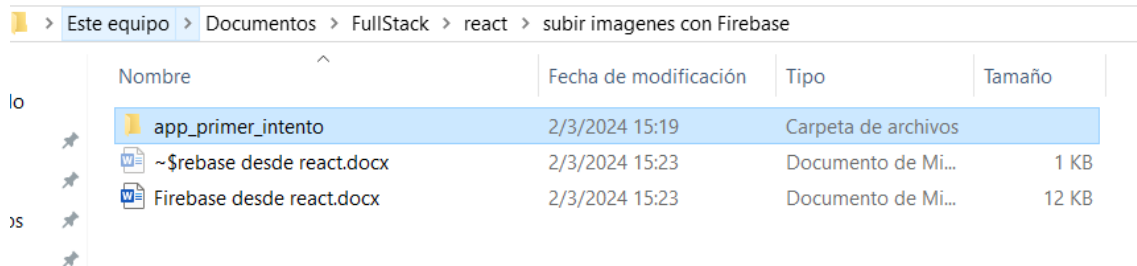


Firebase desde react

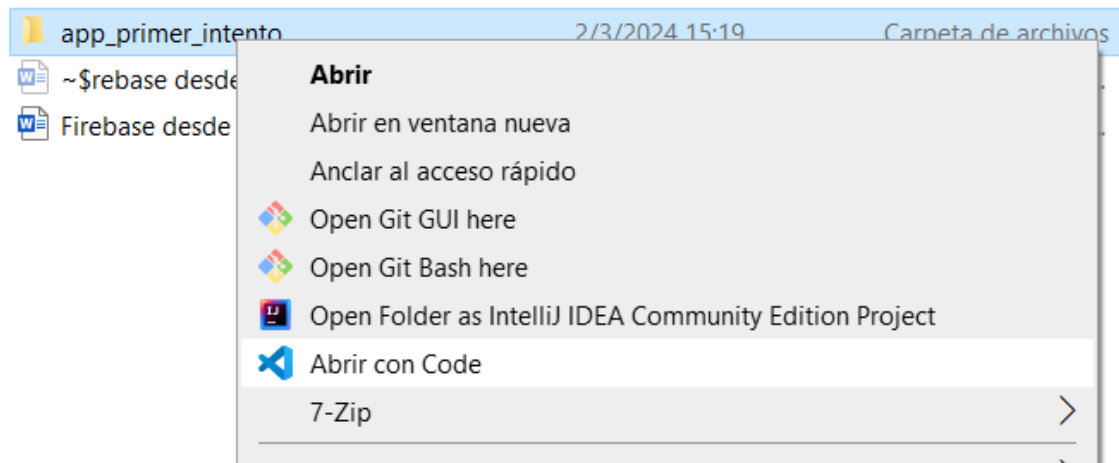
Métodos utilizados UploadBytes,getDownloadURL

1 preparo entorno de trabajo

Creo una carpeta:



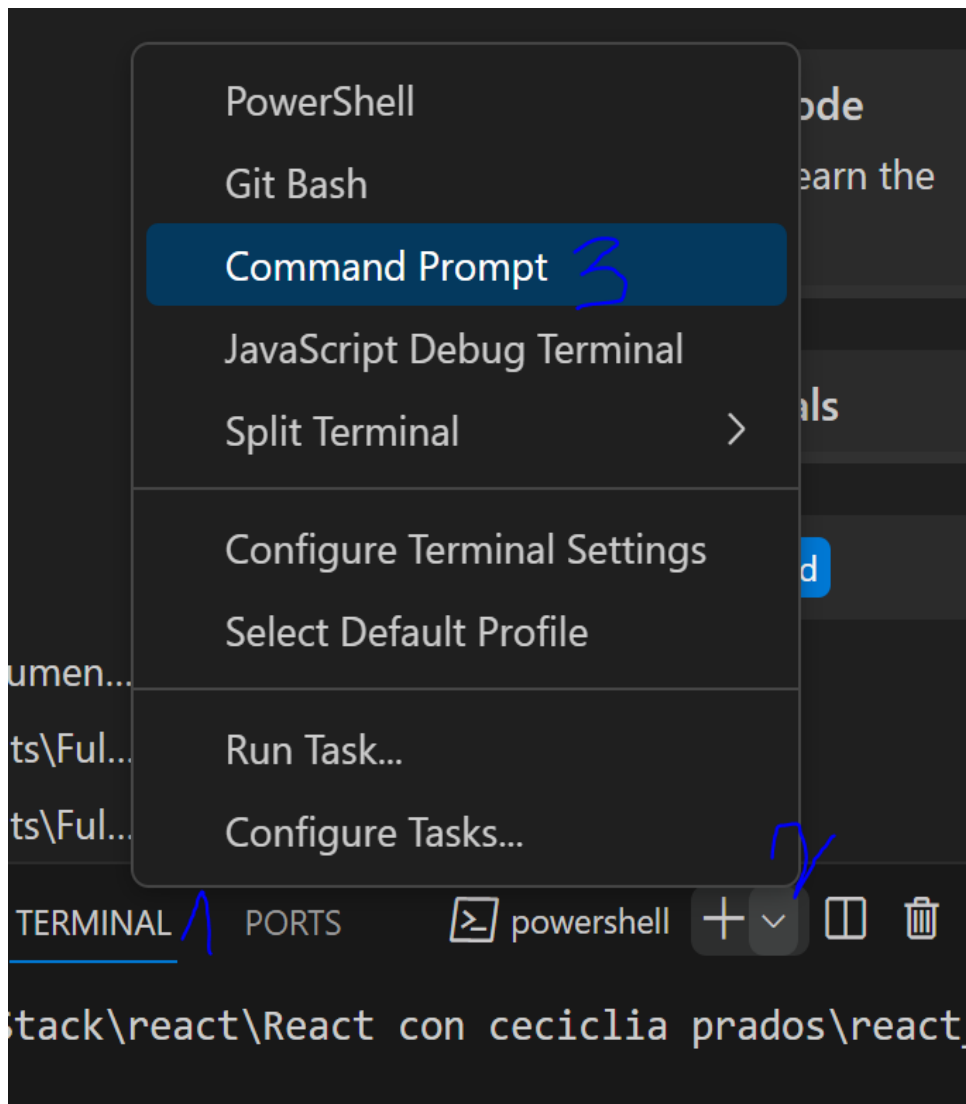
Lo abro desde un editor por ejemplo Visual studio code



Luego desde visual studio , voy a terminal....



Selecciono el tipo de terminal, CMD



Tipeo npm create vite@latest

```
? Project name: » vite-project
```

Le asignamos un nombre al proyecto, si solo le pongo punto es decir . toma la carpeta donde estamos parados.

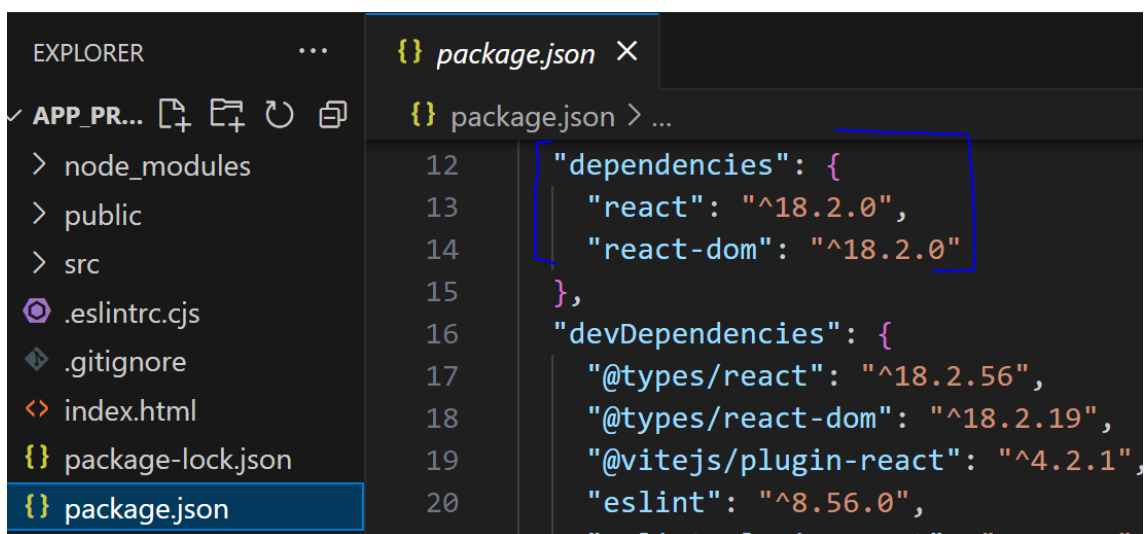
```
Vue
> React ✓
Preact
Lit
Svelte
Solid
Qwik
```

