

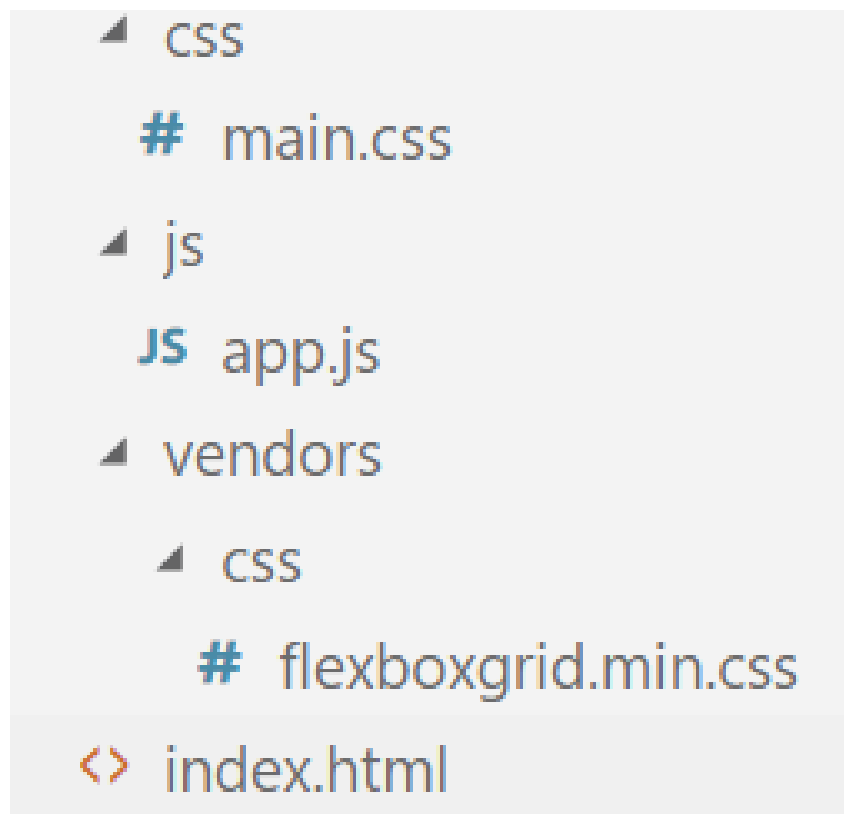
Chat Firebase – Fundamentos básicos

Pre-requisitos

- 1- Tener nodeJS instalado
- 2- npm (se instala junto con nodeJS)
- 3- Tener un entorno de desarrollo como Visual Studio Code
- 4- Descargar el framework flexbox-grid - <http://flexboxgrid.com/>
- 5- Tener los links de jQuery y materialize css y js
- 6- Saber JavaScript, CSS3, HTML5

Comencemos

1. Debemos tener la siguiente estructura de archivos en nuestro proyecto:



2. En el index.html se debe tener ya agregado materialize, flexbox-grid y nuestro(s) archivo(s) css. También deben estar los .js como jQuery, materialize y nuestro(s) archivo(s) js.

3. Vamos a crear nuestra cuenta de firebase en la pág <https://firebase.google.com> y hacemos click en GET STARTED, donde iniciaremos sesión con nuestro correo electrónico.

Firestore helps you build better mobile apps and grow your business.

[GET STARTED](#) [▶ WATCH THE VIDEO](#)

Build apps fast, without managing infrastructure

Firestore gives you functionality like analytics, databases, messaging and crash reporting so you can move quickly and focus on your users.

