

19

## 4.8 Probar connexió amb Firebase

- Modificam l'arxiu *App.jsx* només per provar:

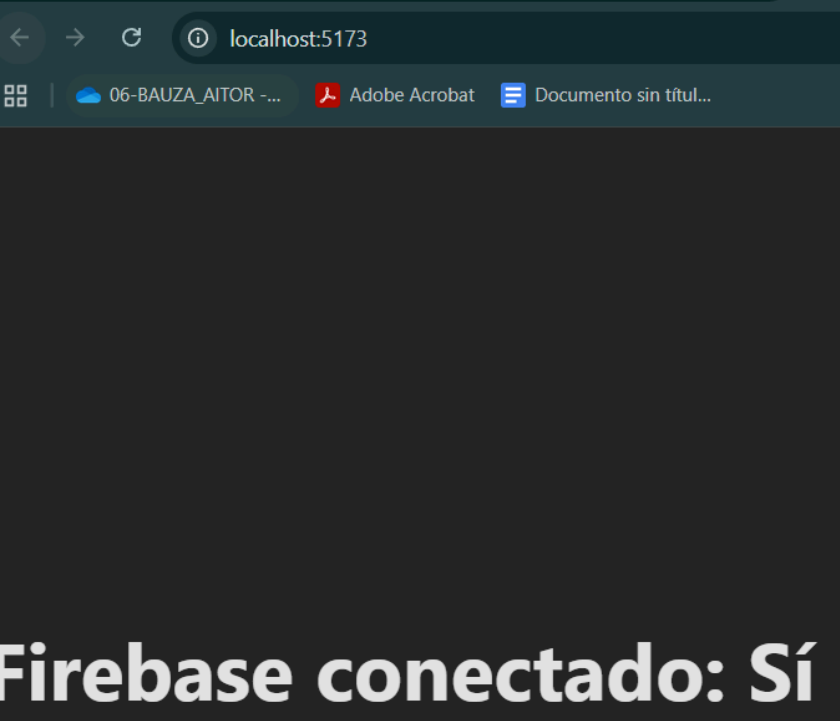
```
App.jsx > ...
import { auth } from "../firebase/firebase";

function App() {
  return <h1>Firebase conectado: {auth ? "Sí" : "No"}</h1>;
}

export default App;
```

20

- Si tornem a arrancar el projecte, veurem això:



A screenshot of a web browser window. The address bar shows 'localhost:5173'. The browser has several tabs open: '06-BAUZA\_AITOR - ...', 'Adobe Acrobat', and 'Documento sin título...'. The main content area of the browser is dark and displays the text 'Firebase conectado: Sí' in a large, white, sans-serif font.

21

## 5. Autenticació d'usuaris

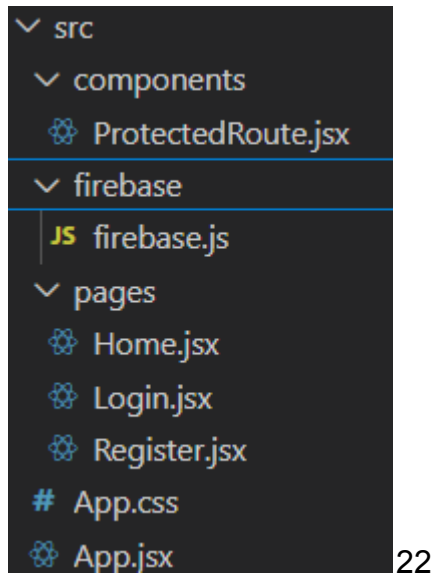
### 5.1 Instal·lar React Router

- En l'arrel del projecte l'instal·lam amb la següent comanda:

```
npm install react-router-dom
```

## 5.2 Creació de carpetes i arxius

- Aquesta és l'estructura de carpetes del projecte:



22

- *Login.jsx* (una part d'ell):

```
src > pages > Login.jsx > Login
1  import { useState } from "react";
2  import { signInWithEmailAndPassword } from "firebase/auth";
3  import { auth } from "../firebase/firebase";
4  import { useNavigate, Link } from "react-router-dom";
5
6  function Login() {
7      const [email, setEmail] = useState("");
8      const [password, setPassword] = useState("");
9      const [error, setError] = useState("");
10     const navigate = useNavigate();
11
12     const handleSubmit = async (e) => {
13         e.preventDefault();
14         setError("");
15
16         try {
17             await signInWithEmailAndPassword(auth, email, password);
18             navigate("/");
19         } catch (err) {
20             setError("Credencials incorrectes. Torna-ho a intentar.");
21         }
22     };
23
24     return (
25         <div>
26             <h2>Inicia sessió</h2>
27
28             <form onSubmit={handleSubmit}>
29                 <input
30                     type="email"
31                     placeholder="Email"
32                     value={email}
33                     onChange={(e) => setEmail(e.target.value)}
34                     required
```

23

(“err” dona error perquè encara no s’emplea)

- *Register.jsx* (una part d'ell):

```
import { useState } from "react";
import { createUserWithEmailAndPassword } from "firebase/auth";
import { auth } from "../firebase/firebase";
import { useNavigate, Link } from "react-router-dom";

function Register() {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [error, setError] = useState("");
  const navigate = useNavigate();

  const handleSubmit = async (e) => {
    e.preventDefault();
    setError("");

    try {
      await createUserWithEmailAndPassword(auth, email, password);
      navigate("/");
    } catch (err) {
      setError("Error al crear el compte. Torna-ho a intentar.");
    }
  };

  return (
    <div>
      <h2>Registre</h2>

      <form onSubmit={handleSubmit}>
        <input
          type="email"
          placeholder="Email"

```

24

(“err” dona error perquè encara no s’emplea)

- *ProtectedRoute.jsx* (sencer):

```
src > components > ProtectedRoute.jsx > default
1  import { Navigate } from "react-router-dom";
2  import { onAuthStateChanged } from "firebase/auth";
3  import { auth } from "../firebase/firebase";
4  import { useEffect, useState } from "react";
5
6  function ProtectedRoute({ children }) {
7    const [user, setUser] = useState(undefined);
8
9    useEffect(() => {
10      const unsubscribe = onAuthStateChanged(auth, (currentUser) => {
11        setUser(currentUser);
12      });
13
14      return () => unsubscribe();
15    }, []);
16
17    if (user === undefined) return <p>Carregant...</p>;
18
19    return user ? children : <Navigate to="/login" />;
20  }
21
22  export default ProtectedRoute;
```

25

(si l’usuari no està autenticat, depenent la pàgina on vulgui anar, el redireccionarà cap el login)

- Modificam *App.jsx*:

```
src > App.jsx > App
1 import { BrowserRouter, Routes, Route } from "react-router-dom";
2 import Login from "../pages/Login";
3 import Register from "../pages/Register";
4 import Home from "../pages/Home";
5 import ProtectedRoute from "../components/ProtectedRoute";
6
7 function App() {
8   return (
9     <BrowserRouter>
10      <Routes>
11        <Route path="/login" element={<Login />} />
12        <Route path="/register" element={<Register />} />
13
14        <Route
15          path="/"
16          element={
17            <ProtectedRoute>
18              <Home />
19            </ProtectedRoute>
20          }
21        />
22      </Routes>
23    </BrowserRouter>
24  );
25 }
26
27 export default App;
```

26

(configuram les rutes)


- *Home.jsx* (amb logout):

```
src > pages > Home.jsx > Home
1 import { signOut } from "firebase/auth";
2 import { auth } from "../firebase/firebase";
3
4 function Home() {
5   const handleLogout = async () => {
6     await signOut(auth);
7   };
8
9   return (
10     <div>
11       <h1>Home - Usuari autenticat</h1>
12       <button onClick={handleLogout}>Tancar sessió</button>
13     </div>
14   );
15 }
16
17 export default Home;
```

27

### 5.3 Proves del funcionament

- Crear compte:




**Registre**

abauza@gmail.com ..... **Registrar-se**

¿Ja tens compte? [Inicia sessió](#)

28

- Iniciar sessió:



**Inicia sessió**

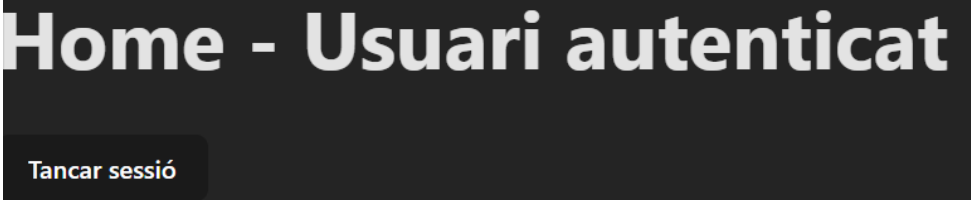
abauza123@gmail.com ..... **Entrar**

Credencials incorrectes. Torna-ho a intentar.

