

Serverless

Firebase

Sobre Firebase

— — —

- Plataforma que facilita creación aplicaciones mobile y web sin la necesidad de contar con infraestructura.
- Libera al developer de la configuración de un server para almacenamiento de datos, autenticación, escalabilidad, entre otros.

Serverless



Build better apps



Cloud Firestore BETA

Store and sync app data at global scale



ML Kit BETA

Machine learning for mobile developers



Cloud Functions BETA

Run mobile backend code without managing servers



Authentication

Authenticate users simply and securely



Hosting

Deliver web app assets with speed and security



Cloud Storage

Store and serve files at Google scale



Realtime Database

Store and sync app data in milliseconds



Improve app quality



Crashlytics

Prioritize and fix issues with powerful, realtime crash reporting



Performance Monitoring

Gain insight into your app's performance



Test Lab

Test your app on devices hosted by Google



Grow your business



Google Analytics

Get free and unlimited app analytics



Predictions BETA

Define dynamic user groups based on predicted behavior



A/B Testing BETA

Optimize your app experience through experimentation



Cloud Messaging

Send targeted messages and notifications



Remote Config

Modify your app without deploying a new version



Dynamic Links

Drive growth by using deep links with attribution



App Indexing

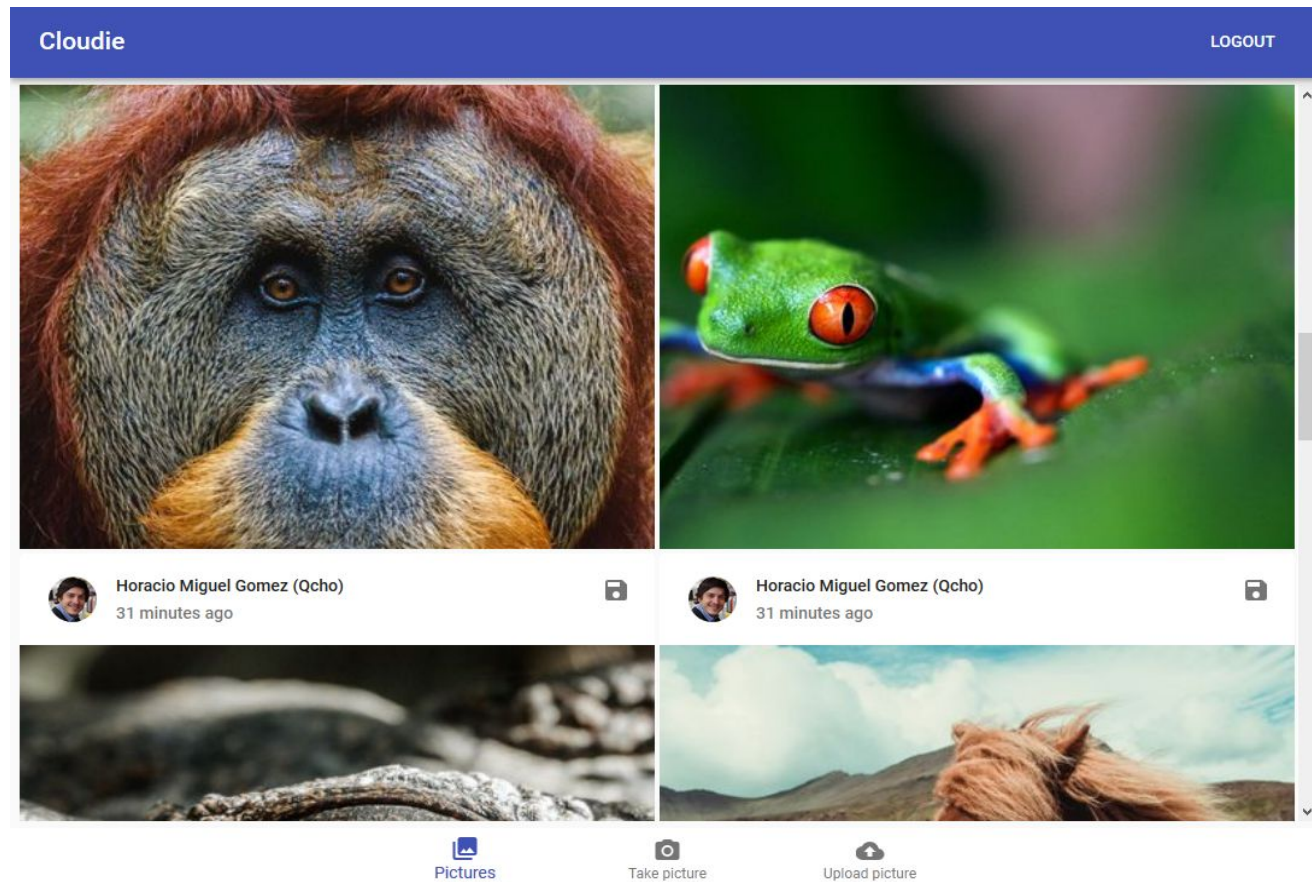
Drive search traffic to your mobile app

Cloudie

serverless-itba
.firebaseapp.com



Cloudie



Cloudie



Build better apps



Cloud Firestore BETA

Store and sync app data at global scale



ML Kit BETA

Machine learning for mobile developers



Cloud Functions BETA

Run mobile backend code without managing servers



Authentication

Authenticate users simply and securely



Hosting

Deliver web app assets with speed and security



Cloud Storage

Store and serve files at Google scale



Realtime Database

Store and sync app data in milliseconds



Improve app quality



Crashlytics

Prioritize and fix issues with powerful, realtime crash reporting



Performance Monitoring