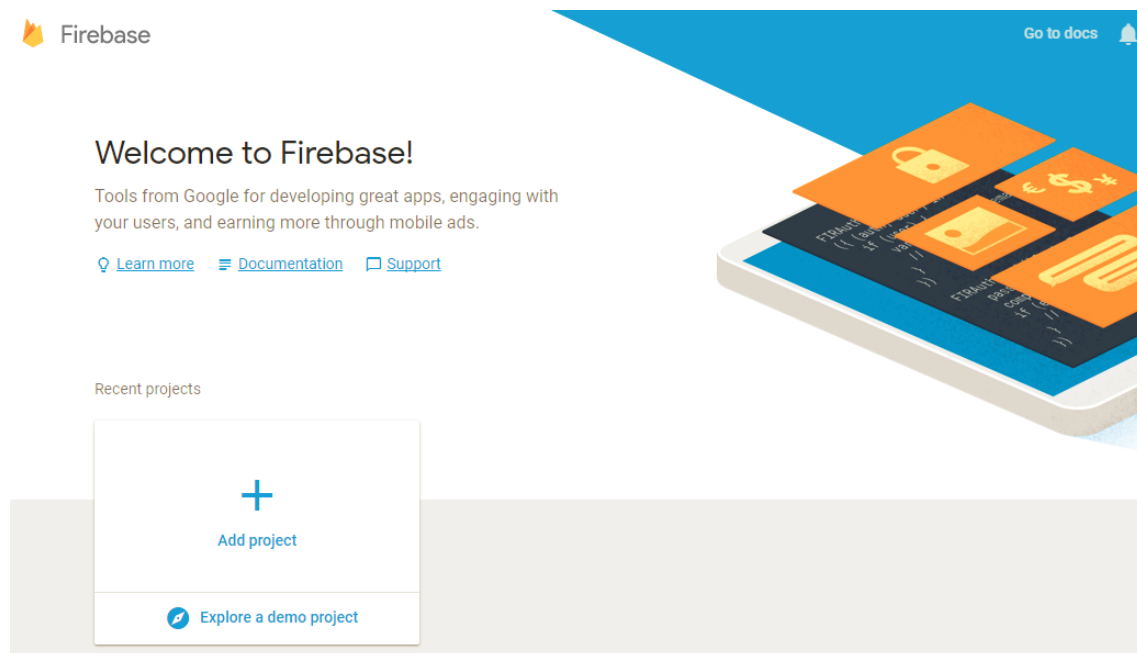
Backed by Google, trusted by top apps

Firestore is built on Google infrastructure and scales automatically, for even the largest apps.

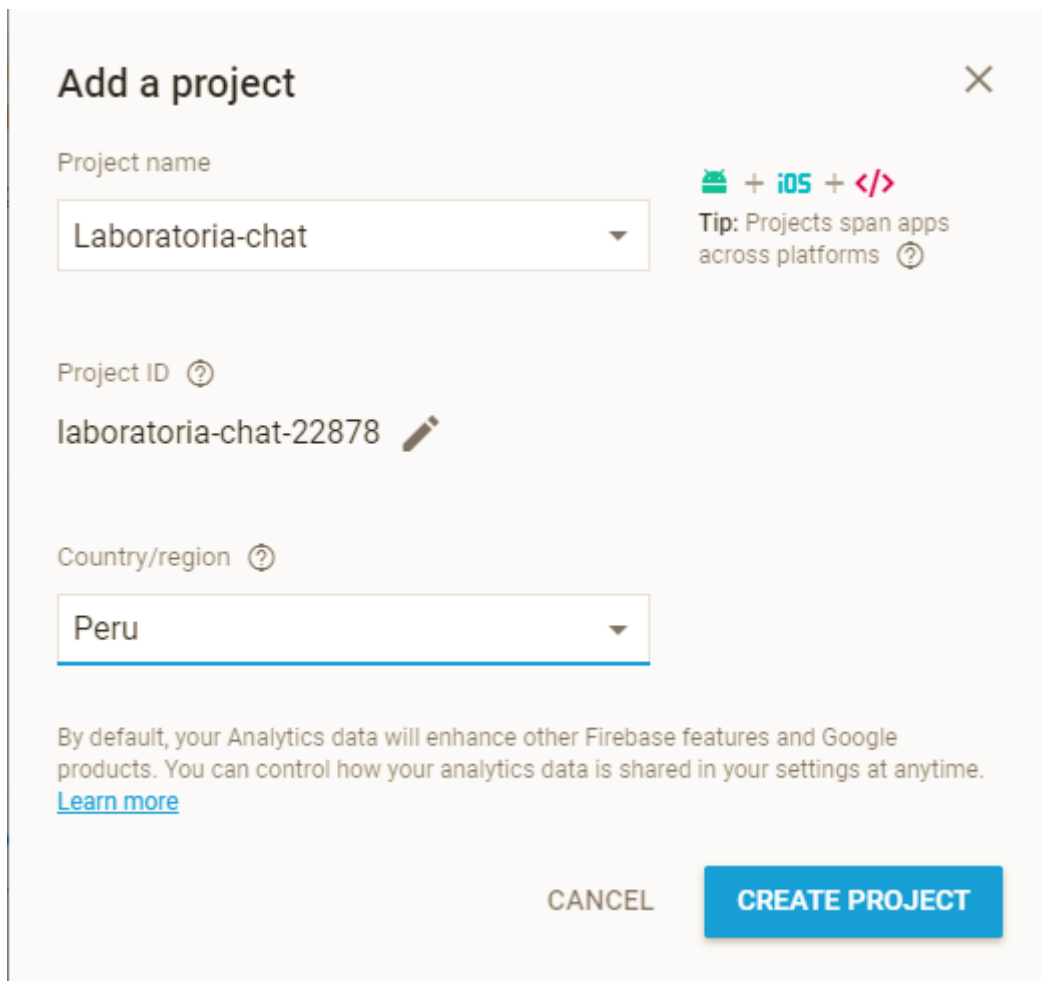
One console, with products that work together