¿No tens compte? [Registra't](#)

29

*(amb credencials incorrectes)*



**Home - Usuari autenticat**

Tancar sessió

30

*(si les credencials són correctes apareix això)*



## 6. Gestió de projectes

### 6.1 Modificació i creació d'arxius

- Modificam *Home.jsx* (una part d'ell):

```
src > pages > Home.jsx > Home
1  import { useEffect, useState } from "react";
2  import { db, auth } from "../firebase/firebase";
3  import { collection, query, where, getDocs, addDoc, serverTimestamp } from "firebase/firestore";
4
5  function Home() {
6    const [projects, setProjects] = useState([]);
7    const [title, setTitle] = useState("");
8
9    const fetchProjects = async () => {
10     const q = query(collection(db, "projects"), where("authorId", "==", auth.currentUser.uid));
11     const querySnapshot = await getDocs(q);
12     const list = querySnapshot.docs.map(doc => ({ id: doc.id, ...doc.data() }));
13     setProjects(list);
14   };
15
16   useEffect(() => {
17     const loadProjects = async () => {
18       await fetchProjects();
19     };
20     loadProjects();
21   }, []);
22
23   const handleCreateProject = async (e) => {
24     e.preventDefault();
25     if (!title.trim()) return;
26
27     await addDoc(collection(db, "projects"), {
28       title,
29       authorId: auth.currentUser.uid,
30       participants: [auth.currentUser.email], // per ara usem email com a nom
31       createdAt: serverTimestamp()
```

31

*(ara es poden crear, editar i eliminar projectes des del Home)*

- Modificam *App.jsx* (una part d'ella):

```
    <Route
      path="/project/:id"
      element={
        <ProtectedRoute>
          <ProjectDetail />
        </ProtectedRoute>
      }
    />
  </Routes>
</BrowserRouter>
);
```

32

*(preparam la ruta i navegació del proper arxiu)*

- `pages/ProjectDetail.jsx` (una part d'ell):

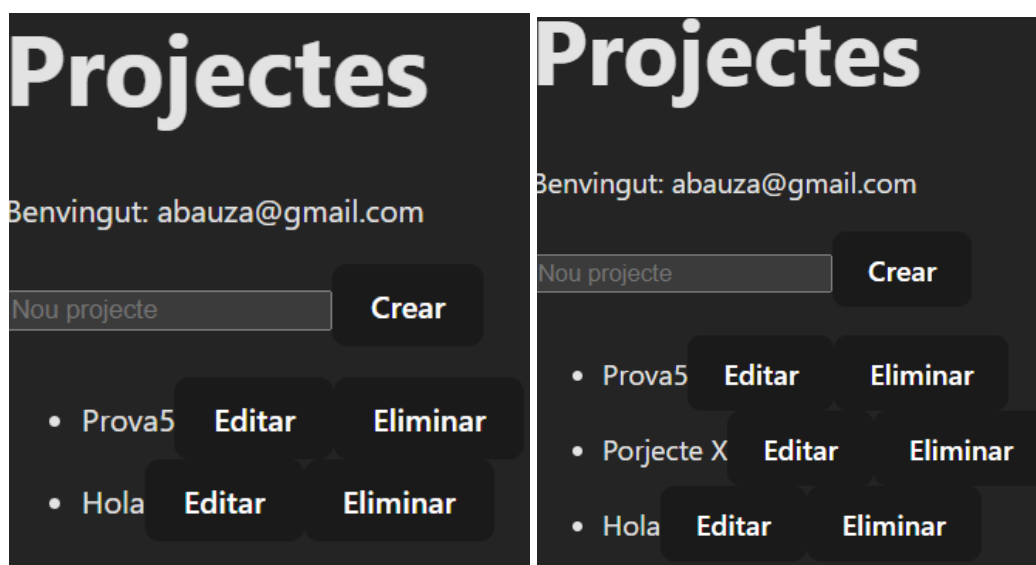
```
src > pages > ProjectDetail.jsx > ProjectDetail > useEffect() callback
1  import { useParams } from "react-router-dom";
2  import { useEffect, useState } from "react";
3  import { db } from "../firebase/firebase";
4  import { doc, getDoc, updateDoc } from "firebase/firestore";
5
6  function ProjectDetail() {
7    const { id } = useParams();
8    const [project, setProject] = useState(null);
9    const [newParticipant, setNewParticipant] = useState("");
10
11    const fetchProject = async () => {
12      const docRef = doc(db, "projects", id);
13      const docSnap = await getDoc(docRef);
14      if (docSnap.exists()) {
15        setProject({ id: docSnap.id, ...docSnap.data() });
16      }
17    };
18
19    useEffect(() => {
20      const loadProjects = async () => {
21        await fetchProject();
22      };
23      loadProjects();
24    }, []);
25
26    const handleAddParticipant = async () => {
27      if (!newParticipant.trim() || project.participants.includes(newParticipant)) return;
28
29      const docRef = doc(db, "projects", id);
```