Gain insight into your app's performance



Test Lab

Test your app on devices hosted by Google



Grow your business



Google Analytics

Get free and unlimited app analytics



Predictions BETA

Define dynamic user groups based on predicted behavior



A/B Testing BETA

Optimize your app experience through experimentation



Cloud Messaging

Send targeted messages and notifications



Remote Config

Modify your app without deploying a new version



Dynamic Links

Drive growth by using deep links with attribution



App Indexing

Drive search traffic to your mobile app

Hosting

<https://firebase.google.com/docs/hosting/>

- Single Page Application
- Progressive Web App
- 99% Código frontend (React).
- Necesitamos hostear este código estático en algún lado y tener una url disponible

— — —

Funcionalidades Clave

— — —

- SSL gratuito, incluso para custom domain.
- CDN incluido y automático.
- Discos SSD para máxima performance.
- Un sólo comando hace deploy de toda la app.
- Versionado automático de cada deploy one-click rollback.
- Subdominio by-default: **serverless-itba.firebaseapp.com**

Ciclo de desarrollo

- Implementación local.
- `firebase deploy --only=hosting`



```
Done in 57.79s.
✓ hosting: Finished running predeploy script.
i hosting: preparing public/build directory for upload...
✓ hosting: 33 files uploaded successfully
```

✓ **Deploy complete!**

Project Console: <https://console.firebase.google.com/project/serverless-itba/overview>
Hosting URL: <https://serverless-itba.firebaseio.com>
serverless-demo-itba git:(master) X

Domain	Status
serverless-itba.firebaseio.com	Default

Deployment history

Status	Time	Deploy	Files
★ Current	7 Jun 2018 07:37	qcho86@gmail.com aZjfkM	33
⬆ Deployed	7 Jun 2018 07:35	qcho86@gmail.com VyL3yn	33
⬆ Deployed	7 Jun 2018 02:58	jporsay@gmail.com WUPC2z	34
⬆ Deployed	7 Jun 2018 02:54	jporsay@gmail.com NaaAem	34
⬆ Deployed	7 Jun 2018 02:51	jporsay@gmail.com RRGNWV	34
⬆ Deployed	7 Jun 2018	jporsay@gmail.com	

Authentication

<https://firebase.google.com/docs/auth/>

- Usuario/contraseña
- Login social (Google, FB, TW)
- Registro de usuarios
- Manejo de usuarios
- Olvidé mi contraseña?