Firestore products work great individually but share data and insights, so they work even better together.

Entraremos en <https://console.firebase.google.com/> donde vamos a hacer click en Add project



4. Ingresamos los datos para el proyecto, y hacemos click en **CREATE PROJECT**



The screenshot shows the 'Add a project' dialog box in the Firebase console. It has a title bar with a close button (X). The form contains three main sections: 'Project name' with a dropdown menu showing 'Laboratoria-chat'; 'Project ID' with a text field showing 'laboratoria-chat-22878' and an edit icon; and 'Country/region' with a dropdown menu showing 'Peru'. To the right of the 'Project name' field, there are icons for Android, iOS, and a code icon, along with a tip: 'Tip: Projects span apps across platforms'. At the bottom, there are two buttons: 'CANCEL' and 'CREATE PROJECT'.

Add a project

Project name

Laboratoria-chat

Project ID

laboratoria-chat-22878

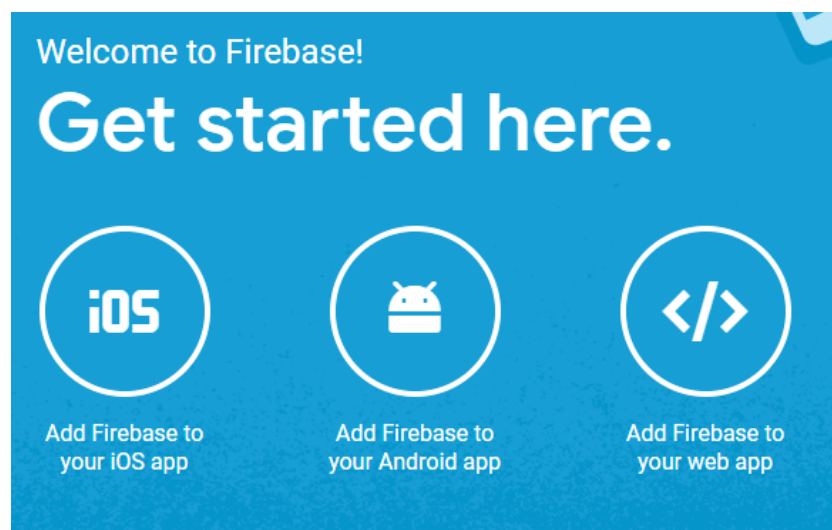
Country/region

Peru

By default, your Analytics data will enhance other Firebase features and Google products. You can control how your analytics data is shared in your settings at anytime. [Learn more](#)

CANCEL CREATE PROJECT

5. Ahora, vamos a agregar nuestra configuración de firebase, hacemos click en: **Add Firebase to your web app**



6. Y copiamos la configuración para que nuestra app pueda establecer comunicación con firebase, estos datos son únicos por cada configuración por ese motivo están tachados en la imagen de referencia.

Add Firebase to your web app

Copy and paste the snippet below at the bottom of your HTML, before other `script` tags.

```
<script src="https://www.gstatic.com/firebasejs/4.8.1/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyBwK1n7F1jPaGMMUd4F55WUw",
    authDomain: "laboratoria-chat-22878.firebaseio.com",
    databaseURL: "https://laboratoria-chat-22878.firebaseio.com",
    projectId: "laboratoria-chat-22878",
    storageBucket: "laboratoria-chat-22878.appspot.com",
    messagingSenderId: "555912436225"
  };
  firebase.initializeApp(config);
</script>
```

COPY

Check these resources to learn more about Firebase for web apps:

- [Get Started with Firebase for Web Apps](#)
- [Firebase Web SDK API Reference](#)
- [Firebase Web Samples](#)