Seleccionamos la variante:

```
? Select a variant: » - U
TypeScript
TypeScript + SWC
> JavaScript ✓
JavaScript + SWC
Remix ↗
```

Instalamos el resto de las librerías con el comando `npm install`

Verificamos el archivo



```
EXPLORER
APP_PR...
  > node_modules
  > public
  > src
.eslintrc.cjs
.gitignore
index.html
package-lock.json
package.json

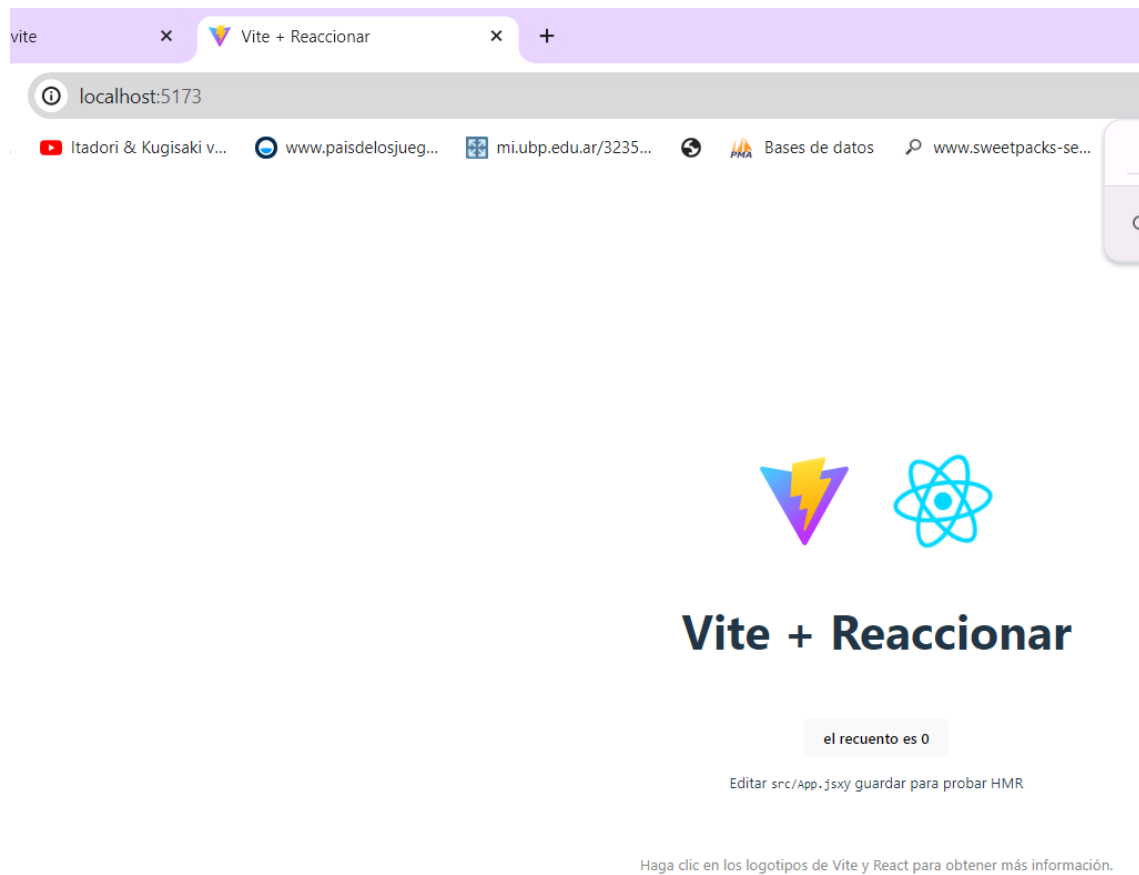
package.json
12  "dependencies": {
13    "react": "^18.2.0",
14    "react-dom": "^18.2.0"
15  },
16  "devDependencies": {
17    "@types/react": "^18.2.56",
18    "@types/react-dom": "^18.2.19",
19    "@vitejs/plugin-react": "^4.2.1",
20    "eslint": "^8.56.0",
```

`npm run dev` //para ejecutar el servidor

Y lo pruebo en el Chrome, copiando el link de Local

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

En el Chrome lo pruebo:



Al archivo de src App.jsx lo dejo:

```
1 import { useState } from 'react'
2 import './App.css'
3
4 function App() {
5   return (
6     <>
7   </>
8   )
9 }
10
11 export default App
12
```

```
1
```

```
1
```

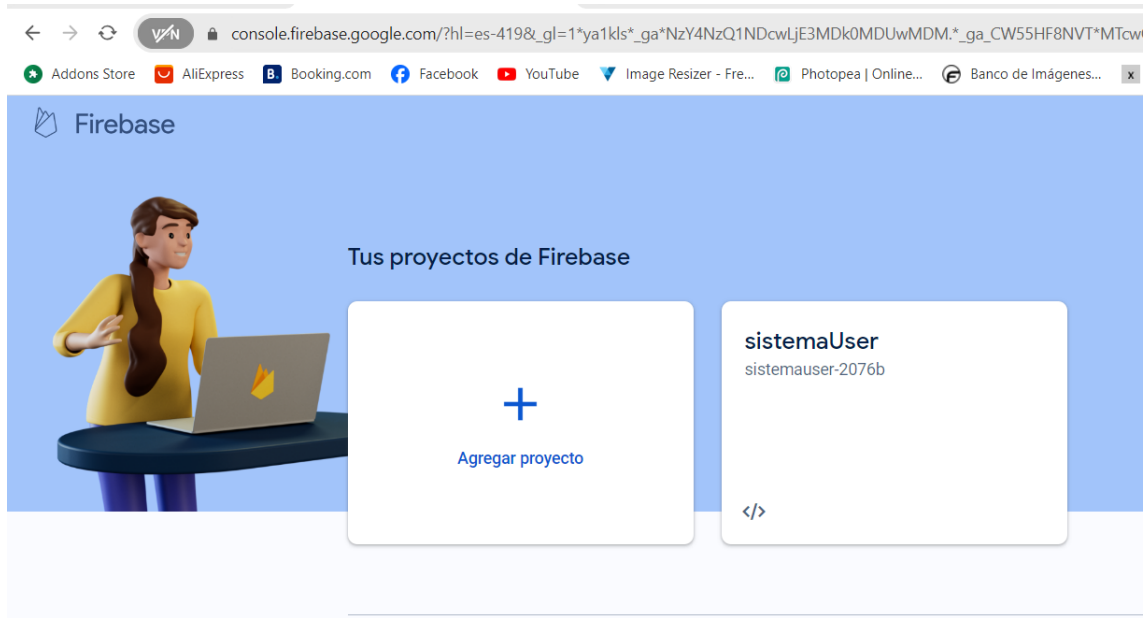
2configuramos nuestra nube FireBase

El link es

<https://firebase.google.com/?hl=es-419>

Comenzar

Y aparece la siguiente pantalla:



Agregar proyecto

.