Como funciona?

— — —



- Obtener credenciales de autenticación del usuario
- Transmitir estas credenciales al SDK de Firebase Authentication
- Los servicios de backend de Firebase verifican esas credenciales y muestran una respuesta al cliente.
- Soporta 2-factor authentication.
- Tiene manejo de emails de registración, resetear contraseña, cambiar email.

Primeros pasos

— — —

- Registrar usuarios nuevos

```
firebase.auth().createUserWithEmailAndPassword(email, password).catch(function(error) {  
  // Handle Errors here.  
  var errorCode = error.code;  
  var errorMessage = error.message;  
  // ...  
});
```

Primeros pasos

— — —

- Accesos de usuarios existentes

```
firebase.auth().signInWithEmailAndPassword(email, password).catch(function(error) {  
  // Handle Errors here.  
  var errorCode = error.code;  
  var errorMessage = error.message;  
  // ...  
});
```

Primeros pasos

— — —

- Observador de estado de autenticación

```
firebase.auth().onAuthStateChanged(function(user) {  
  if (user) {  
    // User is signed in.  
    var displayName = user.displayName;  
    var email = user.email;  
    var emailVerified = user.emailVerified;  
    var photoURL = user.photoURL;  
    var isAnonymous = user.isAnonymous;  
    var uid = user.uid;  
    var providerData = user.providerData;  
    // ...  
  } else {  
    // User is signed out.  
    // ...  
  }  
});
```

Storage

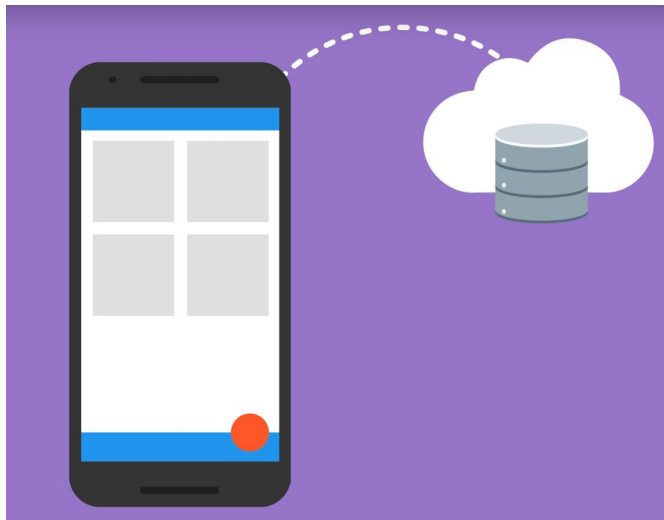
<https://firebase.google.com/docs/storage/>

- Necesitamos un lugar para almacenar las fotos que suben nuestros usuarios.
- Barato, Seguro.

— — —

Storage

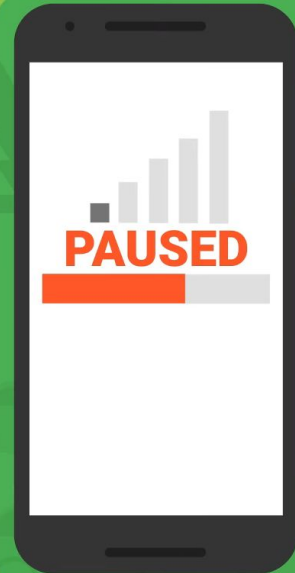
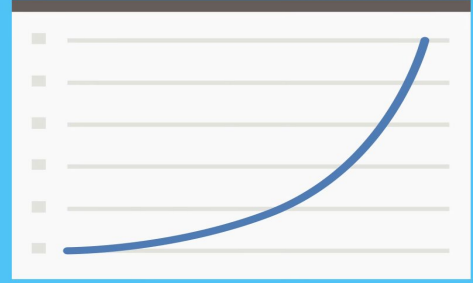
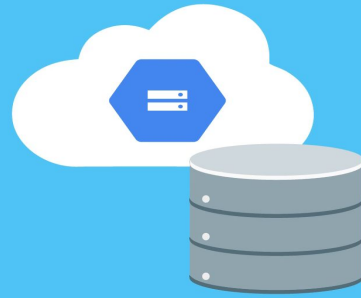
- La mayoría de las apps almacenan contenido para que sea compartido por los usuarios.
- Es un servicio de almacenamiento de objetos potente, simple y rentable.