Debemos desglosar esta estructura y adaptarla a nuestro proyecto, por lo que la línea donde importamos el firebase.js debe estar en el index.html

```
<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script src="https://www.gstatic.com/firebasejs/4.8.1/firebase.js"></script>
<!-- Compiled and minified JavaScript -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/js/materialize.min.js"></script>
<script src="./js/app.js"></script>
</body>
```

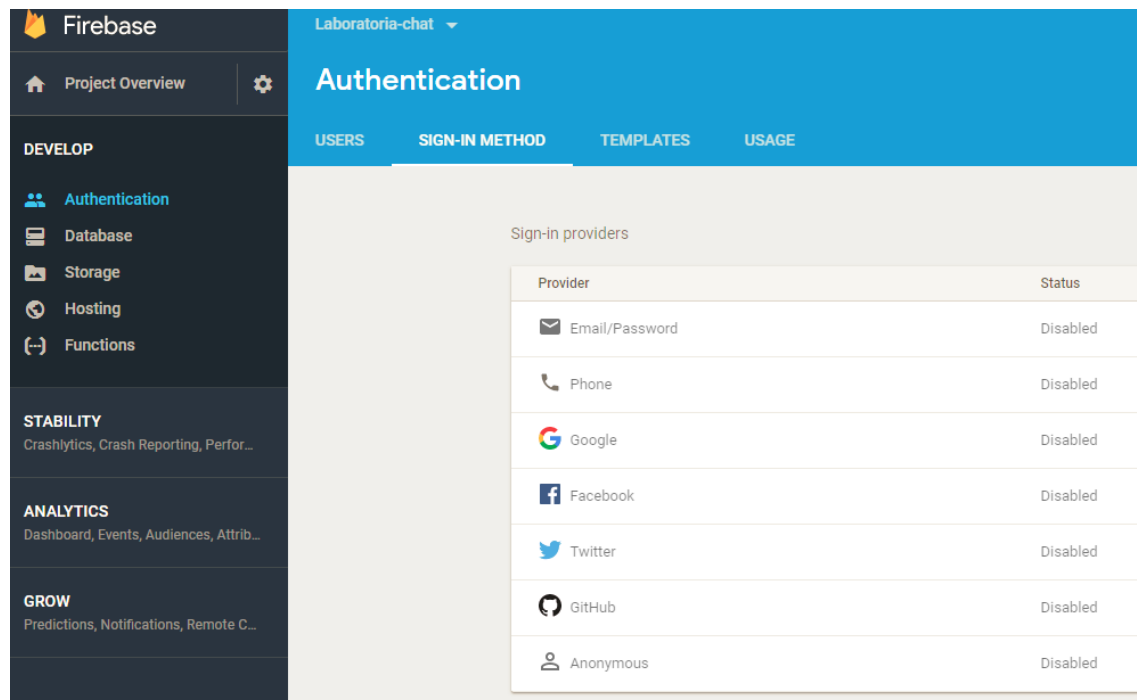
Y en el archivo app.js tendremos:

```
<> index.html JS app.js x # main.css
1 // Initialize Firebase
2 var config = {
3   apiKey: "AizaSyCY3bWzWKnzKIZ3j pACNMNesDVIRPW Kw",
4   authDomain: "laboratoria-chat-22878.firebaseio.com",
5   databaseURL: "https://laboratoria-chat-22878.firebaseio.com",
6   projectId: "laboratoria-chat-22878",
7   storageBucket: "laboratoria-chat-22878.appspot.com",
8   messagingSenderId: "566012436229"
9 };
10
11 firebase.initializeApp(config);
```

En este punto ya nuestra aplicación puede comunicarse con nuestro proyecto en firebase.

Vamos a hacer una configuración para autenticación de usuarios, usando el servicio de Google.