# Comencemos con el nombre de tu proyecto<sup>?</sup>

Nombre del proyecto

react-nubeGoogle-galeria

react-nube-galeria

Continuar

☐ Habilitar Google Analytics para este proyecto  
Recomendado

Anterior

Crear proyecto

Instalamos en la terminal CMD de nuestro editor de código ejemplo Visual y tipiamos en el Google firebase npm:

Microsoft Bing

npm firebase

BÚSQUEDA

COPILOT

VÍDEOS

IMÁGENES

MAPAS

NOTICIAS

MÁS

Aproximadamente 2.080.000 resultados



npm

<https://www.npmjs.com/package/firebase>

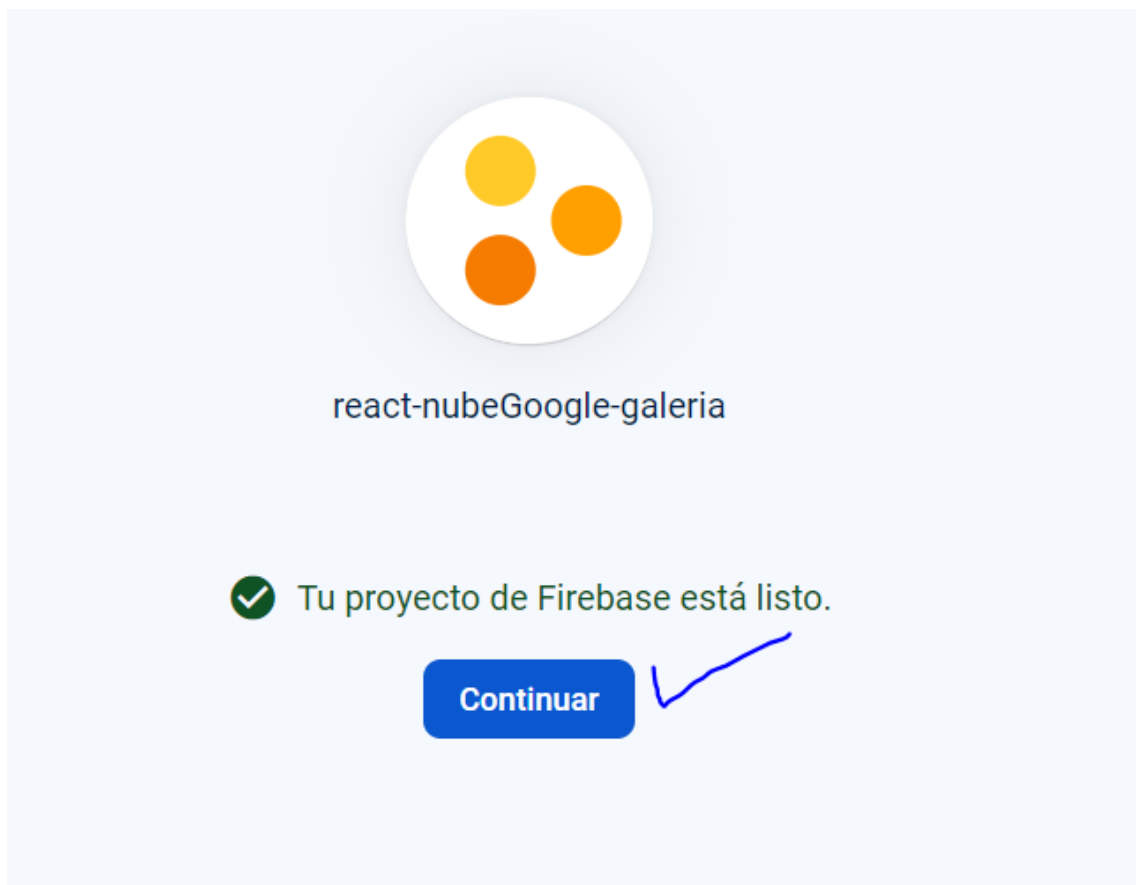
firebase - npm

Web Firebase is a platform that provides tools and infrastructure for web, mobile, and server development. Learn how to install, use, and configure **Firebase** services with the **npm** ...

For more information, visit:

- **Firestore Realtime Database** - The Firestore Realtime Database lets you store and query user data, and makes it available between users in realtime.
- **Cloud Firestore** - Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firestore and Google Cloud Platform.
- **Firestore Storage** - Firestore Storage lets you upload and store user generated content, such as files, and images.

Y nos lleva a la siguiente pantalla <https://firebase.google.com/docs/storage/web/start?hl=es-419>



3ero configuro mi archivo App.jsx:

La cual le agrego una caja de texto con el tipo file con console.log para analizar lo que trae:

```
import './App.css'

function App() {
  return (
    <>
      <div>
        <input type='file' name='' id='' onChange={e => console.log(e)}/>
      </div>
    </>
  )
}
```

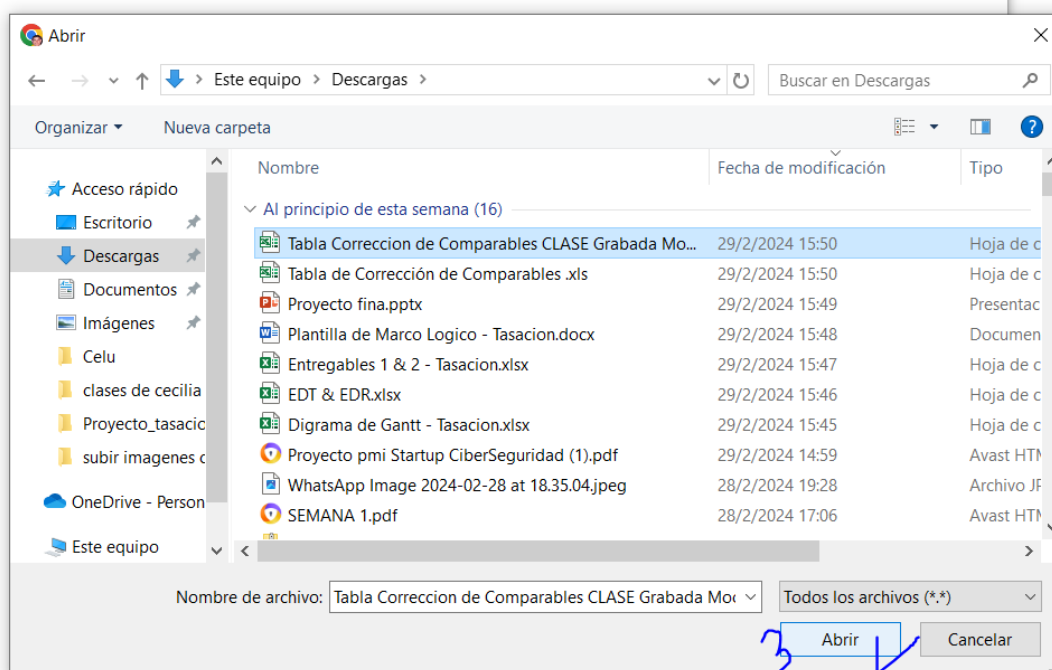
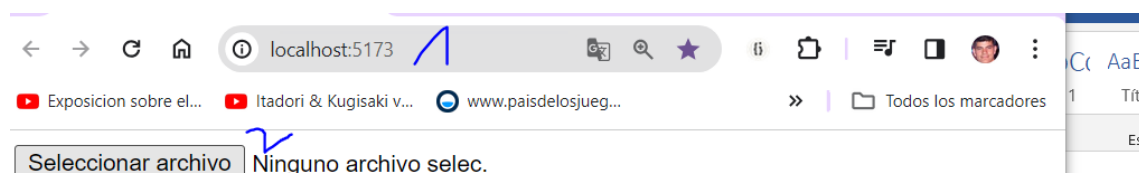
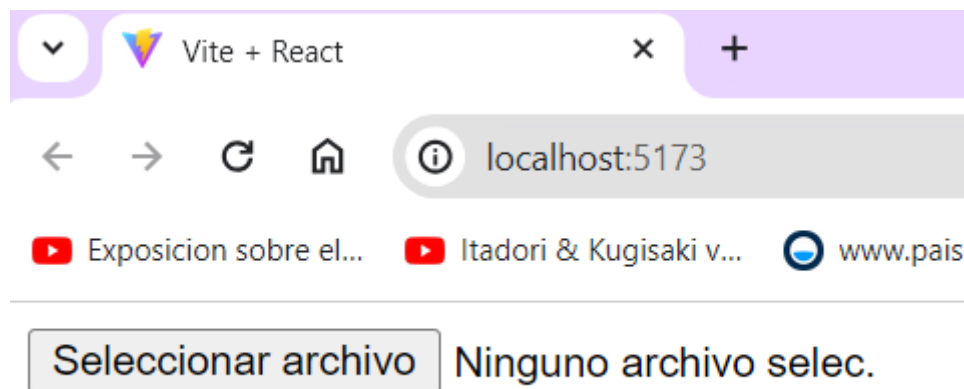


```

    </>
  )
}
export default App

```

y si pruebo en el navegador:



Click derecho en el cuerpo del navegador

▲

Buscar imágenes con Google



Enviar a tus dispositivos



Crear código QR para esta página

Traducir a español

Abrir en modo Lectura **NUEVO**



Adblock: el mejor bloqueador de anuncios



Concéntrate con BlockSite

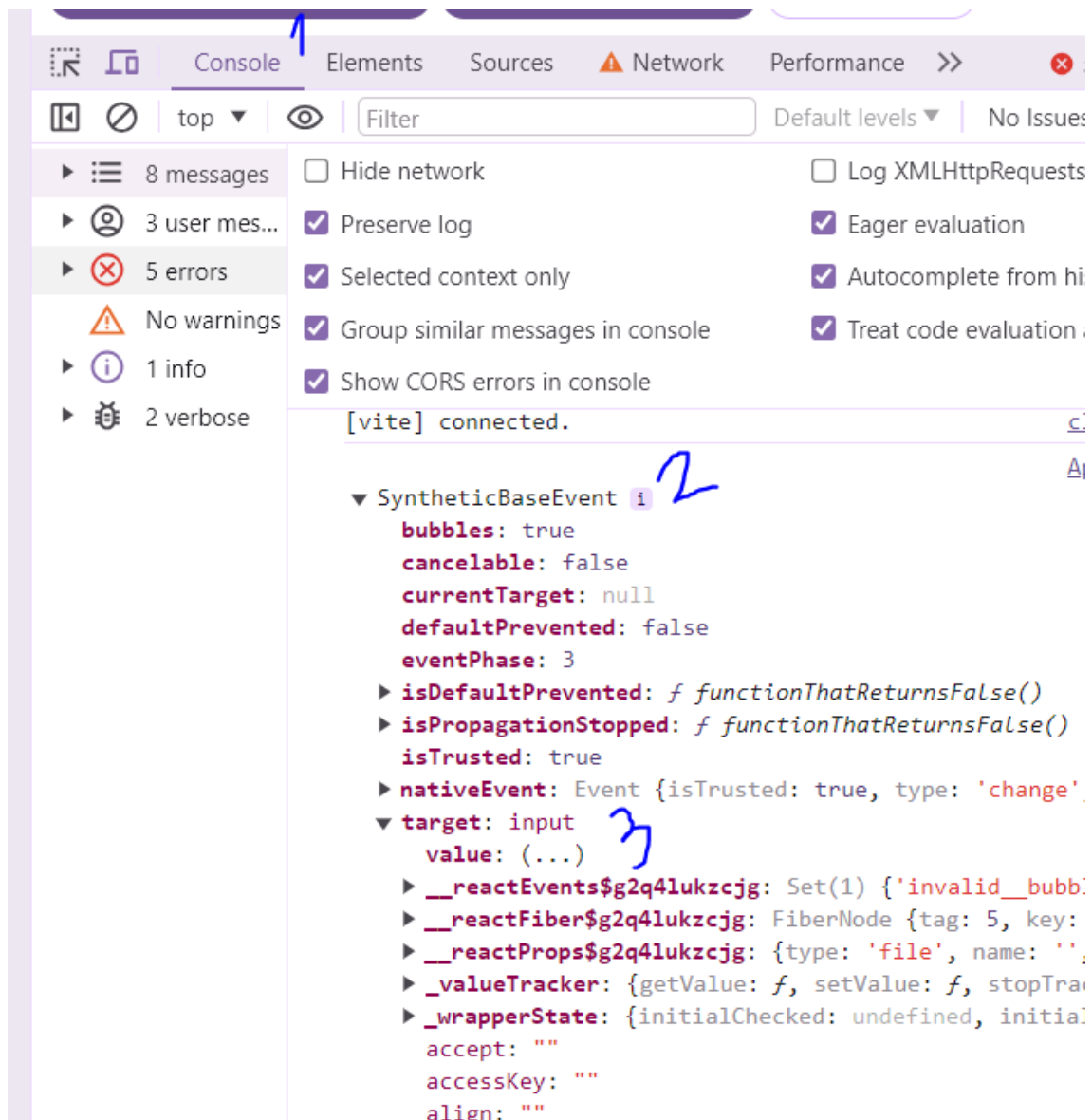


Ver código fuente de la página

Ctrl + U

Inspeccionar





Y busco el valor file que en realidad se comporta como lista que es lo que nos interesa

El mismo muestra un objeto con índice

```

enter keyname.
▼ files: FileList
  ▼ 0: File
    lastModified: 1709232647609
    ► lastModifiedDate: Thu Feb 29 2024 15:50:47 GMT-0300 (hor
      name: "Tabla Correccion de Comparables CLASE Grabada Moc
      size: 58880
      type: "application/vnd.ms-excel"
      webkitRelativePath: ""
    ► [[Prototype]]: File
    length: 1
    ► [[Prototype]]: FileList
    firstChild: null
    firstElementChild: null

```

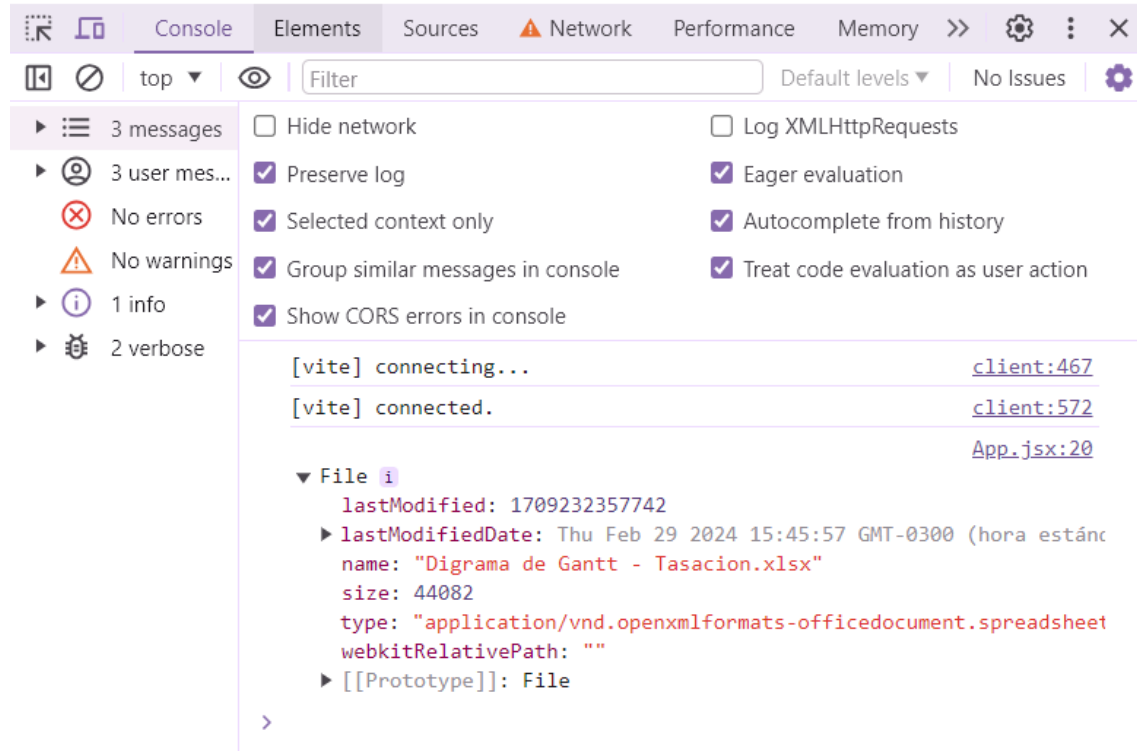
```
Ahora configuramos <input type='file' name='' id='' onChange={e =>
console.log(e.target.files)}>
```

El e trae todo target y files

```
▶ FileList {0: File, length: 1}
```

De toda la lista solo necesito la primera posición entonces configuro:

`e.target.files[0]`



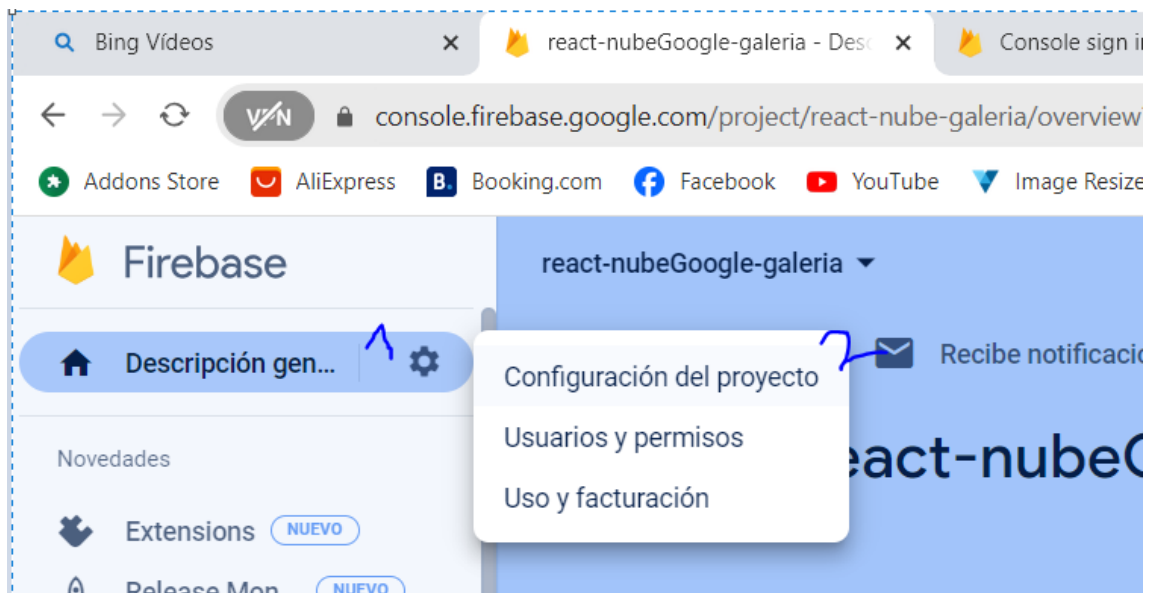
3ro ahora mmos conectamos a Firebase

Voy a mi backend y nos aparece la siguiente pantalla:

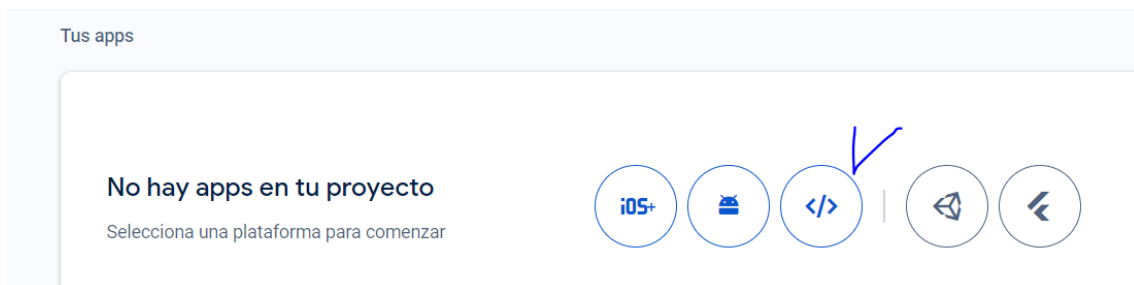
<https://console.firebase.google.com/project/react-nube-galeria/overview?hl=es-419>



## Setting



## Me conecto a WEB



## Le asigno un nombre

## × Agrega Firebase a tu aplicación web

### 1 Registrar app

Sobrenombre de la app ?

react-firebase-upload

☐

Además, configura **Firebase Hosting** para esta app.

[Más información](#) 

Hosting también se puede configurar más adelante. Puedes comenzar cuando quieras sin costo alguno.

Registrar app



### 2 Agrega el SDK de Firebase

De esta forma creo un proyecto y me da los pasos para poder dar vida a esta App y empiezo por la consola/terminal en cmd y tipeo npm i firebase

Lo siguiente es configurar creo una carpeta dentro de src que se llama firebase y dentro de la carpeta generar dos archivos config.js y pego todo el contenido que me brinda firebase

```
// 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: "AIzaSyCTTE2p1pSkZLMu_VY9WvYvQEAjd1W8QJI",
  authDomain: "react-nube-galeria.firebaseio.com",
  projectId: "react-nube-galeria",
```

```

storageBucket: "react-nube-galeria.appspot.com",

messagingSenderId: "564812513872",

appId: "1:564812513872:web:6813ddc3740dccbc06c755"

};

```

```
// Initialize Firebase
```

```
const app = initializeApp(firebaseConfig);
```

```
//-----ahora busco el servicio de firebase
```

Vuelvo a la configuración del proyecto

<https://console.firebase.google.com/project/react-nube-galeria/settings/general/web:Y2MxMTc0MDItZGI2Yy00ODRILWFjNTktMDdjMDkxZjg5NWQw?hl=es-419>

The screenshot shows the Firebase console interface. On the left is a sidebar with navigation options like 'Descripción gen...', 'Novedades', 'Extensiones', 'Release Mon...', and 'Categorías de producto'. The main area is titled 'Configuración de proyecto' and includes tabs for 'General', 'Cloud Messaging', 'Integraciones', 'Cuentas de servicio', 'Privacidad de los datos', and 'Usuarios y permisos'. Under the 'General' tab, there's a section 'Tu proyecto' containing a table of project details:

Nombre del proyecto	react-nubeGoogle-galeria
ID del proyecto	react-nube-galeria
Número de proyecto	564812513872
Ubicación predeterminada de los recursos de GCP	Aún sin seleccionar
Clave de API web	No hay clave de API web para este proyecto

Below this table is the 'Entorno' (Environment) section, which includes a description of the environment parameter and a field for 'Tipo de entorno' (Environment type) set to 'Sin especificar'.

← → ↻ console.firebase.google.com/project/react-nube-galeria/storage?hl=es-419

Addons Store AliExpress Booking.com Facebook YouTube Image Resizer - Fre... Photopea | Online... Banco de Imágenes...

**Firebase**

react-nubeGoogle-galeria ▾

Descripción gen... ⚙

Accesos directos a proyectos

**Storage**

Novedades

Extensions **NUEVO**

Release Mon... **NUEVO**

Categorías de producto

Compilación ▾

Lanzamiento y supervisión ▾

Analytics ▾

# Storage

Almacena y recupera archivos generados por el usuario (como imágenes, audio y video) sin necesidad de usar código de servidor

**Comenzar**

## Configura Cloud Storage

- 1 Reglas de seguridad para Cloud Storage
- 2 Configura la ubicación de Cloud Storage

Después de definir la estructura de tus datos, **debes crear reglas para protegerlos.**  
[Más información](#)

☐ Iniciar en **modo de producción**

Tus datos son privados de forma predeterminada. El acceso de lectura/escritura de los clientes solo se otorgará como se indica en tus reglas de seguridad.

☒ Comenzar en **modo de prueba**

Para permitir una configuración rápida, los datos se abren de forma predeterminada. Sin embargo, debes actualizar las reglas de seguridad dentro de 30 días a fin de habilitar el acceso de lectura/escritura a largo plazo para los clientes.

```
rules_version = '2';

service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if
        request.time < timestamp.date(2024, 4, 1);
    }
  }
}
```

**!** Las reglas de seguridad predeterminadas del modo de prueba permiten que cualquier usuario con acceso a tu referencia de bucket de almacenamiento pueda ver, editar y borrar todos los datos que este contenga durante los siguientes 30 días.

Cancelar **Siguiente** ✓



### Configura Cloud Storage

✓ Reglas de seguridad para Cloud Storage

2 Configura la ubicación de Cloud Storage

El parámetro de configuración de la ubicación determina en qué lugar se almacenarán tu bucket de Cloud Storage predeterminado y sus datos.

⚠ No podrás cambiar la ubicación después de configurarla. Este parámetro de configuración de ubicación también será el predeterminado para Cloud Firestore.