Features

— — —

- Operaciones robustas para carga y descarga
- Seguridad
- Escalabilidad



Configuracion

- Para poder utilizar Cloud Storage:
 - consola de Firebase-> menu de Storage->pestana ARCHIVOS
 - Agregar el atributo storageBucket al objeto config:

```
// Only authenticated users can read or write to the bucket
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

Reglas

- Lenguaje de reglas declarativas que permiten definir:
 - Estructura de datos
 - Indexación de datos
 - Acceso de lectura y escritura

```
// Only authenticated users can read or write to the bucket
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

Cloud Firestore

<https://firebase.google.com/docs/firestore/>

- Necesitamos almacenar en una base de datos metadata de las imágenes para el feed

— — —

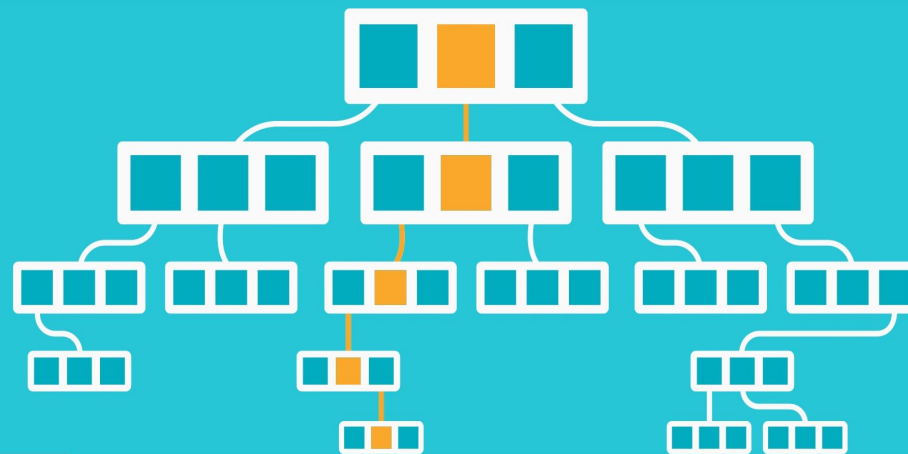
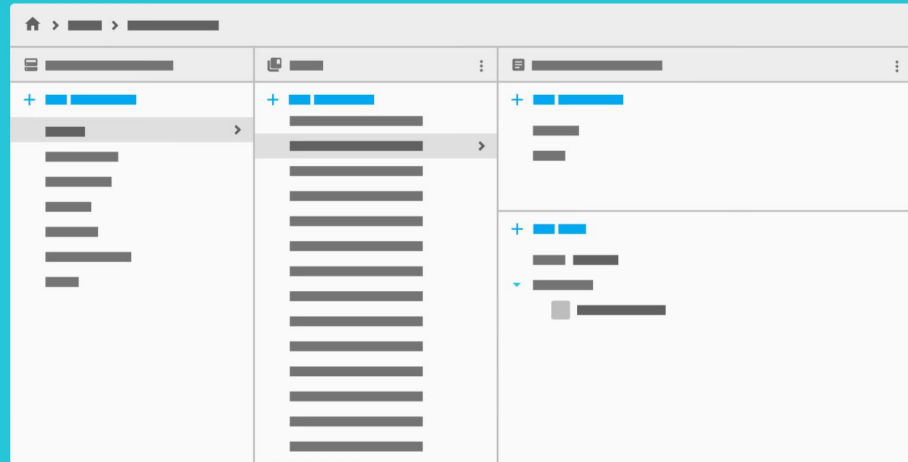
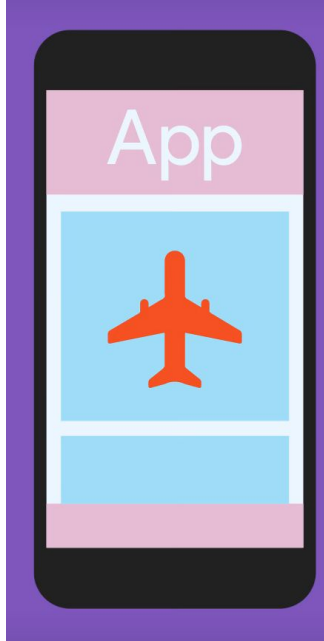
Firestore