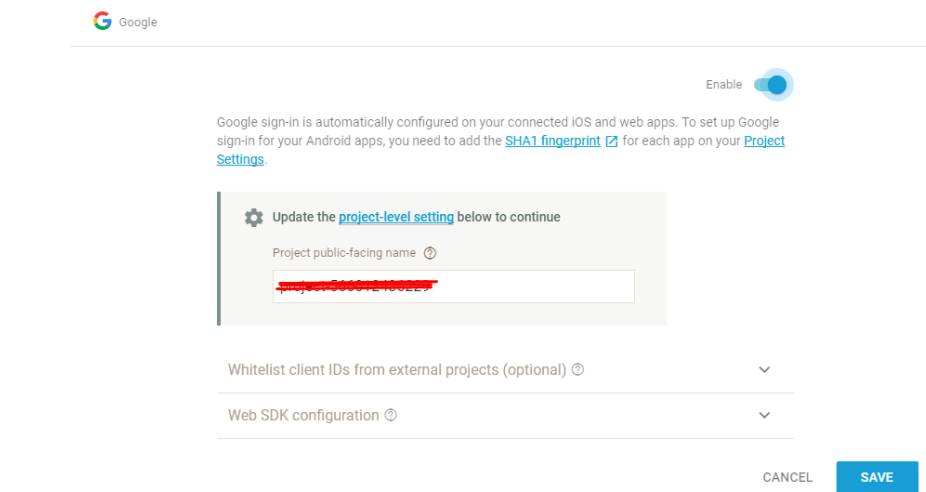
1. Vamos a dirigirnos en nuestra consola de firebase a Authentication → SIGN-IN METHOD



The screenshot shows the Firebase Authentication console for the project 'Laboratoria-chat'. The 'SIGN-IN METHOD' tab is selected, displaying a table of sign-in providers. All providers are currently disabled.

Provider	Status
Email/Password	Disabled
Phone	Disabled
Google	Disabled
Facebook	Disabled
Twitter	Disabled
GitHub	Disabled
Anonymous	Disabled

2. Vamos a activar la autenticación de nuestro proveedor Google y guardamos los cambios



Google

Enable ☒

Google sign-in is automatically configured on your connected iOS and web apps. To set up Google sign-in for your Android apps, you need to add the [SHA1 fingerprint](#) for each app on your [Project Settings](#).

Update the [project-level setting](#) below to continue

Project public-facing name ⓘ

Whitelist client IDs from external projects (optional) ⓘ

Web SDK configuration ⓘ

CANCEL SAVE

3. Ahora creamos el siguiente código para la funcionalidad en el app.js

```
13  var user = null;
14
15  var $loginBtn = $('#start-login');
16
17  $loginBtn.on('click', googleLogin);
18
19  function googleLogin() {
20    var provider = new firebase.auth.GoogleAuthProvider();
21
22    // esta en la doc de firebase
23    firebase
24      .auth()
25      .signInWithPopup(provider)
26      .then(function(result) {
27        // guardamos el usuario que nos trae result
28        user = result.user;
29        // mostramos su contenido
30        console.log(user);
31        // ocultamos el div de login
32        $('#login').fadeOut();
33      });
34  }
```

En el index.html

```
15 <body>
16   <div id="login" class="white row middle-xs center-xs">
17     <div class="col-xs-12">
18       <div class="box">
19         <p>Antes de entrar al chat, tienes que iniciar sesión</p>
20         <button id="start-login" class="btn btn-raised red white-text">
21           iniciar sesión con Google
22         </button>
23       </div>
24     </div>
25   </div>
```

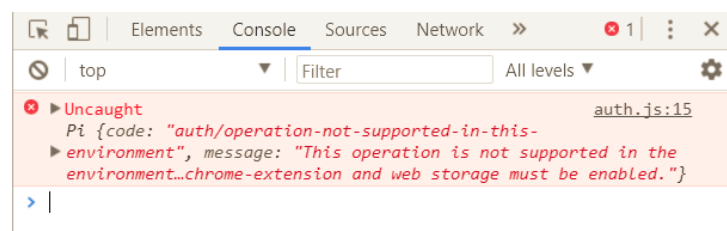
Y en el css

```
1 #login {
2   height: 100vh;
3   width: 100%;
4   position: fixed;
5   z-index: 100;
6 }
```

Ahora podemos desplegar nuestra aplicación, y hacer click en iniciar sesión con google:

Antes de entrar al chat, tienes que iniciar sesión

INICIAR SESIÓN CON GOOGLE



Error: *This operation is not supported in the environment this application is running on. "location.protocol" must be http, https or chrome-extension and web storage must be enabled.*

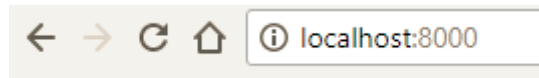
Este error surge porque debemos hacer un despliegue en un entorno diferente, al que hemos venido trabajando tradicionalmente, por lo que necesitamos un servidor local (algo parecido a lo que nos ofrece github con gh-pages).

Instalando un servidor local con nodeJS

1. En el directorio del proyecto, debemos abrir git bash
2. Y ejecutamos `$ npm install -g http-server`
3. Después de la instalación ejecutamos el servidor local con `$ http-server -p 8000`

```
$ http-server -p 8000
Starting up http-server, serving ./
Available on:
  http://[redacted]
  http://127.0.0.1:8000
Hit CTRL-C to stop the server
|
```

4. Ahora en el navegador escribimos <http://127.0.0.1:8000> o localhost:8000



Con todo esto ya tendremos funcionando el proyecto, y podremos hacer una autenticación usando Google como proveedor, la variable user definida debe contener un objeto con las siguientes propiedades (displayName, email, photoURL, etc).

Otro error:

```
Uncaught ▶ {code: "auth/operation-not-allowed", message: "The given sign-in provider is disabled for this Fi...under the sign-in method tab of the Auth section."} auth.js:15
```

Error: *The given sign-in provider is disabled for this Fi...under the sign-in method tab of the Auth section.*

Solución: Deben ir a la consola de Firebase y activar la autenticación con Google (Autentication → SIGN-IN METHOD).

Vamos por más...

Tenemos las siguientes variables ahora

```
13  var user = null;
14  var usuariosConectados = null;
15  // conexión con la base de datos
16  var database = firebase.database();
17  // llave unica por usuario-conexión
18  var conectadoKey = '';
19
20  var $loginBtn = $('#start-login');
```

Eventos

```
22  // eventos
23  $loginBtn.on('click', googleLogin);
24  $(window).on('unload', signOut); // cuando cierra la ventana se desloguea
```

Funciones

```
26  function googleLogin(){
27      var provider = new firebase.auth.GoogleAuthProvider();
28      // esta en la doc de firebase
29      firebase
30      .auth()
31      .signInWithPopup(provider)
32      .then(function(result){
33          // guardamos el usuario que nos trae result
34          user = result.user;
35          // mostramos su contenido
36          console.log(user);
37          // ocultamos el div de login
38          $('#login').fadeOut();
39
40          // si el usuario se loguea podra entrar a la app
41          initApp();
42      });
43  }
```

```

45 function initApp(){
46   // referencia a una tabla en base de datos
47   usuariosConectados = database.ref('/connected');
48
49   // 2 parametros, el 2do si no tiene displayName toma el email
50   login(user.uid, user.displayName || user.email);
51 }

53 function login(uid, name){
54   // se crea un objeto en la base de datos y se guarda la referencia
55   var conectado = usuariosConectados.push({
56     uid: uid,
57     name: name
58   });
59   // identificador unico del registro
60   conectadoKey = conectado.key;
61 }

```

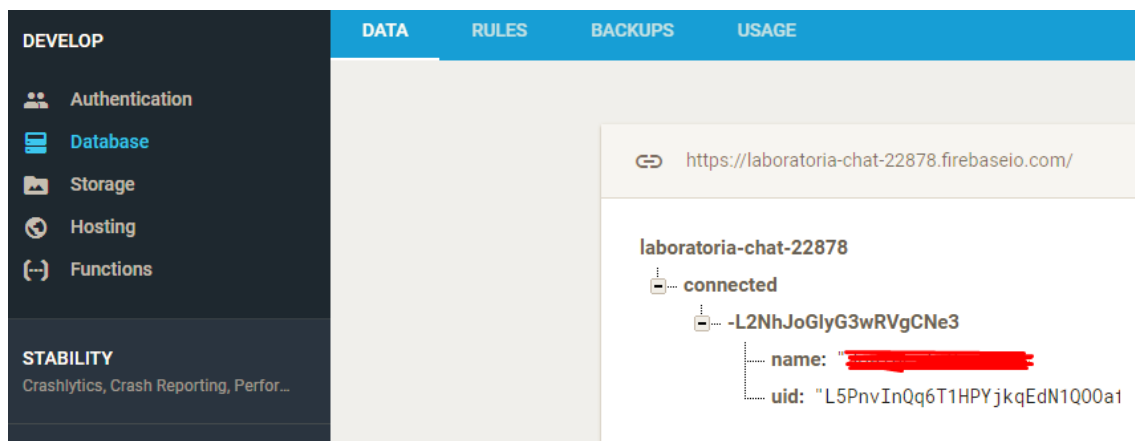
Al cerrar la pestaña se borrará el registro

```

63 function signOut(){
64   database.ref('/connected/' + conectadoKey).remove();
65 }

```

Con éste código tenemos el siguiente resultado al loguearnos:



Documentación