33

(mostra els detalls dels projectes i pot afegir participants)

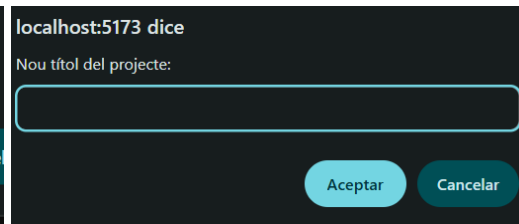
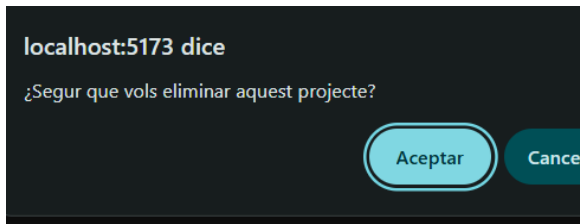
## 6.2 Proves del funcionament

- Podem crear, editar i eliminar projectes:



34/35

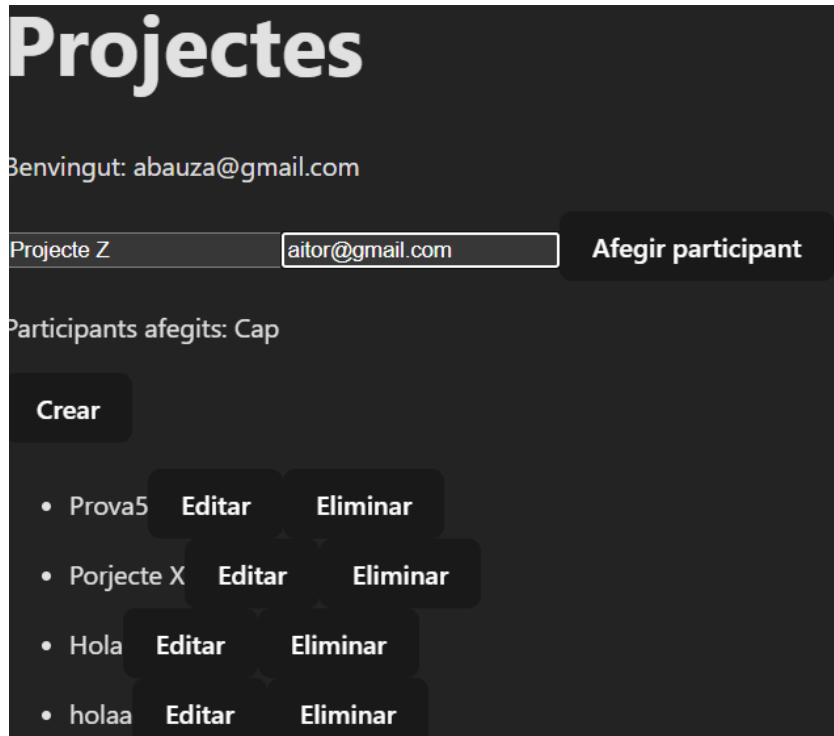
(projectes creats / nou projecte creat)



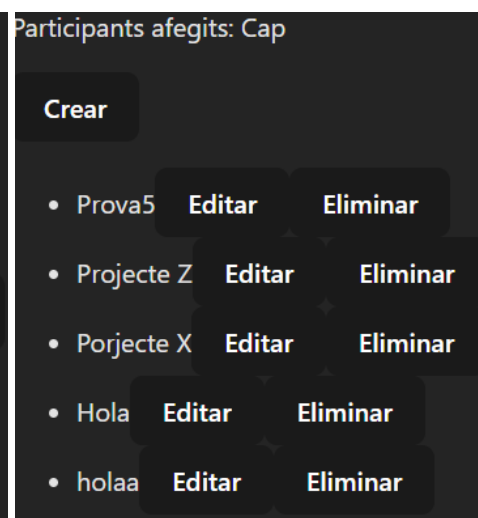
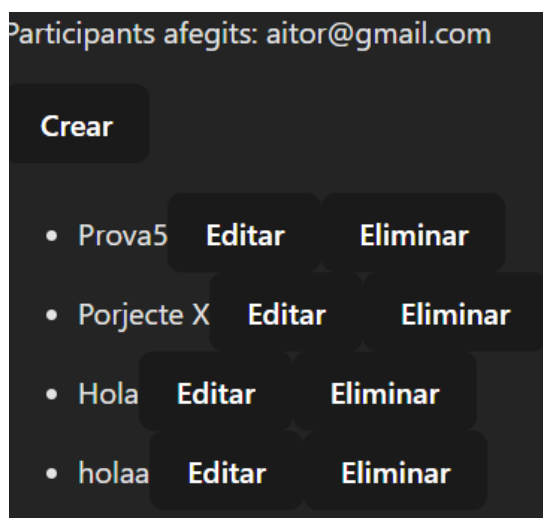
36/37

(eliminar projecte / editar projecte)

- Afegir participant:

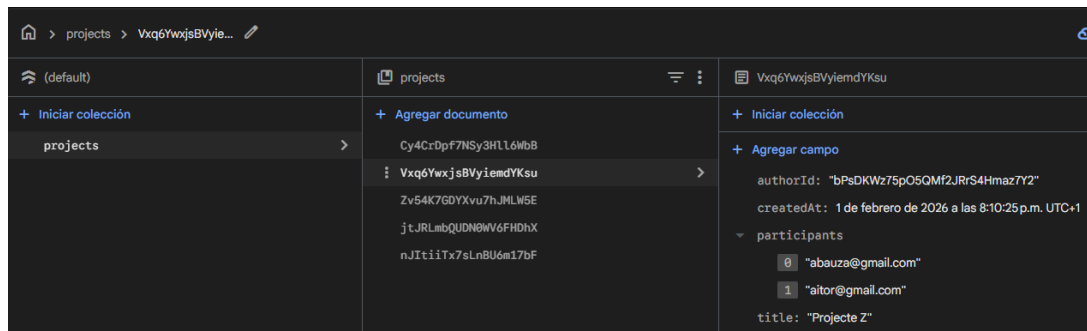


38



39/40

- Dins *Firebase*:



41

## 7. Gestió de despeses

### 7.1 Modificació i creació d'arxius

- *ProjectDetail.jsx*:

```
// Añadir despesa
const handleAddExpense = async (e) => {
  e.preventDefault();
  if (!concept.trim() || !amount || splitBetween.length === 0) return;

  await addDoc(collection(db, "projects", id, "expenses"), {
    concept,
    amount: parseFloat(amount),
    paidBy,
    splitBetween
  });

  setConcept("");
  setAmount("");
  setPaidBy(auth.currentUser.email);
  setSplitBetween(project.participants);
  fetchExpenses();
};
```

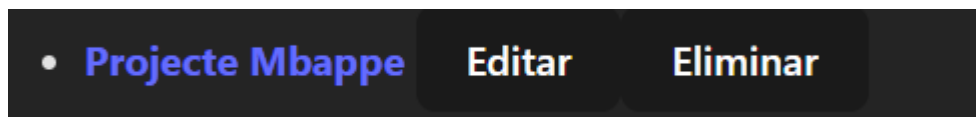
42

```
// Cargar despeses
const fetchExpenses = async () => {
  if (!project) return;
  const expensesRef = collection(db, "projects", id, "expenses");
  const snapshot = await getDocs(expensesRef);
  const list = snapshot.docs.map(doc => ({ id: doc.id, ...doc.data() }));
  setExpenses(list);
};
```

43

### 7.2 Proves del funcionament

- Si clicam dins el projecte:



44

- Ens apareixerà el formulari de les despeses:

**Projecte Mbappe**

**Participants**

- abauza@gmail.com
- aitor@gmail.com

Afegir participant **Afegir**

**Despeses**

Concepte Quantia abauza@gmail.com ▼

Dividir entre: abauza@gmail.com aitor@gmail.com

☒ abauza@gmail.com ☒ aitor@gmail.com

**Afegir despesa**

45

- Afegir participant:

**Participants**

- abauza@gmail.com
- aitor@gmail.com
- hola@gmail.com

Afegir participant **Afegir**

**Despeses**

Concepte Quantia abauza@gmail.c

Dividir entre:

☒ abauza@gmail.com ☒ aitor@gmail.com ☒ hola@gmail.com

46

(“dividir entre” té un nou participant)

- Afegir despesa:

**Despeses**

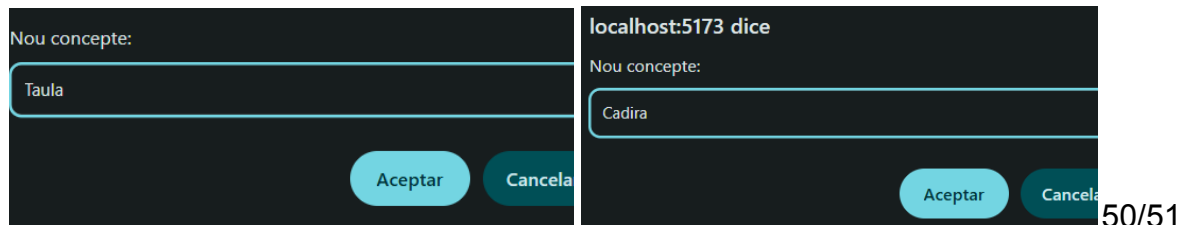
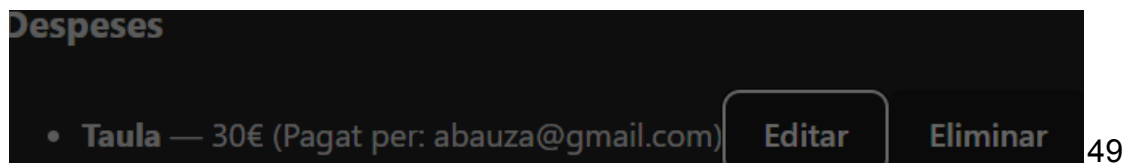
Taula 30 abauza@gmail.com ▼

47

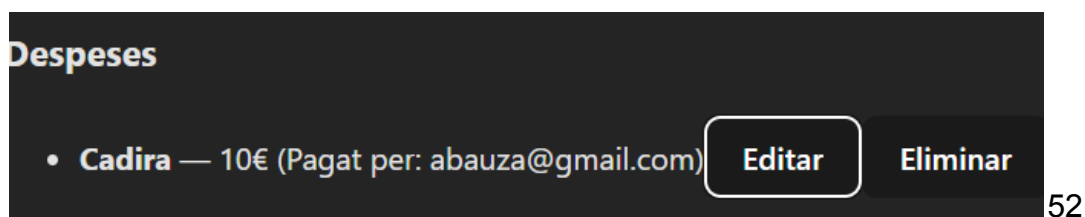
• Taula - 30 € - Pagat per: abauza@gmail.com - Dividit entre: abauza@gmail.com, aitor@gmail.com **Eliminar**

48

- Editar despesa:

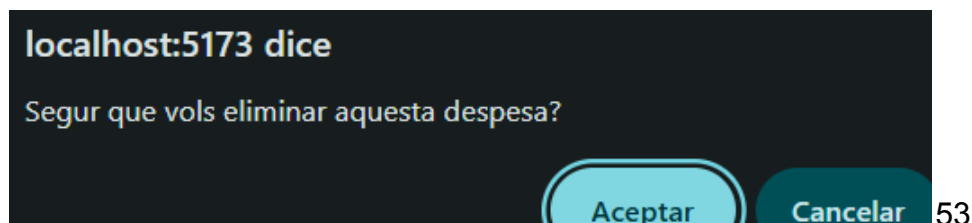


(de taula a cadira)

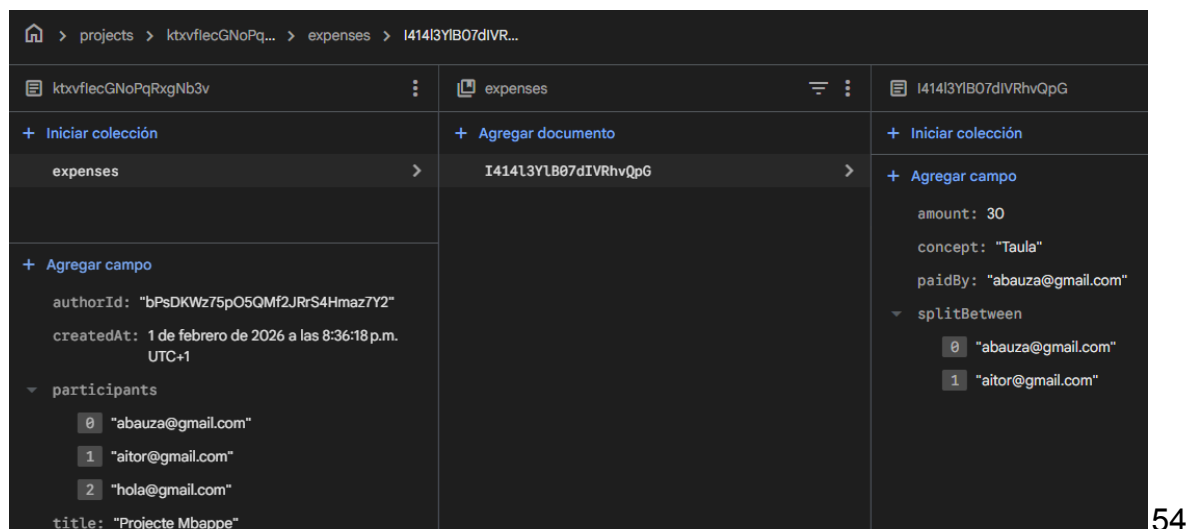


(també hem camviat el preu)

- Eliminar despesa:



- Dins Firebase:



## 8. Gestió de participants

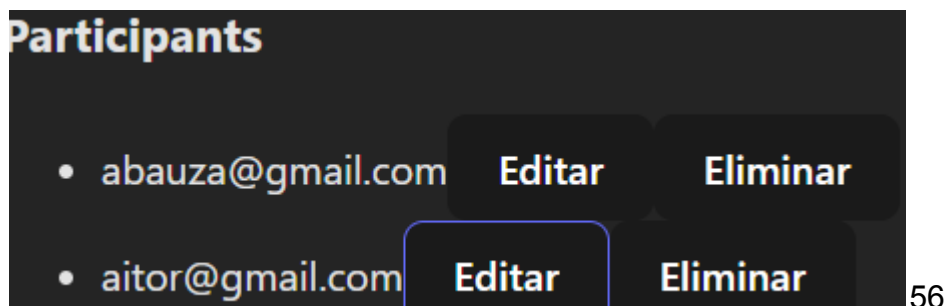
### 8.1 Modificació i creació d'arxius

- *ProjectDetail.jsx*:

```
src > pages > ProjectDetail.jsx > ProjectDetail
14  function ProjectDetail() {
47    const handleAddParticipant = async () => {
57    };
58
59    // Editar participant
60    const handleEditParticipant = async (oldName) => {
61      const newName = prompt("Nou nom del participant:", oldName);
62      if (!newName || !newName.trim()) return;
63
64      const updated = project.participants.map((p) =>
65      | p === oldName ? newName.trim() : p
66      );
67
68      await updateDoc(doc(db, "projects", id), {
69      | participants: updated
70      });
71
72      fetchProject();
73    };
74
75    // Eliminar participant
76    const handleDeleteParticipant = async (name) => {
77      if (!window.confirm("Eliminar participant?")) return;
78
79      const updated = project.participants.filter((p) => p !== name);
80
81      await updateDoc(doc(db, "projects", id), {
82      | participants: updated
83      });
84
85      fetchProject();
86    };
55
```

### 8.2 Proves del funcionament

- Nous botons:



- Editar participant:

localhost:5173 dice  
Nou nom del participant:  
aitor@gmail.com  
Aceptar

localhost:5173 dice  
Nou nom del participant:  
aitoraitor@gmail.com  
Aceptar

(abans vs després)

• abauza@gmail.com Editar Eliminar

• aitoraitor@gmail.com Editar Eliminar

(resultat final)

- Eliminar participant:

localhost:5173 dice  
Eliminar participant?  
Aceptar Cancelar

projepte Mbappe

Participants

• abauza@gmail.com Editar Eliminar

• aitoraitor@gmail.com Editar Eliminar

Participants

• abauza@gmail.com Editar Eliminar

(resultat final)



9. Afegir un panell de resum que mostri el total de despeses per projecte i la quantitat que cada participant deu o ha de rebre segons la divisió de les despeses.

### 9.1 Modificació i creació d'arxius

- *ProjectDetail.jsx*:

```
const calculateSummary = () => {
  const summary = {};

  // inicializar
  project.participants.forEach((p) => {
    summary[p] = 0;
  });

  expenses.forEach((e) => {
    const share = e.amount / e.splitBetween.length;

    // quien paga
    summary[e.paidBy] += e.amount;

    // quien debe
    e.splitBetween.forEach((p) => {
      summary[p] -= share;
    });
  });

  return summary;
}
```

62

### 9.2 Proves del funcionament

- Si no hi ha despeses:

**Despeses**

No hi ha despeses

**Resum del projecte**

**Total despeses: 0 €**

- abauza@gmail.com: està a zero
- aitor@gmail.com: està a zero

63

- Cream una despesa de 370€ pagada per abauza:

## Despeses

- **Ps4** — 370€

Pagat per: abauza@gmail.com

Dividit entre: abauza@gmail.com, aitor@gmail.com

Editar

Eliminar

## Resum del projecte

**Total despeses:** 370 €

- abauza@gmail.com: ha de rebre 185.00 €
- aitor@gmail.com: deu 185.00 €

64

## 10. CONCLUSIÓ

Gràcies a aquest treball he descobert un món totalment nou, tant a nivell de back com a nivell de front, ja que he après i entès com funciona React, els seus estats i components, a més de que he descobert el BaaS Firebase (el desconeixia totalment) aportant-me moltes facilitats, com el seu propi *Auth* o *Firestore*.

De fet, aporta tantes facilitats que no fa falta escriure cap línia de codi :)

## 11. BIBLIOGRAFÍA

- Documentació d'aquest mòdul emprada:

*Carlos Canals Gozálvez. (2026). Introducció a React. fpadistancia:*

[https://fpadistancia.caib.es/pluginfile.php/211411/mod\\_resource/content/6/Introduccio-a-React-final.pdf](https://fpadistancia.caib.es/pluginfile.php/211411/mod_resource/content/6/Introduccio-a-React-final.pdf)

*Carlos Canals Gozálvez. (2026). React amb Firebase fpadistancia:*

[https://fpadistancia.caib.es/pluginfile.php/211452/mod\\_resource/content/2/React-amb-Firebase.pdf](https://fpadistancia.caib.es/pluginfile.php/211452/mod_resource/content/2/React-amb-Firebase.pdf)

- Documentació externa:

*W3Schools. (2026). React Tutorial. w3schools:*

<https://www.w3schools.com/REACT/DEFAULT.ASP>

(abans de començar li he donat una ullada a aquest recurs, ja que he fet molt poc React)

## 12. REPOSITORI GITHUB

<https://github.com/aitorbauza/gestor-despesas.git>