- Es una base de datos NoSQL (de archivos) que permite:
 - Almacenar y sincronizar datos



Features

- Flexibilidad
- Consultas expresivas
- Actualizaciones en tiempo real
- Asistencia sin conexión
- Diseñado para ajustarse a escalas



Ruta de implementación

— — —

- Crear un proyecto de Cloud Firestore
- Configurar el entorno de programación: Agrega las dependencias y las bibliotecas cliente necesarias a tu app
- Inicializar una instancia de Cloud Firestore

```
firebase.initializeApp({  
  apiKey: '### FIREBASE API KEY ###',  
  authDomain: '### FIREBASE AUTH DOMAIN ###',  
  projectId: '### CLOUD FIRESTORE PROJECT ID ###'  
});
```

```
// Initialize Cloud Firestore through Firebase  
var db = firebase.firestore();
```

Ruta de implementación

— — —

- Agregar Datos

```
db.collection("users").add({
  first: "Ada",
  last: "Lovelace",
  born: 1815
})
.then(function(docRef) {
  console.log("Document written with ID: ", docRef.id);
})
.catch(function(error) {
  console.error("Error adding document: ", error);
});
```

- Leer Datos

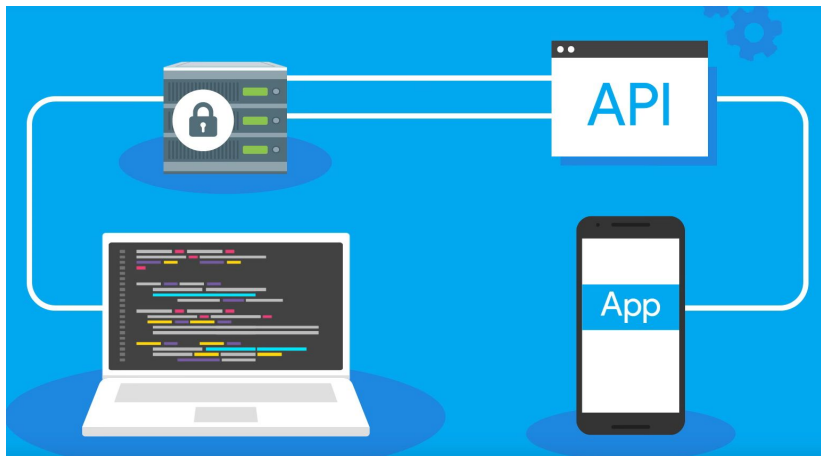
```
db.collection("users").get().then((querySnapshot) => {
  querySnapshot.forEach((doc) => {
    console.log(`${doc.id} => ${doc.data()}`);
  });
});
```


Cloud Functions

[https://firebase.google.com/
docs/functions/](https://firebase.google.com/docs/functions/)

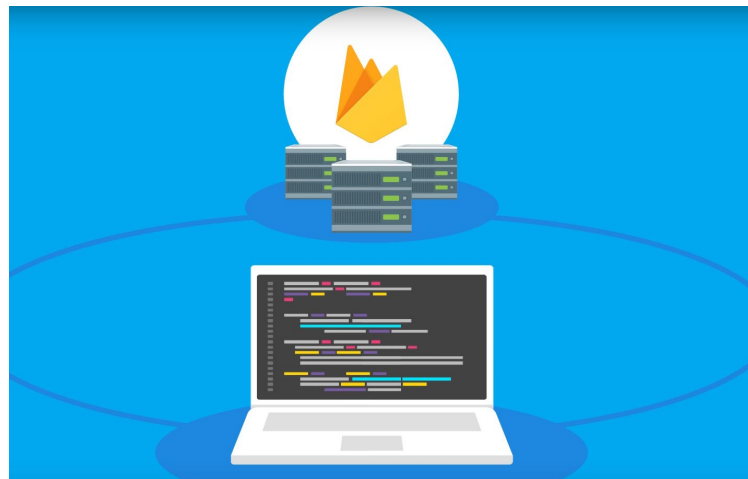
- Integración automática (triggers)
- Bajo costo mantenimiento
- Lógica segura y privada
- Scaling automático

— — —



Sin firebase

- Tenemos que implementar toda la API nosotros.
- Deploy de la API a un servidor.



Con firebase

- Firebase ofrece una API de código, para implementar y se encarga automáticamente de hacer el deploy.

Como funciona?

— — —

- Implementación en Typescript o Javascript
- Deploy a Firebase
- Triggers
 - HTTP
 - Database
 - Storage
 - Autenticación
 - Analytics
 - Crashlytics
 - Message Bus
- Transparencia de escalamiento

Ciclo de vida de una función

- Definición de trigger
- Implementación de callback
- `firebase deploy (--only=functions)`

- Event trigger
- Callback execution

```
serverless-demo-itba git:(master) X firebase deploy --only=functions
```

```
=== Deploying to 'serverless-itba'...
```

```
i deploying functions
```

```
Running command: npm --prefix "$RESOURCE_DIR" run lint
```

```
> functions@ lint /home/qcho/Code/serverless-demo-itba/functions
```

```
> tslint --project tsconfig.json
```

```
Running command: npm --prefix "$RESOURCE_DIR" run build
```

```
> functions@ build /home/qcho/Code/serverless-demo-itba/functions
```

```
> tsc
```

```
✓ functions: Finished running predeploy script.
```

```
i functions: ensuring necessary APIs are enabled...
```

```
✓ functions: all necessary APIs are enabled
```

```
i functions: preparing functions directory for uploading...
```

```
i functions: packaged functions (66.96 KB) for uploading
```

```
✓ functions: functions folder uploaded successfully
```

```
i functions: updating function generateThumbnail...
```

```
✓ functions[generateThumbnail]: Successful update operation.
```

```
✓ Deploy complete!
```

```
Project Console: https://console.firebase.google.com/project/serverless-itba/overview
```

```
serverless-demo-itba git:(master) X
```

The screenshot shows the Firebase Functions console for the project 'serverless-itba'. The 'Functions' tab is selected, displaying a table of functions. The function 'generateThumbnail' is listed with the event 'google.storage.object.finalize' and a trigger of 'serverless-itba.appspot.com'. The table also shows 1.2K executions and a median run time of 57.40 ms. The console includes navigation links for Dashboard, Health, Logs, and Usage, and a sidebar with various icons for project management.

Function	Event	Executions	Median run time
generateThumbnail...	google.storage.object.finalize serverless-itba.appspot.com	1.2K	57.40 ms

Ciclo de vida de una función

- Si la función está ocupada con muchos eventos, Google crea más instancias para controlar el trabajo con más rapidez. Si la función está inactiva, las instancias se borran.
- Cuando el programador actualiza la función mediante la implementación del código actualizado, todas las instancias de la versión antigua se borran y se reemplazan por instancias nuevas.
- Cuando un programador borra la función, se borran todas las instancias y se quita la conexión entre la función y el proveedor de eventos.

Firestore Demo

<http://serverless-itba.firebaseio.com/>

<https://console.firebase.google.com/project/serverless-itba/overview>

¿Preguntas?

Gracias