Más información [↗](#)

Ubicación de Cloud Storage

nam5 (us-central) ▼

Los clientes del plan Blaze pueden elegir otras ubicaciones para los buckets adicionales [haz clic aquí para actualizar](#)

Cancelar **Listo** ✓

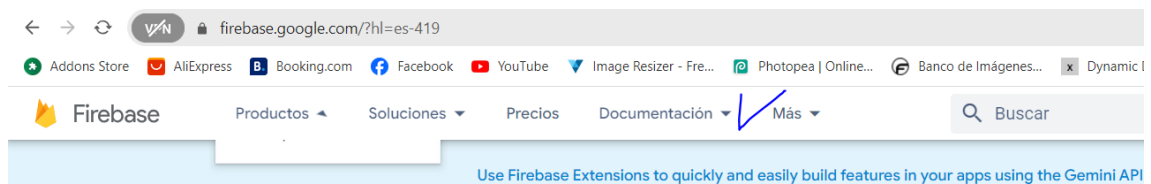
En el archivo config.js agrego el import y de esta forma nos conectamos al backend

```
import {getStorage} from 'firebase/storage';
```

y es la ruta donde exporto los archivos

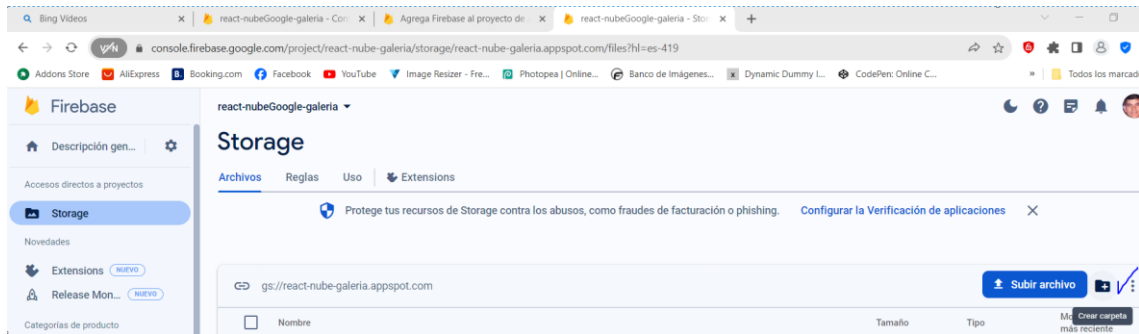
```
// Initialize Firebase
const app = initializeApp(firebaseConfig);
export const storage = getStorage(app)
```

Ahora voy a la documentación y consulto como subir archivos



Y leo un poco la parte web-> subir archivos

Voy a storage



Y creo una carpeta ejemplo Avatar

```
import {getStorage, ref} from 'firebase/storage'
```

Y agrego una función como sale en la documentación:

```
18 function uploadFile (file){
19     const storageRef = ref(storage)
20 }
```

Para subir los archivos agrego la función uploadByte que también lo importo

```
import {getStorage, ref, uploadBytes} from 'firebase/storage';
```

Ahora completo la función y hago un console.log para ver que trae:

```
export function uploadFile (file){
    const storageRef = ref(storage)
    uploadBytes(storageRef, file).then(snapshot => {
        console.log(snapshot);
    })
}
```

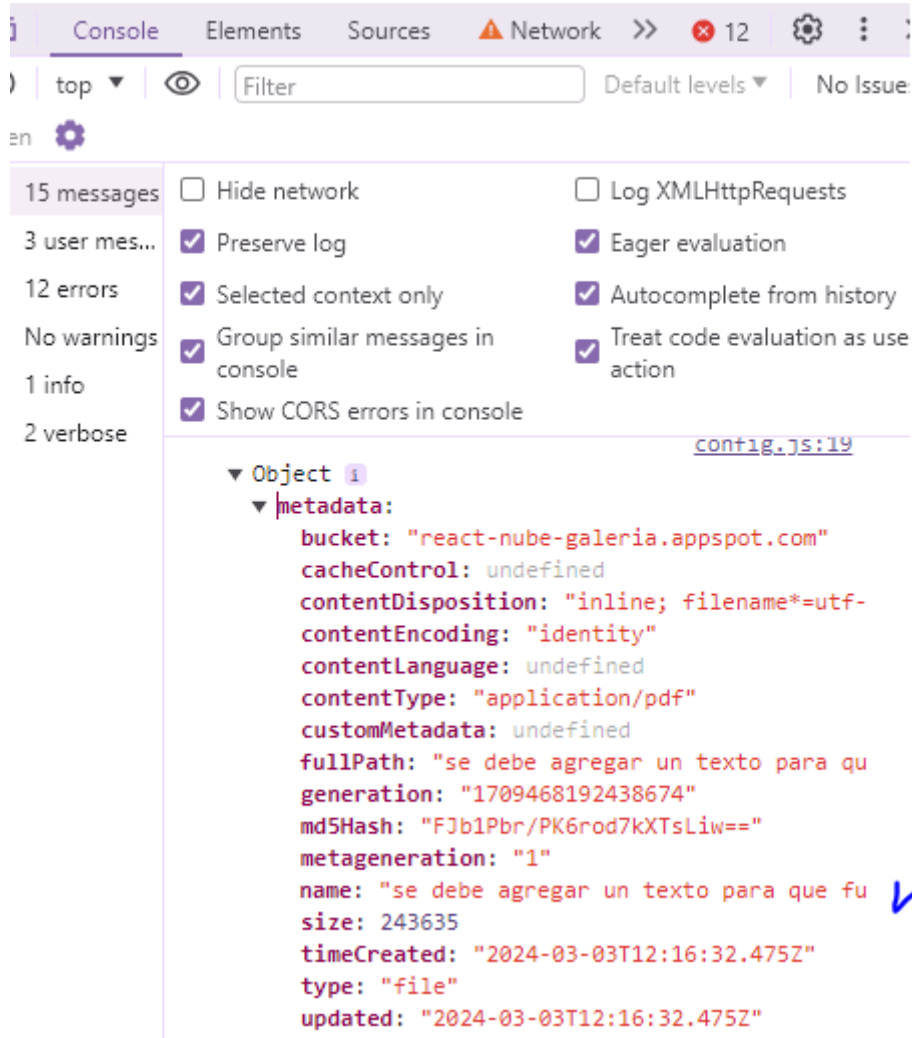
Y desde el App.jsx lo importo al config.js

```
import { uploadFile } from './firebase/config.js';

function App() {
    return (
        <div>
            <input type='file' name='' id='' onChange={e =>
uploadFile(e.target.files[0])}/>
            </div>
        </>
    )
}
```

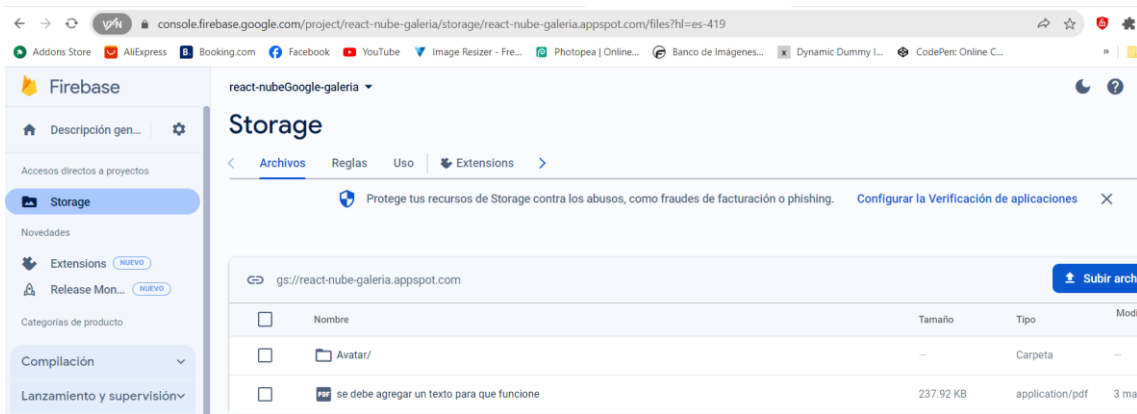
```
export default App
```

y por consola.log se ve:



Si voy a storage y refresco me muestra:

<https://console.firebase.google.com/project/react-nube-galeria/storage/react-nube-galeria.appspot.com/files?hl=es-419>



Para que no se pisen las imágenes le agrego un id, y para generarlos instalo la librería uuid

Instalo desde <https://www.npmjs.com/package/uuid>

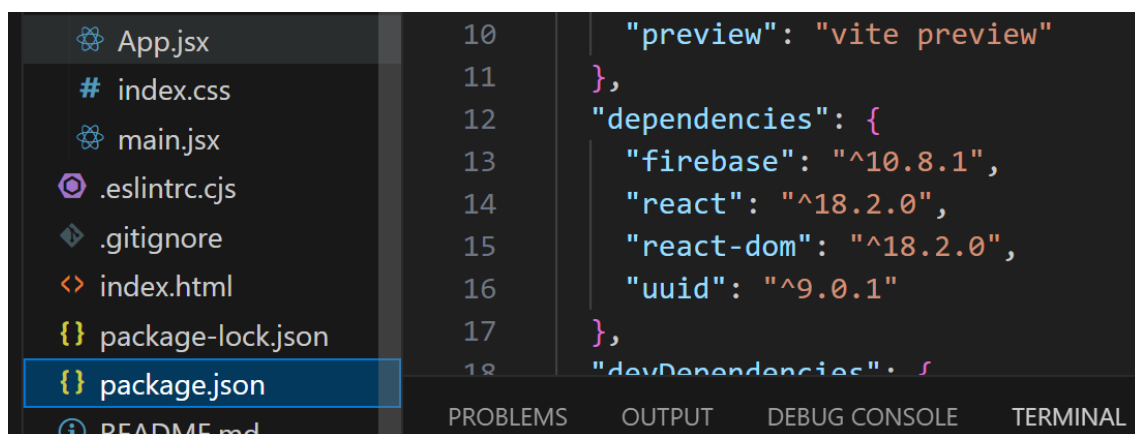
npm i uuid

el tipo de string de id que genera es:

## 2. Create a UUID (ES6 module syntax)

```
import { v4 as uuidv4 } from 'uuid';
uuidv4(); // => '9b1deb4d-3b7d-4bad-9bdd-2b0d7b3dcb6d'
```

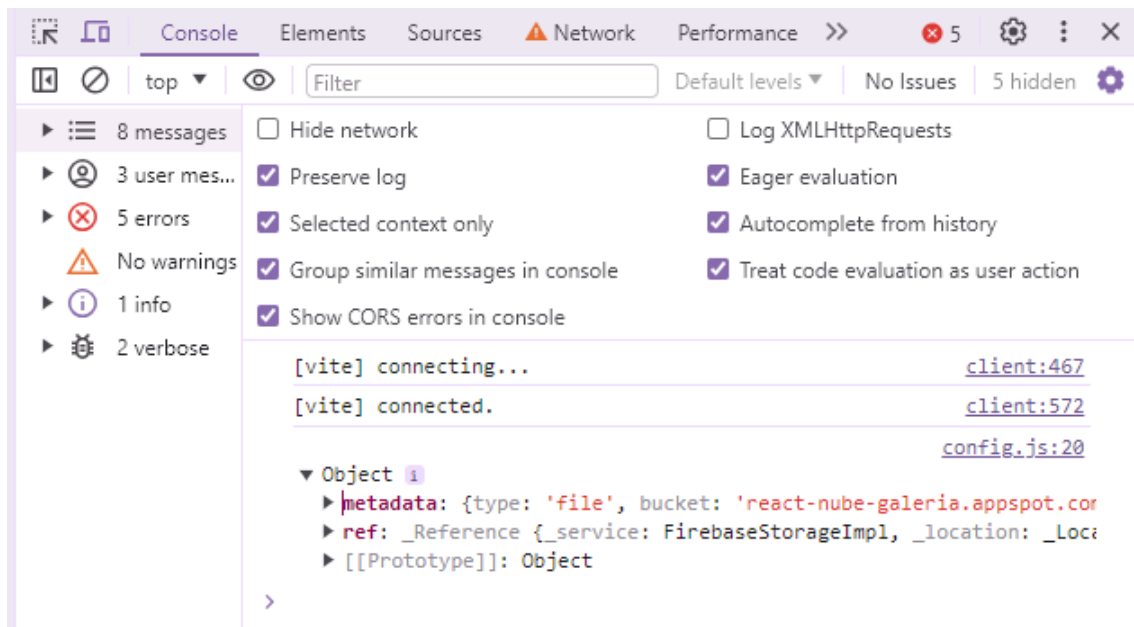
verifico en el json que estén las dependencias:



El config.js nos queda:

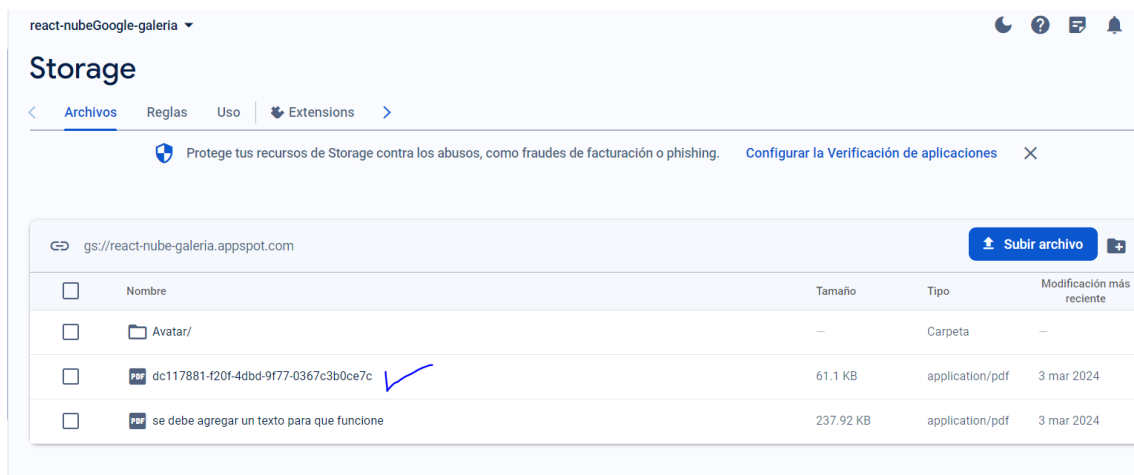
```
src > firebase > JS config.js > uploadFile
16
17 export function uploadFile (file){
18     const storageRef = ref(storage,v4())
19     uploadBytes(storageRef, file).then(snapshot => {
20         console.log(snapshot)
21     })
22 }
```

Y en el console del navegador nos muestra:



Y en mi storage:

Vemos que me genero el id con la librería v4



Otra forma de hacer este circuito es con un estado el useState en vez del input file:

Y para esto agrego un formulario con un botón mas una función , donde las funciones siempre se prueban con un console.log para saber si se disparan

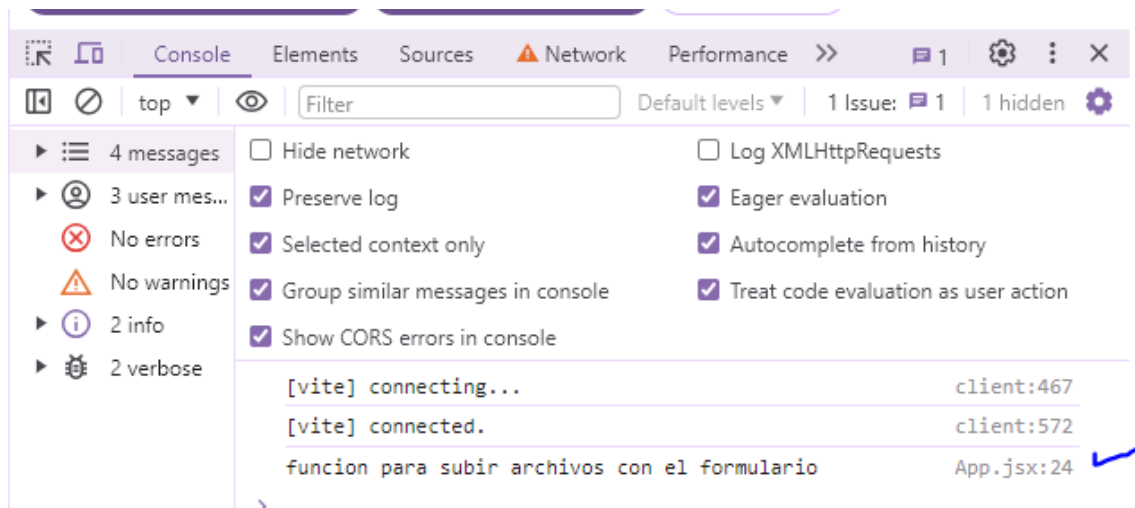
Por el cual el App.jsx queda:

```
import { useState } from 'react';
import { uploadFile } from './firebase/config.js';

function App() {
  const [file, setFile] = useState(null)
  const handleSubmit = () => {
    console.log('funcion para subir archivos con el formulario')
  }

  return (
    <>
      <form onSubmit={handleSubmit}>
        <input type='file' name='' id='' onChange={e =>
setFile(e.target.files[0])} />
        <button>UpLoad</button>
      </form>
    </>
  )
}
export default App
```

y pruebo desde el navegador:

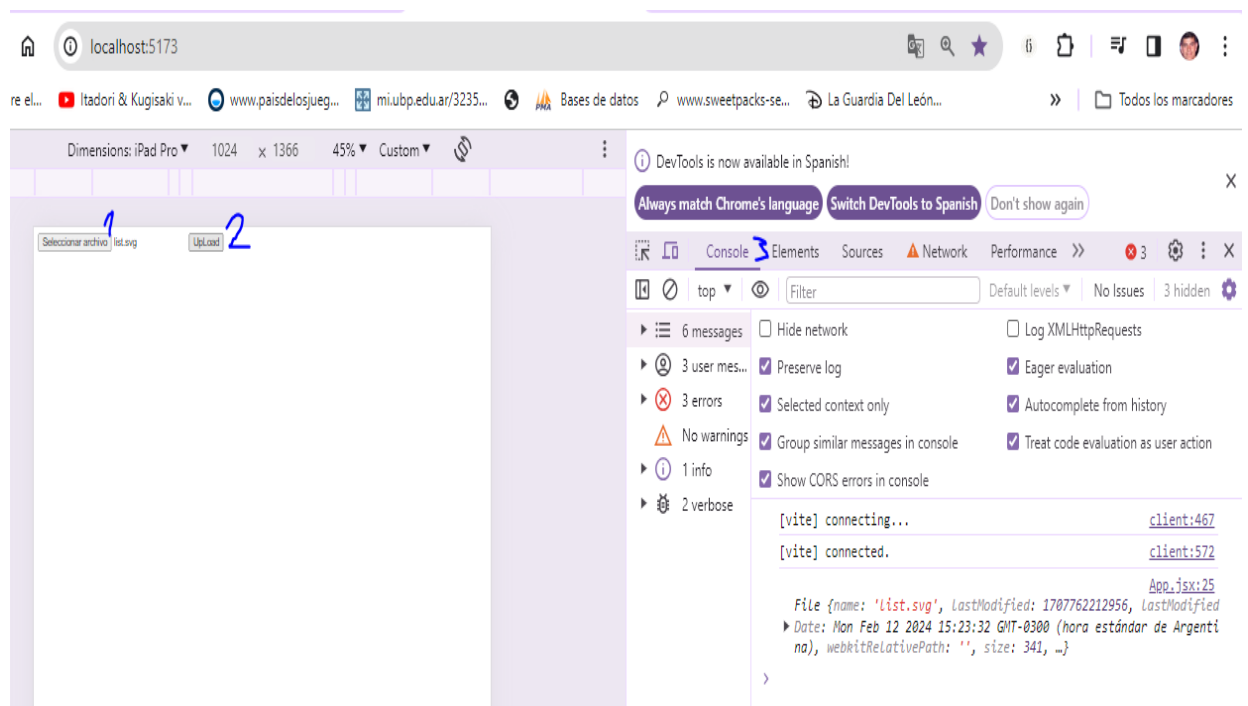


Agregamos el e.preventDefault para que el formulario no refresque, ahora pruebo en el navegador, donde completamos un poco mas el App.jsx::

```
import { useState } from 'react';
import { uploadFile } from './firebase/config.js';
```

```
function App() {
  const [file, setFile] = useState(null);
  const handleSubmit = (e) => {
    e.preventDefault();
    console.log(file)
  }
  return (
    <>
      <form onSubmit={handleSubmit}>
        <input type='file' name='' id='' onChange={(e)=>
setFile(e.target.files[0])} />
        <button>UpLoad</button>
      </form>
    </>
  )
}
export default App
```

ahora pruebo en el navegador:



Ahora en la función handleSubmit agrego la sentencia

uploadFile(file) que se acciona desde el submit del formulario

refresco el storage y vemos que sube los archivos con éxito.

Ahora buscamos que nos informe cuando falla para esto usamos la promesa del then y como es asíncrona y al uploadFile lo almaceno en una variable result desde el App:

Entonces el App queda:

```
import { useState } from 'react';
import { uploadFile } from './firebase/config.js';

function App() {
  const [file, setFile] = useState(null);

  const handleSubmit = (e) => {
    e.preventDefault();
    const result = uploadFile(file)
    console.log(result)
  }

  return (
    <>
      <form onSubmit={handleSubmit}>
        <input type='file' name='' id='' onChange={(e)=>
setFile(e.target.files[0])} />
        <button>UpLoad</button>
      </form>
    </>
  )
}
export default App
```

y el config.js queda:

```
import { initializeApp } from "firebase/app";
import { getStorage, ref, uploadBytes } from 'firebase/storage';
import { v4 } from 'uuid';

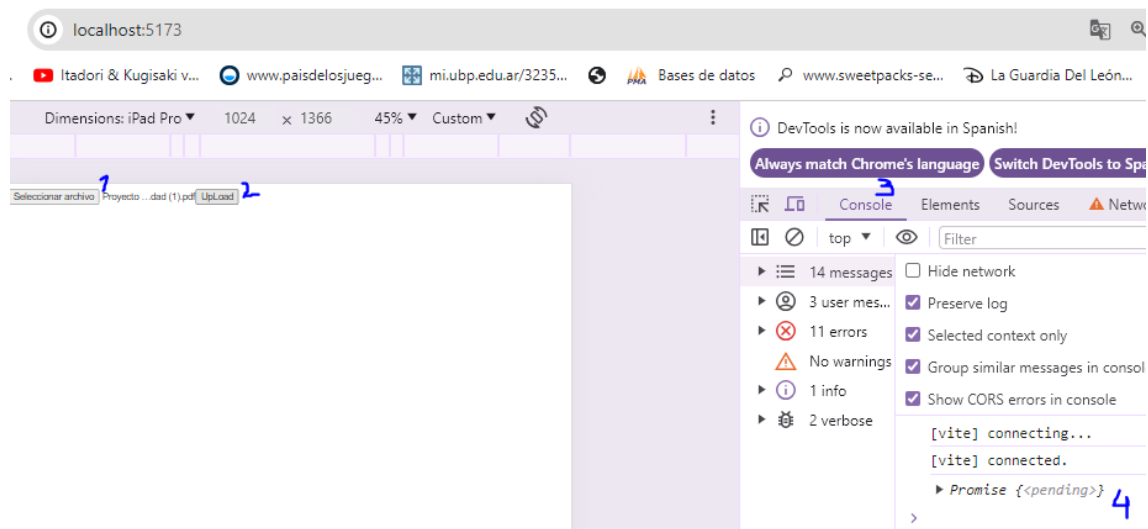
const firebaseConfig = {
  apiKey: "AIzaSyCTTE2p1pSkZLMu_VY9WvYvQEAjd1W8QJI",
  authDomain: "react-nube-galeria.firebaseio.com",
  projectId: "react-nube-galeria",
  storageBucket: "react-nube-galeria.appspot.com",
  messagingSenderId: "564812513872",
  appId: "1:564812513872:web:6813ddc3740dccbc06c755"
};

const app = initializeApp(firebaseConfig);
export const storage = getStorage(app)

export async function uploadFile (file){
  const storageRef = ref(storage,v4())
  return await uploadBytes(storageRef, file)
}
```



Y en el navegador:



En el App.jsx usamos el uploadFile y como es asíncrono:

```
const handleSubmit = async(e) => {
  e.preventDefault();
  const result = await uploadFile(file)
  console.log(result)
}
```

Y para capturar el error hago:

```
const handleSubmit = async(e) => {
  e.preventDefault();
  try{
    const result = await uploadFile(file)
    console.log(result)
  }catch(error){
    console.error(error);
  }
};
```

Como obtengo la URL , me fijo en la documentación de FireBase en el apartado Descargar archivo->descarga desde URL

En la config.js agrego la librería getDownloadURL

```
import {getStorage, ref, uploadBytes, getDownloadURL} from
'firebase/storage';
```

```
export async function uploadFile (file){
  const storageRef = ref(storage,v4())
  await uploadBytes(storageRef, file)
```

```
const url = await getDownloadURL(storageRef)
return url
}
```

Lo pruebo en el navegador:

