

INDICE

[1. Idea del proyecto 2](#_Toc58777560)

[2. Objetivos 2](#_Toc58777561)

[3. Tecnologías empleadas 3](#_Toc58777562)

[IDE y lenguaje de programación 3](#_Toc58777563)

[Diseño y prototipo 4](#_Toc58777564)

[Base de datos 4](#_Toc58777565)

[4. Fases del proyecto 6](#_Toc58777566)

[5. GitHub 7](#_Toc58777567)

[6. Base de datos (Firebase) 11](#_Toc58777568)

[Implementación Firebase en app Android 11](#_Toc58777569)

[Servicios de Firebase utilizados 18](#_Toc58777570)

[Firebase Authentication 18](#_Toc58777571)

[Firebase Storage 20](#_Toc58777572)

[Firebase Cloud Firestore 22](#_Toc58777573)

[Consultas en la base de datos Firebase Firestore 24](#_Toc58777574)

[Modelo de datos 25](#_Toc58777575)

[7. Casos de Uso 31](#_Toc58777576)

[8. Mejoras de la aplicación 35](#_Toc58777577)

[9. Bibliografía 36](#_Toc58777578)

## 1. Idea del proyecto

Nuestra aplicación, basada en Android, intenta hacer la vida más fácil a los padres que necesitan un cuidador/a para sus hijos. Hay muchos padres que, debido sobre todo al trabajo, no disponen de todo el tiempo que quisieran para poder cuidar a sus hijos y es una tarea muy complicada poder encontrar a alguien de confianza para que los cuide.

De ahí nace nuestra aplicación, donde los padres pueden ver los anuncios de los cuidadores/as y disponer de toda la transparencia y confianza ya que implementamos valoraciones y chat para que los padres estén seguros de elegir el mejor candidato y los niños estén en las mejores manos.

## 2. Objetivos

Uno de los objetivos primordiales de nuestra aplicación es la confianza. Por ello implementamos valoraciones para que las familias puedan calificar a los cuidadores y así sea más fácil poder elegir.

Otro objetivo es la comunicación segura y eficaz entre familias y cuidadores/as, por ello implementamos un chat para que la comunicación sea fluida y rápida. (Mejora)

Por último, un objetivo muy importante es hacer la vida más fácil a nuestros clientes. Encontrar a un cuidador/a de confianza puede ser muy difícil y por ello creamos esta plataforma para que padres y cuidadores/as puedan comunicarse de manera sencilla y transparente.

El usuario se registra en nuestra aplicación mediante un correo electrónico el cual verificamos desde Firebase. Una vez verificado el usuario tiene que crearse un perfil de familia o de canguro rellenando todos sus datos. Cuando entramos en la aplicación en el caso de ser una familia el usuario podrá ver todos los canguros que hay cerca de ella, los más valorados y los más baratos aparte de poder consultar en el mapa los canguros que tiene por la zona en la que esté en ese momento ya que accedemos a su ubicación.

En el usuario familia también puede crear un anuncio si lo desea, para que los canguros puedan contactar con ella mediante el menú desplegable introduciendo todos los datos para que el canguro sepa que necesidades tiene la familia.

El objetivo es que tanto las familias como los canguros tengan una experiencia de uso plena y gratificante, que puedan comunicarse entre familias y canguros con la mayor transparencia posible.

## 3. Tecnologías empleadas

### IDE y lenguaje de programación

Android Studio

Android Studio es el [entorno de desarrollo integrado](https://es.wikipedia.org/wiki/Ambiente_de_desarrollo_integrado) oficial para la plataforma [Android](https://es.wikipedia.org/wiki/Android). Fue anunciado el [16 de mayo](https://es.wikipedia.org/wiki/16_de_mayo) de [2013](https://es.wikipedia.org/wiki/2013) y reemplazó a [Eclipse](https://es.wikipedia.org/wiki/Eclipse_(software)) como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Desde el 7 de mayo de 2019, [Kotlin](https://es.wikipedia.org/wiki/Kotlin_(lenguaje_de_programaci%C3%B3n)) es el lenguaje preferido de Google para el desarrollo de aplicaciones de Android.[3](https://es.wikipedia.org/wiki/Android_Studio#cite_note-3)​ Aun así, Android Studio admite otros lenguajes de programación, como Java y C ++.

Java

Java es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) y una [plataforma informática](https://es.wikipedia.org/wiki/Plataforma_inform%C3%A1tica) que fue comercializada por primera vez en 1995 por [Sun Microsystems](https://es.wikipedia.org/wiki/Sun_Microsystems). Hay muchas [aplicaciones](https://es.wikipedia.org/wiki/Aplicacion) y [sitios web](https://es.wikipedia.org/wiki/Sitios_web) que no funcionarán, probablemente, a menos que tengan Java instalado y cada día se crean más.

Java es rápido, seguro y fiable. Desde [ordenadores portátiles](https://es.wikipedia.org/wiki/Ordenadores_port%C3%A1tiles) hasta centros de datos, desde [consolas para juegos](https://es.wikipedia.org/wiki/Consolas_para_juegos) hasta computadoras avanzadas, desde [teléfonos móviles](https://es.wikipedia.org/wiki/Tel%C3%A9fonos_m%C3%B3viles) hasta [Internet](https://es.wikipedia.org/wiki/Internet), Java está en todas partes, si es ejecutado en una plataforma no tiene que ser [recompilado](https://es.wikipedia.org/wiki/Compilaci%C3%B3n_en_tiempo_de_ejecuci%C3%B3n) para correr en otra.

Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de [cliente-servidor](https://es.wikipedia.org/wiki/Cliente-servidor) de [web](https://es.wikipedia.org/wiki/Web), con unos diez millones de usuarios reportados.

JSON

JSON (acrónimo de JavaScript Object Notation) es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de [objetos](https://es.wikipedia.org/wiki/Objeto_(programaci%C3%B3n)) de [JavaScript](https://es.wikipedia.org/wiki/JavaScript), aunque, debido a su amplia adopción como alternativa a [XML](https://es.wikipedia.org/wiki/XML), se considera un formato independiente del lenguaje.

### Diseño y prototipo

Adobe XD

Adobe XD es un [editor de gráficos vectoriales](https://es.wikipedia.org/wiki/Editor_de_gr%C3%A1ficos_vectoriales) desarrollado y publicado por [Adobe Inc](https://es.wikipedia.org/wiki/Adobe_Systems_Incorporated) para diseñar y crear un prototipo de la [experiencia del usuario](https://es.wikipedia.org/wiki/Experiencia_de_usuario) para páginas web y [aplicaciones móviles.](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil)

Adobe Photoshop

Adobe Photoshop es un editor de [fotografías](https://es.wikipedia.org/wiki/Fotograf%C3%ADa) desarrollado por Adobe Systems Incorporated. Usado principalmente para el [retoque](https://es.wikipedia.org/wiki/Retoque_fotogr%C3%A1fico) de [fotografías](https://es.wikipedia.org/wiki/Fotograf%C3%ADa) y gráficos.

Lo hemos utilizado para la creación de logos, imágenes de Login, de bienvenida y el menú desplegable

Material Desing

Material Design es una normativa de diseño enfocado en la visualización del sistema operativo Android, además en la web y en cualquier plataforma. Fue desarrollado por [Google](https://es.wikipedia.org/wiki/Google).

Más tarde lo explicamos con más detalle.

Lottie

Lottie es una biblioteca para Android, iOS, Web y Windows que analiza animaciones [de Adobe After Effects](http://www.adobe.com/products/aftereffects.html) exportadas como json con [Bodymovin](https://github.com/airbnb/lottie-web" \t "_blank) y las representa de forma nativa en el móvil y en la web

Más tarde lo explicamos con más detalle.

AirDroid

AirDroid es una aplicación que hace de necesario puente entre el PC y tu móvil Android para que puedas gestionarlo desde la comodidad del PC: ver tus imágenes, leer tus mensajes o incluso manejar tu teléfono de forma remota.

La utilizamos para poder controlar el móvil sin necesidad de cables ni maquinas virtuales

### Base de datos

Firebase Firestore

Cloud Firestore es la base de datos más reciente de Firebase para el desarrollo de apps para dispositivos móviles. Es una solución eficiente y de baja latencia destinada a las apps para dispositivos móviles que necesitan estados sincronizados entre los clientes en tiempo real.

Más tarde lo explicamos con más detalle.

Firebase Authentication

Firebase Authentication nos permite implementar fácilmente la autenticación en cualquier proyecto sea web o app. Firebase ofrece la posibilidad de autenticar al usuario a través de diferentes proveedores más allá de email o password.

Más tarde lo explicamos con más detalle.

Firebase Storage

Cloud Storage para Firebase es un servicio de almacenamiento de objetos potente, simple y rentable construido para la escala de Google. Los SDK de Firebase para Cloud Storage agregan la seguridad de Google a las operaciones de carga y descarga de archivos para tus apps de Firebase, sin importar la calidad de la red.

Más tarde lo explicamos con más detalle.

GitHub

Github es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, y que fue comprada por Microsoft [en junio del 2018](https://www.xataka.com/aplicaciones/oficial-microsoft-compra-github-7-500-millones-dolares).

La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.

Más tarde lo explicamos con más detalle.

## 4. Fases del proyecto

* Planificación de requisitos
* Diseño de la arquitectura de software
* Diseño de la base de datos Firebase
* Diseño de la aplicación Android
* Desarrollo de la aplicación Android
* Lanzamiento y testeo de la aplicación

## 5. GitHub

GitHub es un sistema de gestión de proyectos y control de versiones de código, así como una plataforma de red social diseñada para desarrolladores que permite trabajar en colaboración con otras personas de todo el mundo, planificar proyectos y realizar un seguimiento del trabajo.

[GitHub](https://github.com/) es también uno de los [repositorios online más grandes](https://octoverse.github.com/) de trabajo colaborativo en todo el mundo.

¿Qué es un control de versiones? Cuando los desarrolladores hacen un nuevo proyecto, siempre continúan haciéndole modificaciones al código. Incluso después de la puesta en marcha de los proyectos, todavía necesitan actualizar las versiones, corregir errores, agregar nuevas funciones, etc.

El sistema de control de versiones ayuda a registrar los cambios realizados al código. Aún más, registra quién realizó los cambios y puede restaurar el código borrado o modificado.

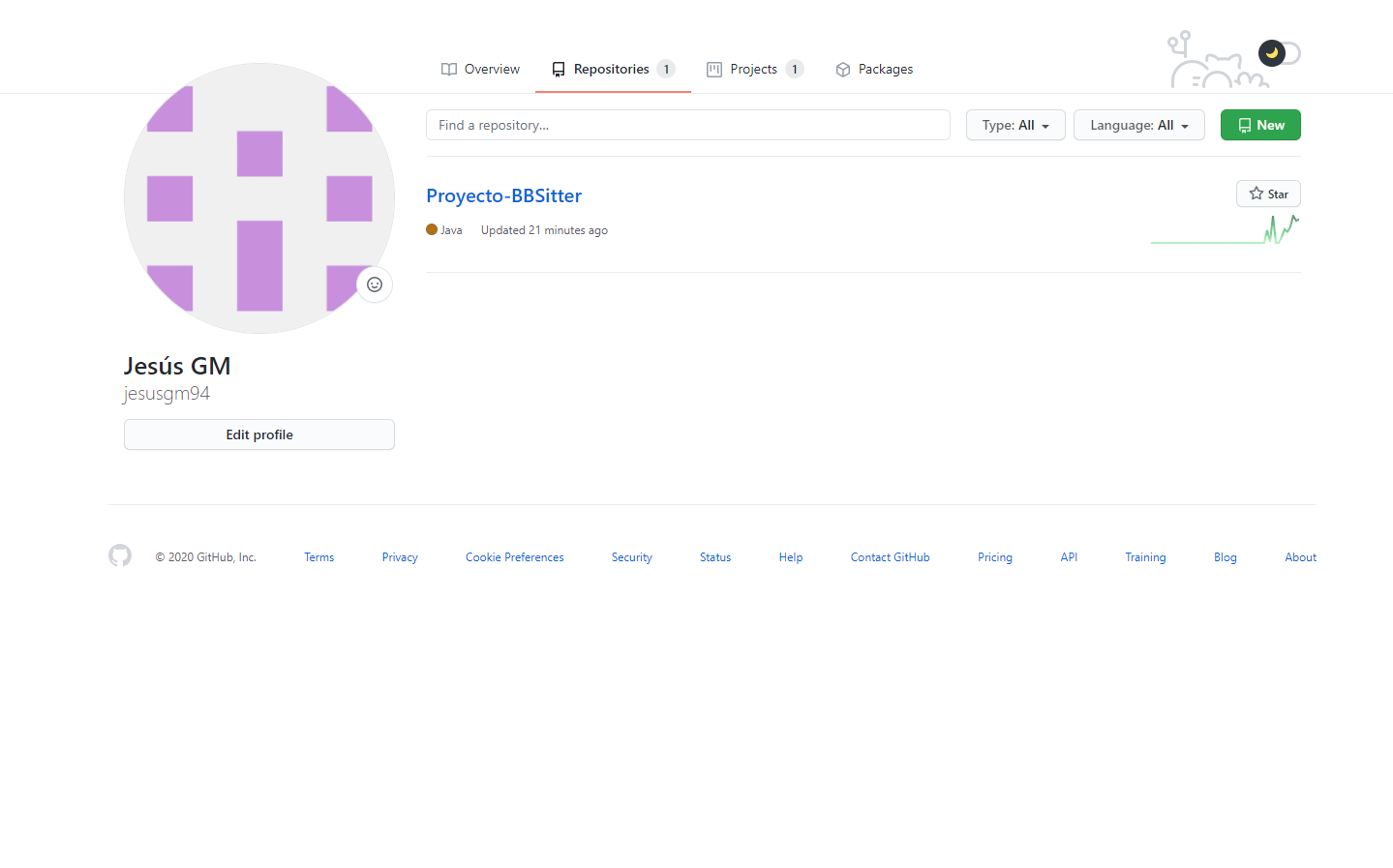
No hay códigos sobrescritos ya que Git guarda varias copias en el repositorio.

Esta parte del proyecto es la que más nos ha costado entender ya que esto no lo vimos en clase y además es un poco lioso. Finalmente entendimos el proceso de enviar datos a GitHub y de recoger datos después de vernos innumerables videos tutoriales.

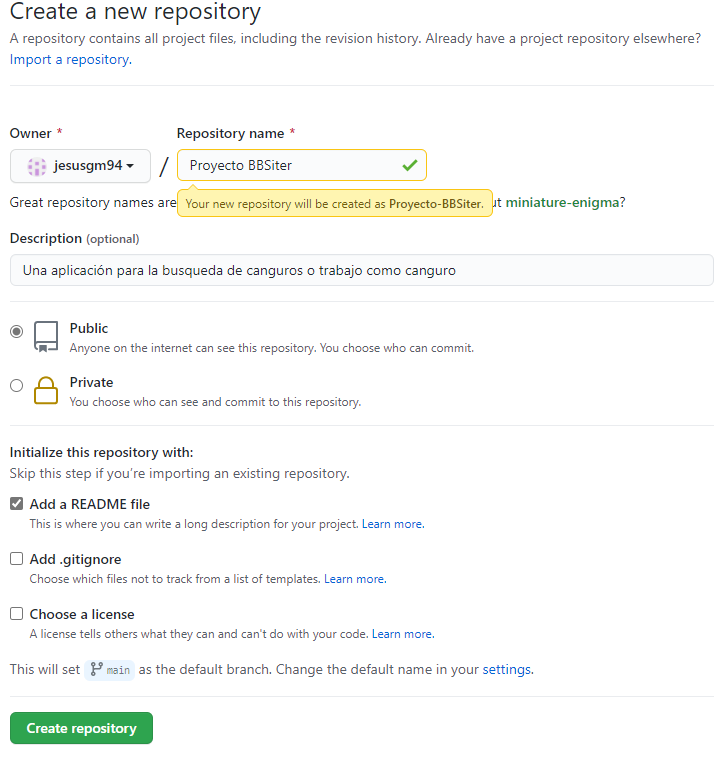
Su funcionamiento es el siguiente:

Paso 1: Crear un repositorio en GitHub.

Para ello debemos registrarnos en [GitHub.com](https://github.com/), buscar la pestaña “Repositories” y hacer clic en “New”



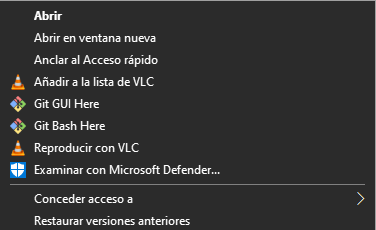
Rellenamos el nombre de nuestro repositorio, le ponemos una descripción y le damos a “Create repository” y nos saldrá en nuestra pantalla principal un repositorio vacío donde debemos rellenar con los datos de nuestra app.



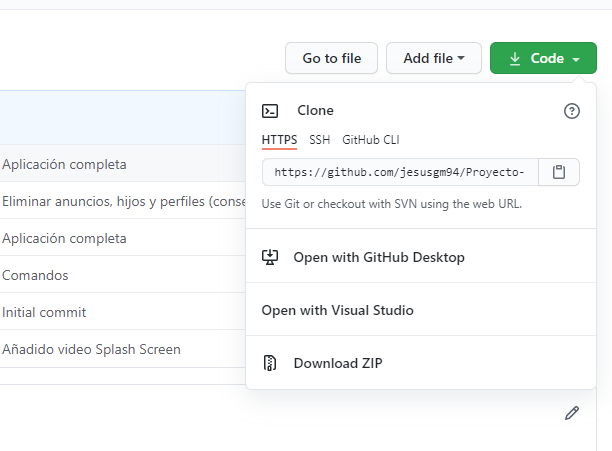
Paso 2. Clonar repositorio en nuestro pc

En este paso es necesario descargarse [Git](https://git-scm.com/downloads) mediante este enlace.

Una vez instalado en nuestro pc nos vamos a cualquier carpeta y presionamos el botón derecho para ver las opciones y seleccionamos “Git Bash Here”



Se nos abrirá la consola de Git y es el momento de clonar nuestro repositorio. Para ello necesitamos la URL de nuestro repositorio que lo encontraremos en nuestro repositorio de GitHub



En la consola de Git debemos poner el siguiente comando



Esto nos permitirá tener todo lo que haya en GitHub en nuestro pc y viceversa. Con lo cual el siguiente paso es crear nuestro proyecto de Android Studio dentro de esa carpeta para poder subirlo.

Una vez creado el proyecto de Android para poder subir el proyecto a GitHub tendrá que ser el siguiente comando



En cambio, si lo que queremos es recoger los datos de GitHub tendremos que hacer lo siguiente



## 6. Base de datos (Firebase)

Uno de los grandes retos que hemos tenido que superar ha sido la base de datos, ya que elegimos una base de datos noSQL, llamada Firebase.

Esta plataforma fue creada por Google en 2014 para el desarrollo de aplicaciones móviles y aplicaciones web. Es una plataforma ubicada en la nube, integrada con [Google Cloud Platform](https://es.wikipedia.org/wiki/Google_Cloud), que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización.

Firebase dota a sus usuarios de una gran documentación​ para crear aplicaciones usando esta plataforma. Aparte de esto, ofrece soporte gratuito mediante correo electrónico para todos sus usuarios, y además sus desarrolladores participan en plataformas como [Github](https://es.wikipedia.org/wiki/GitHub), como más tarde explicaremos.

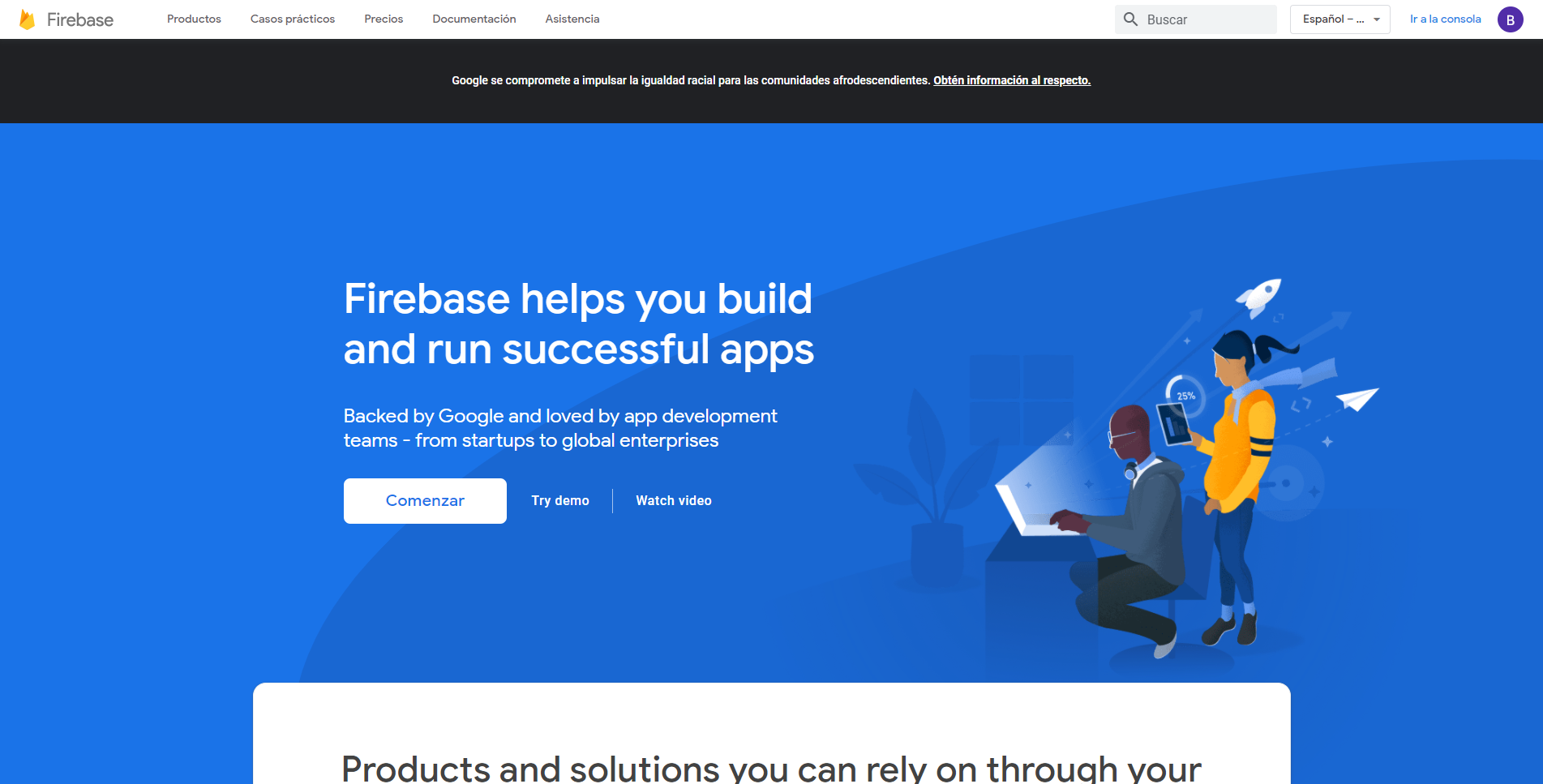
### Implementación Firebase en app Android

Para implementar Firebase en la aplicación hay que seguir unos pasos que los explicamos a continuación. Los requisitos previos son los siguientes:

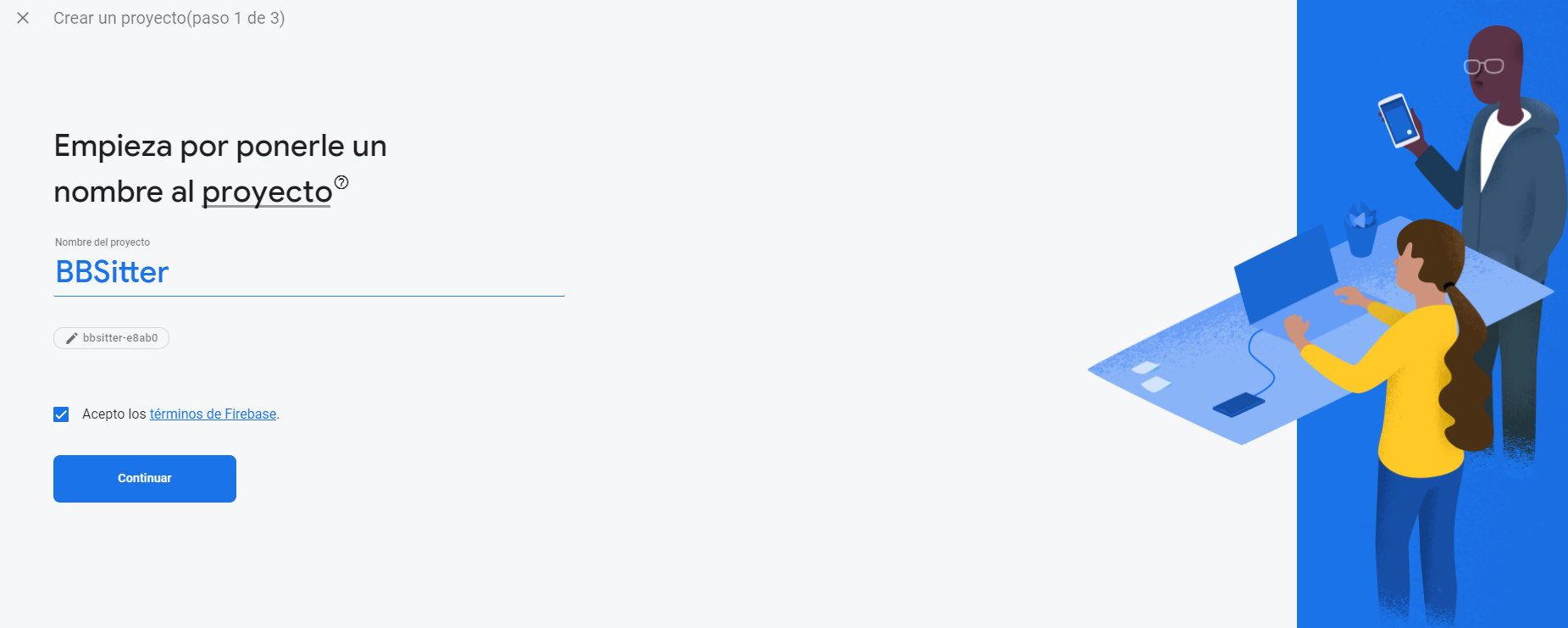
* Instalación de [Android Studio](https://developer.android.com/sdk?authuser=0) o actualizar a su versión más reciente.
* Se orienta al nivel de API 16 (Jelly Bean) o una versión posterior.
* Usa Gradle 4.1 o una versión posterior.
* Implementación de com.android.tools.build:gradle 3.2.1 o una versión posterior
* compileSdkVersion 28 o una versión posterior
* Configura un dispositivo físico o utiliza un emulador para ejecutar tu app.  
  Los [emuladores](https://developer.android.com/studio/run/managing-avds?authuser=0) deben usar una imagen que cuente con Google Play.
* [Accede a Firebase](https://console.firebase.google.com/?authuser=0) con tu Cuenta de Google.

Una vez completados los requisitos previos debemos conectar la aplicación con nuestro proyecto de Firebase. Para ello lo primeros que debemos hacer es crear un proyecto de Firebase.

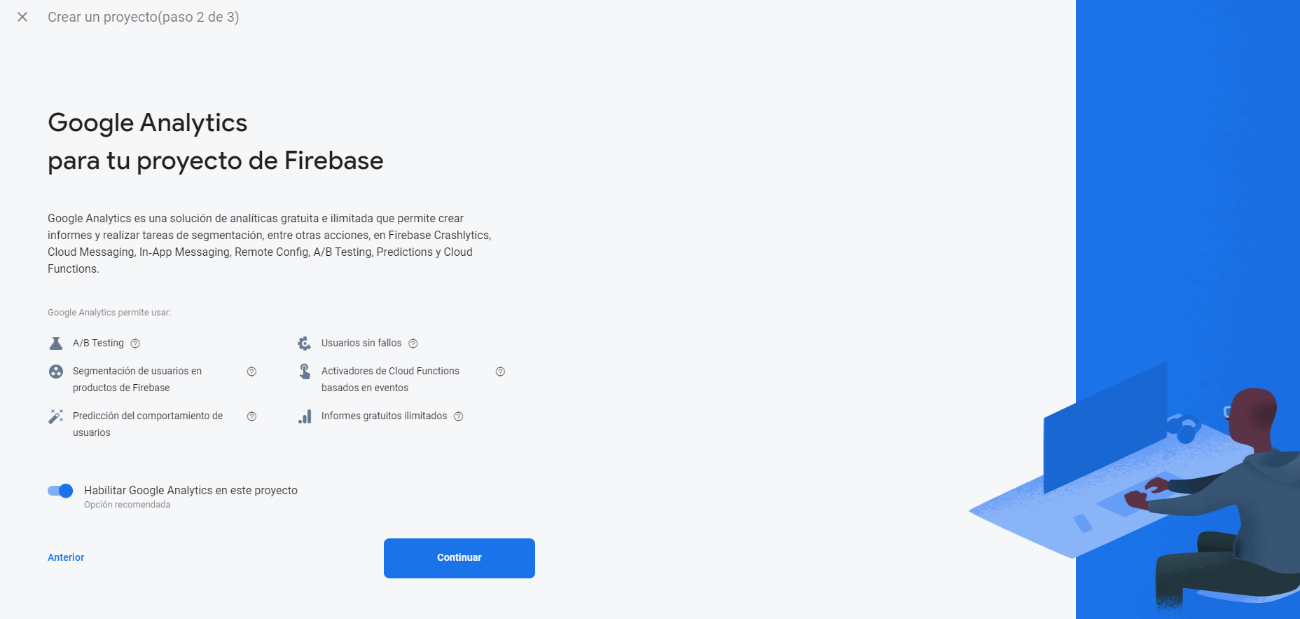
Nos iremos a [Firebase (google.com)](https://firebase.google.com/?authuser=1), para crear un nuevo proyecto tenemos que presionar el botón Comenzar.

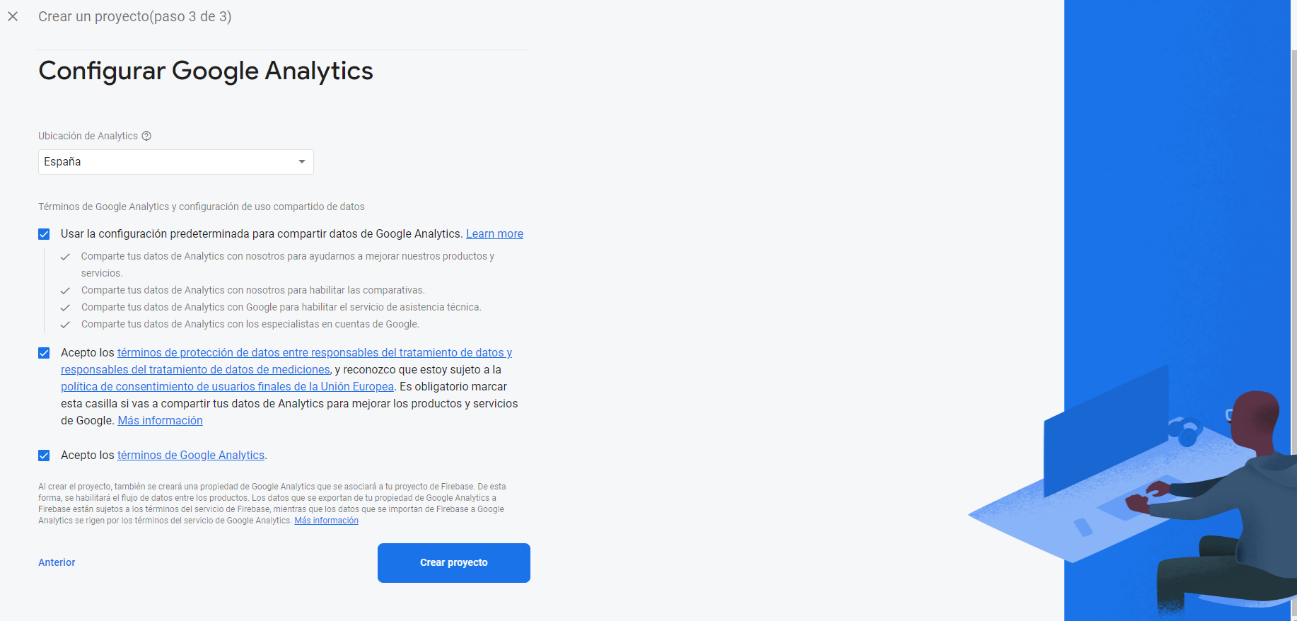


Debemos ponerle un nombre al proyecto y aceptar los términos de Firebase

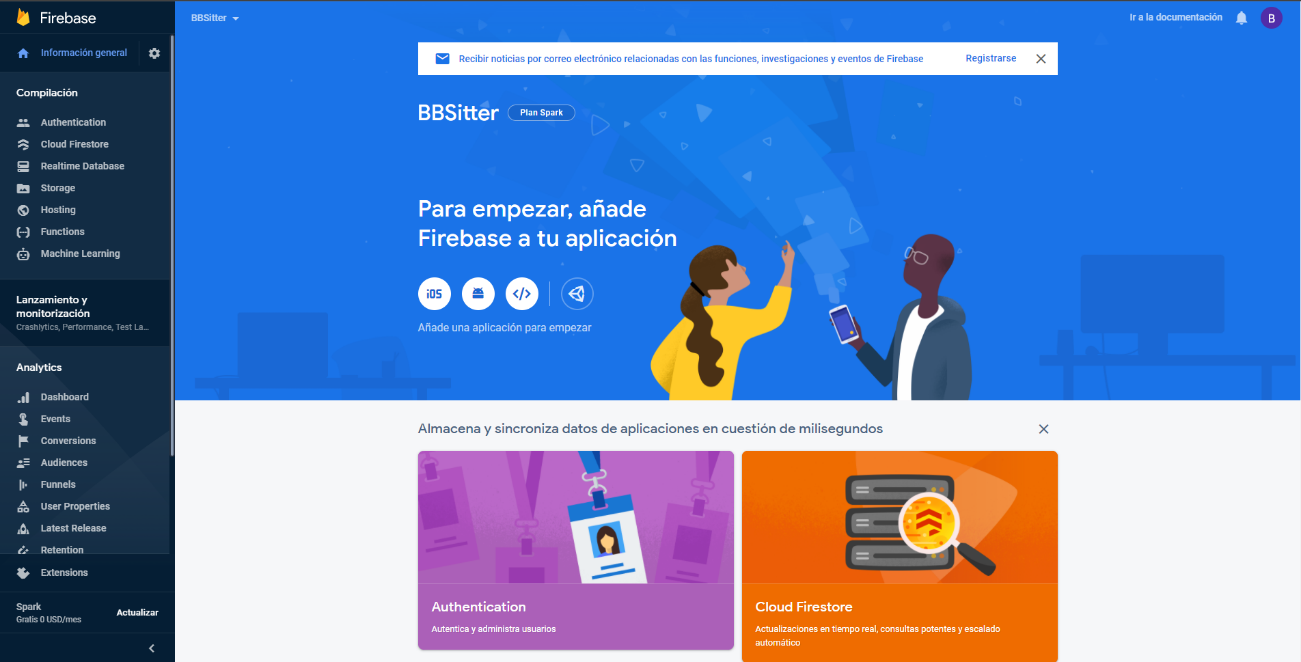


Activamos Google Analytics para Firebase ya que proporciona informes ilimitados y gratuitos de hasta 500 eventos diferentes. El SDK captura de manera automática ciertos eventos clave y propiedades del usuario; además, puedes definir tus propios eventos personalizados a fin de medir factores particularmente importantes.



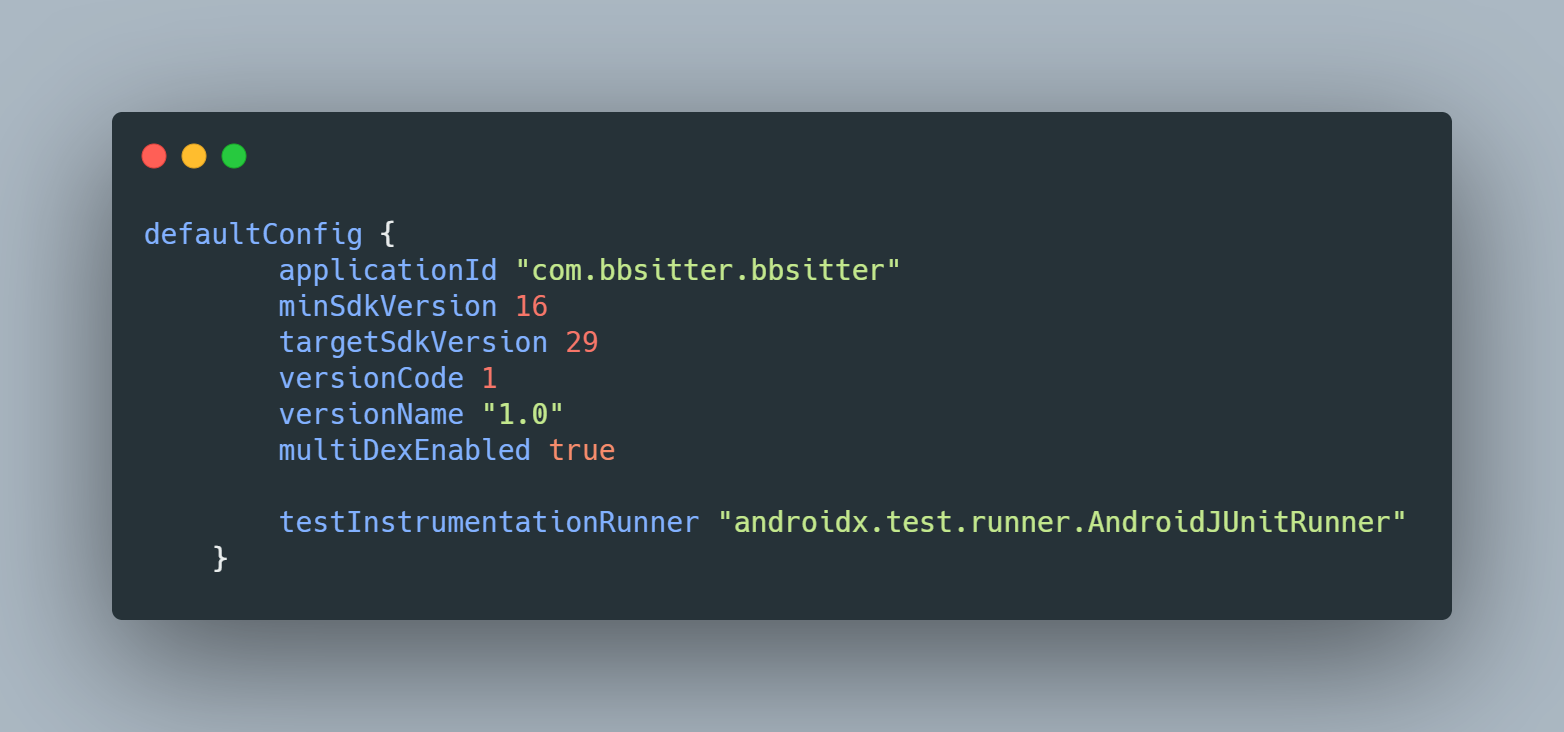
Y configuramos Google Analytics para nuestro país

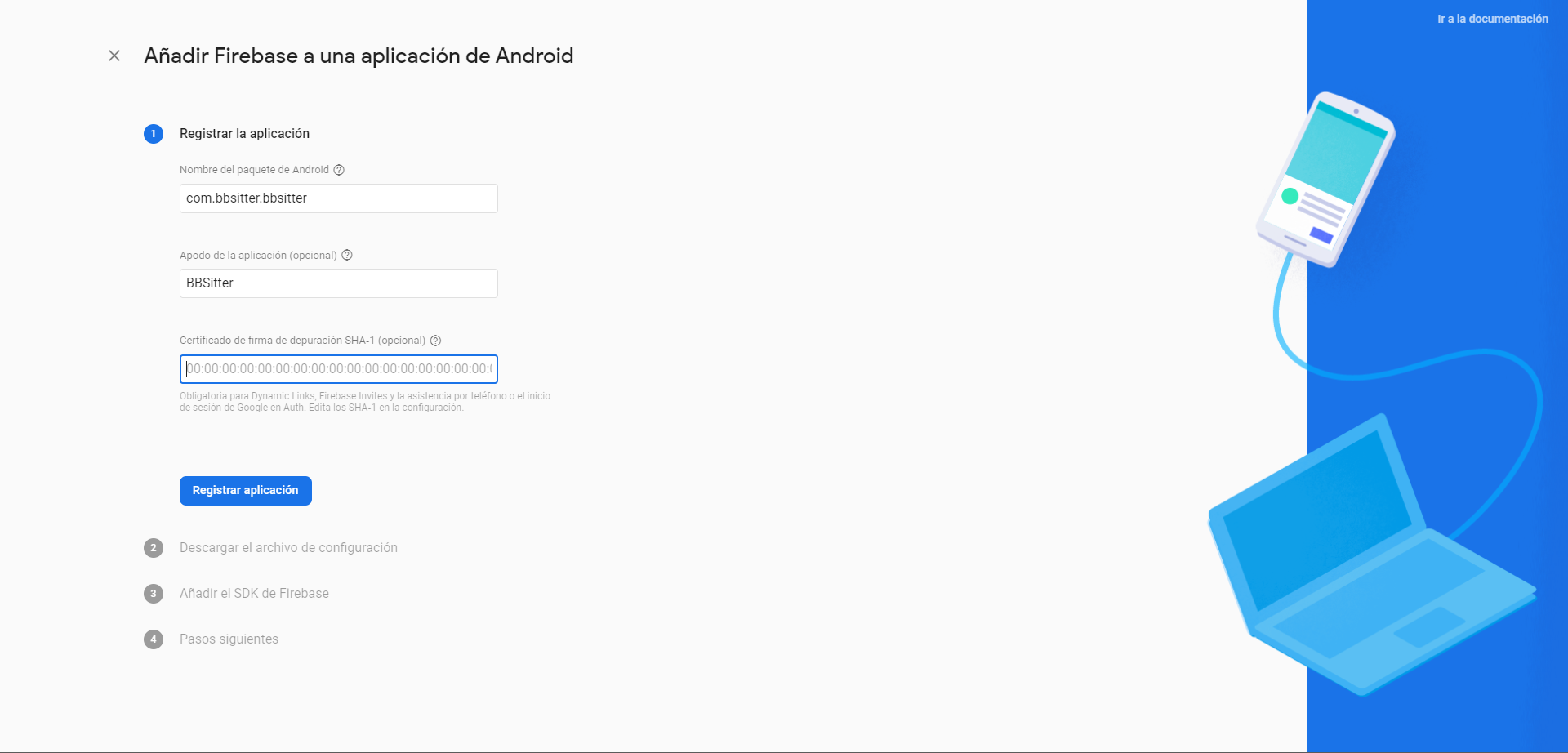
Una vez creado el proyecto nos llevará a la pantalla principal de Firebase donde podremos encontrar todos los servicios de la plataforma y donde podremos enlazar el proyecto con nuestra app en Android Studio haciendo clic en el icono de Android situado en el centro de la pantalla.



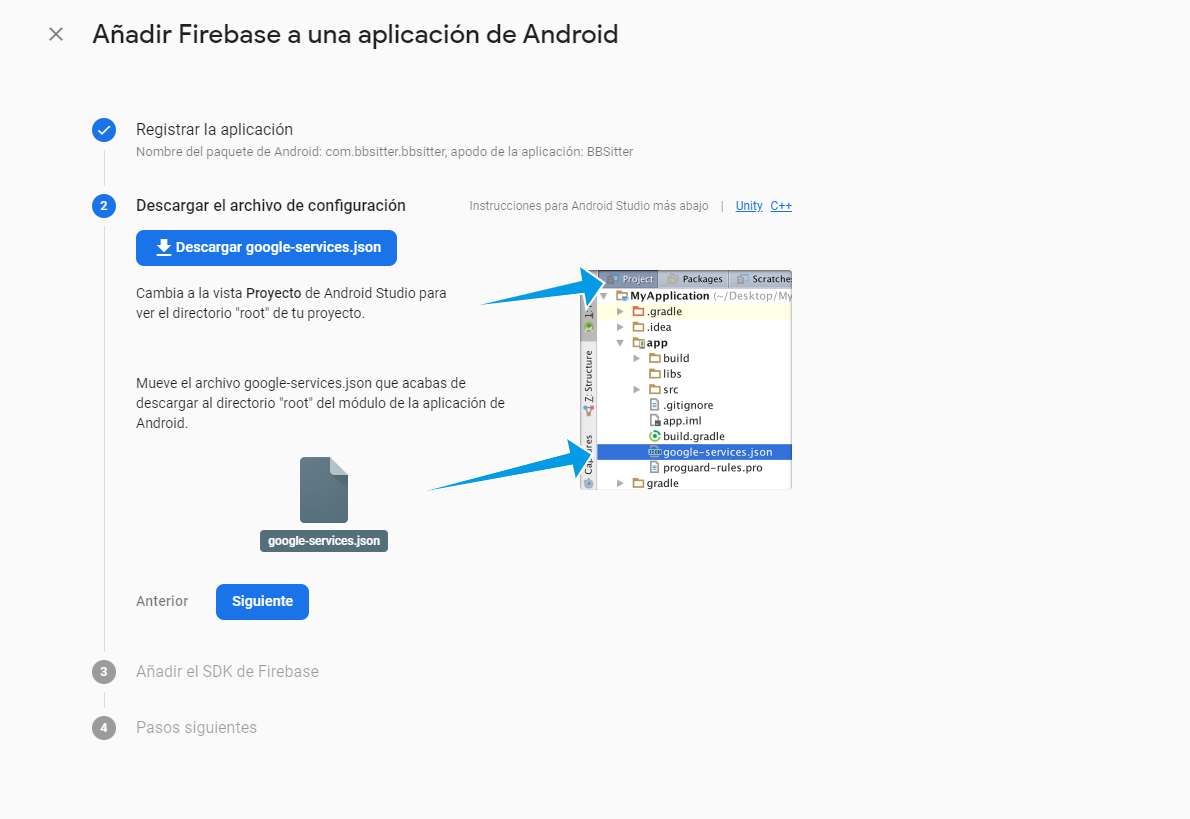
Para ello nos pedirán dos datos obligatorios: el nombre del paquete Android y un apodo.

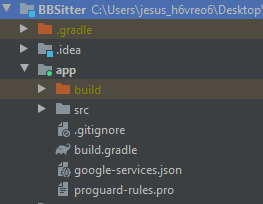
Para saber el nombre del paquete de Android tenemos que acceder al build.Gradle de nuestra app y cogeremos el dato “applicationId” para utilizarlo en Firebase

Y ponemos los datos que hemos recogido en Firebase, hacemos clic en Registrar aplicación



Nos pedirá que descarguemos el archivo google-services.json y que lo implementemos en la carpeta app de nuestro proyecto

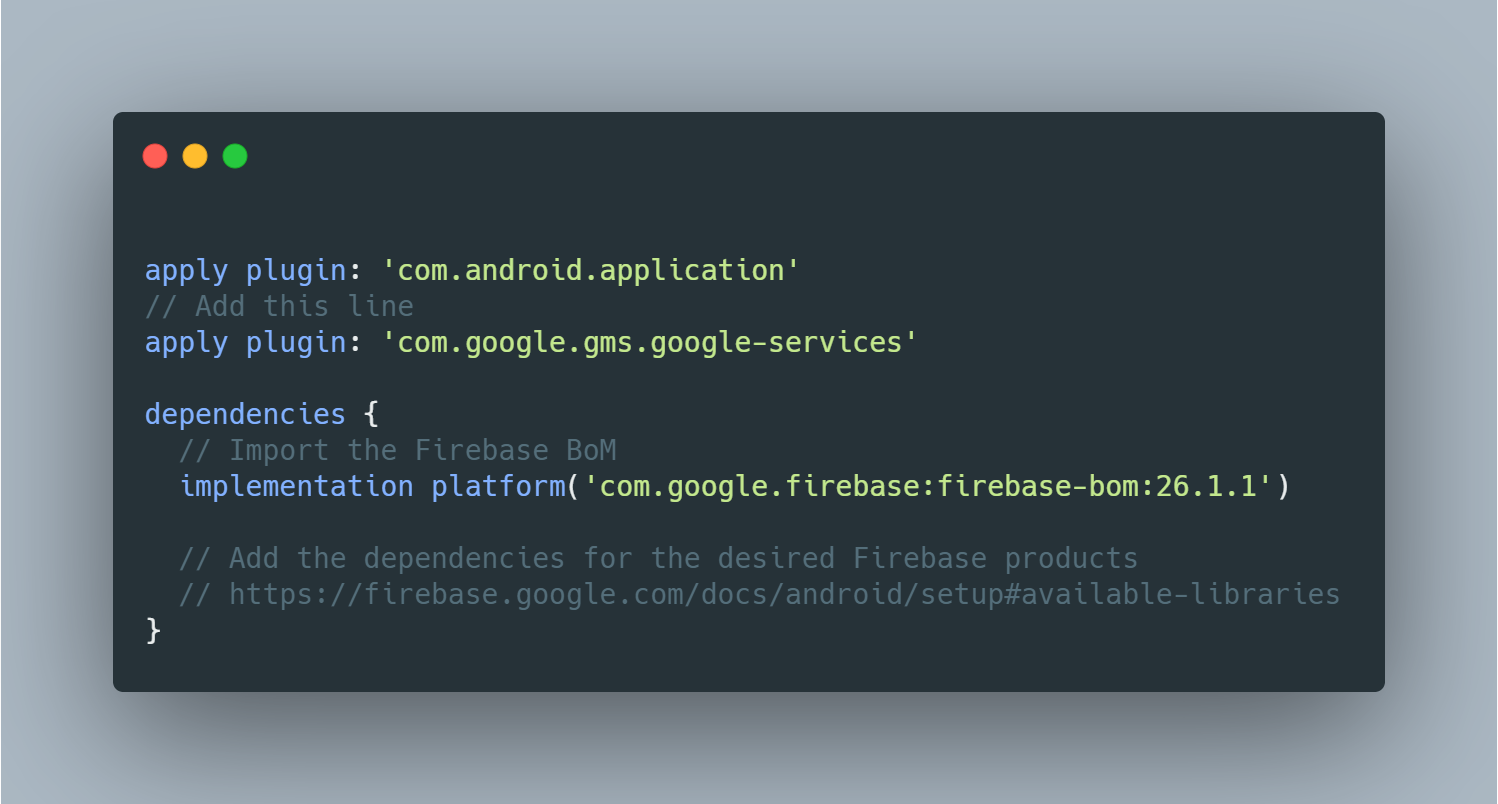




El siguiente paso es verificar que nuestra aplicación tiene implementados los servicios de Google. Para ello vamos otra vez a build.gradle a nivel de proyecto de nuestra aplicación y tiene que tener estas líneas



Y en nuestro build.gradle a nivel de aplicación debemos poner estas líneas



Por último tenemos que sincronizar en build.gradle a nivel de proyecto para que ejecute los cambios



### Servicios de Firebase utilizados

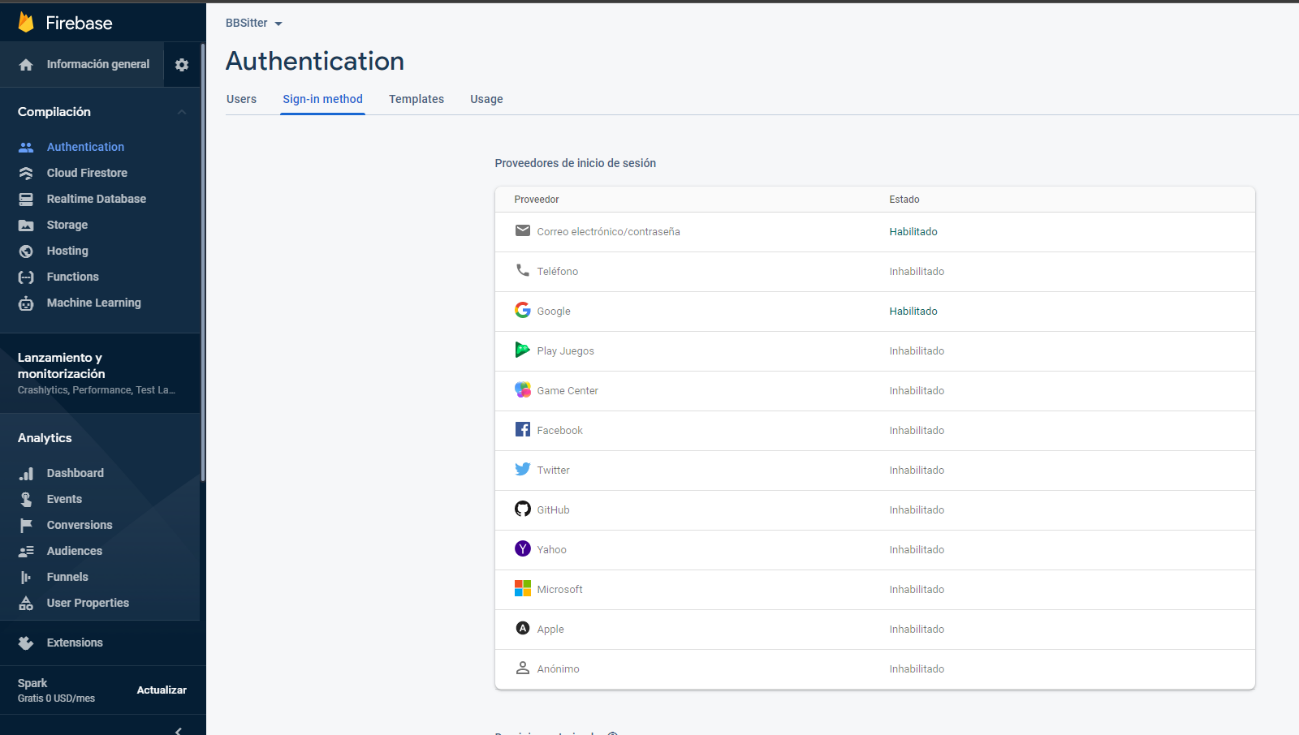
Firebase tiene muchos servicios disponibles, pero en nuestro caso hemos utilizado 3 servicios: Firebase Authentication, Firebase Storage y Firebase Cloud Firestore.

#### Firebase Authentication

La mayoría de las apps necesitan identificar a los usuarios. Conocer la identidad de un usuario permite que una app guarde sus datos en la nube de forma segura y proporcione la misma experiencia personalizada en todos los dispositivos del usuario.

Firebase Authentication proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas (en GitHub) para autenticar a los usuarios en tu app. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook y Twitter, y mucho más, pero en nuestro caso implementamos únicamente registro mediante email y contraseña y Google.

Para empezar, tenemos que habilitar la opción de autentificación por correo en Firebase, para ello vamos a ir al servicio Authentication y habilitamos las opciones que vayamos a utilizar para registrarnos.



Una vez habilitados tendremos que implementar en nuestro build.gradle a nivel de proyecto la dependencia de Firebase Auth.



Existen múltiples métodos para usar el servicio Authentication, los más usados sirven para crear usuarios, registrar usuarios o para cerrar sesión.

Crear usuario



Registrar usuario



Cerrar sesión



#### Firebase Storage

Cloud Storage para Firebase es un servicio de almacenamiento de objetos potente, simple y rentable construido para la escala de Google. Los SDK de Firebase para Cloud Storage agregan la seguridad de Google a las operaciones de carga y descarga de archivos para tus apps de Firebase, sin importar la calidad de la red. Puedes usar nuestros SDK para almacenar imágenes, audio, video y otros tipos de contenido generado por el usuario. En el servidor, puedes usar [Google Cloud Storage](https://cloud.google.com/storage?hl=es) para acceder a los mismos archivos.

En este servicio guardamos imágenes de logos de BBSitter y las imágenes de nuestros usuarios para que sea mucho más intuitiva y eficaz la recogida de datos al estar todo dentro de Firebase.

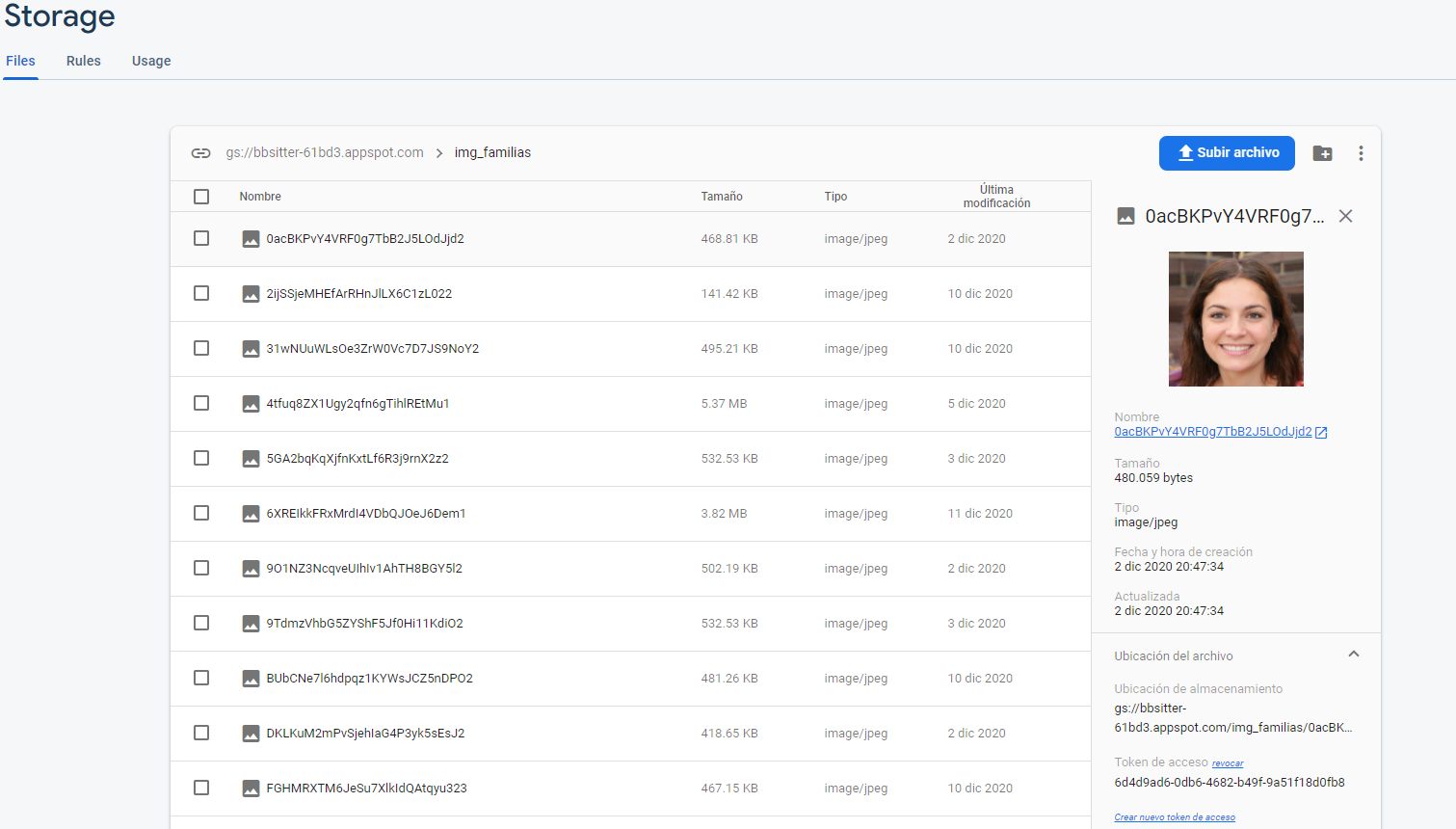
Para comenzar a cargar imágenes en Firebase Storage tenemos de implementar su dependencia.



En nuestra aplicación la utilizamos con el método putFile() para que suba la imagen a nuestro Storage. Lo hacemos de esta forma:



Al guarda la imagen de esta forma se crea una referencia y una url de esa imagen



Con Firebase Cloud Firestore accederemos a esa URL para poder utilizarla dentro de la aplicación.

#### Firebase Cloud Firestore

Cloud Firestore es una base de datos flexible y escalable para la programación en servidores, dispositivos móviles y la Web desde Firebase y Google Cloud Platform.

Al igual que Firebase Realtime Database (versión anterior de base de datos de Firestore), mantiene tus datos sincronizados entre apps cliente a través de agentes de escucha en tiempo real y ofrece asistencia sin conexión para dispositivos móviles y la Web, por lo que puedes compilar apps con capacidad de respuesta que funcionan sin importar la latencia de la red ni la conectividad a Internet.

Las funciones clave más importantes de Firebase Cloud Firestore son las siguientes:

* **Flexibilidad**

El modelo de datos de Cloud Firestore admite estructuras de datos flexibles y jerárquicas. Almacena tus datos en documentos, organizados en colecciones. Los documentos pueden contener objetos anidados complejos, además de subcolecciones.

* **Consultas expresivas**

En Cloud Firestore, puedes usar consultas para recuperar documentos individuales específicos o para recuperar todos los documentos de una colección que coinciden con los parámetros de la consulta.

Las consultas pueden incluir varios filtros en cadena y combinar los filtros con criterios de orden. También se indexan de forma predeterminada, por lo que el rendimiento de las consultas es proporcional al tamaño de tu conjunto de resultados, no del conjunto de datos.

* **Actualizaciones en tiempo real**

Al igual que Realtime Database, Cloud Firestore usa la sincronización de datos para actualizar los datos de cualquier dispositivo conectado.

Sin embargo, también está diseñado para ejecutar consultas de recuperación únicas y sencillas de manera más eficiente que en Realtime Database.

* **Diseñado para ajustarse a la escala de la aplicación**

Cloud Firestore te ofrece lo mejor de la poderosa infraestructura de Google Cloud Platform: replicación automática de datos multirregión, garantías de coherencia sólida, operaciones atómicas por lotes y asistencia real sobre transacciones.

Cloud Firestore sirve para controlar las cargas de trabajo de las bases de datos más complejas de las apps más grandes del mundo. Que no es nuestro caso porque nuestra aplicación es pequeña, pero podría ser más grande en un futuro.

Para poder empezar a utilizar esta potente base de datos NoSQL debemos implementar su dependencia en el build.gradle de nuestro proyecto



Una vez implementada la dependencia podemos escribir datos en Firestore desde nuestra aplicación creando un Mapa de datos y utilizando el método .add() de esta manera:

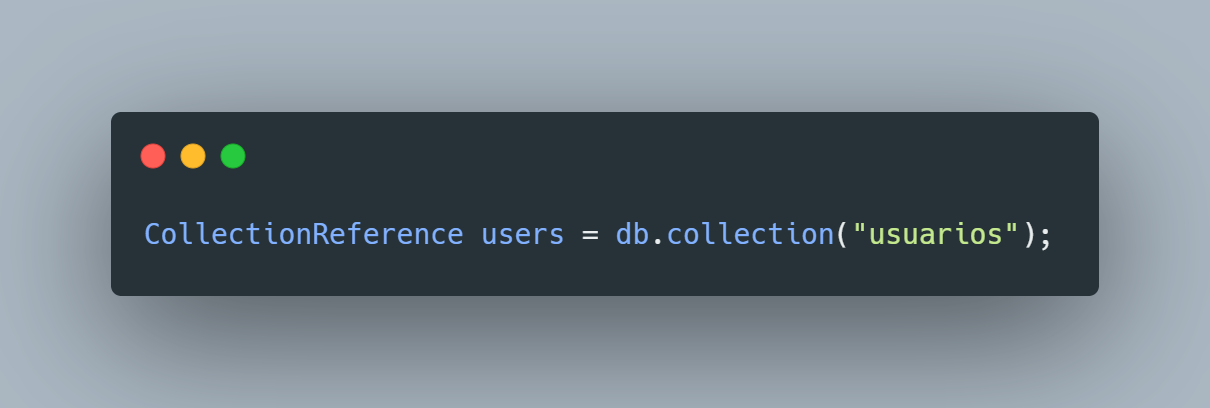
También podemos leer datos de Firestore desde nuestra aplicación. Para ello utilizaremos el método .get()



##### Consultas en la base de datos Firebase Firestore

Cloud Firestore proporciona una potente función de consulta para especificar qué documentos deseas recuperar de una colección o de un grupo de colecciones. Estas consultas también se pueden usar con get() o addSnapshotListener().

Para empezar, tenemos que llamar a la clase CollectionReference.

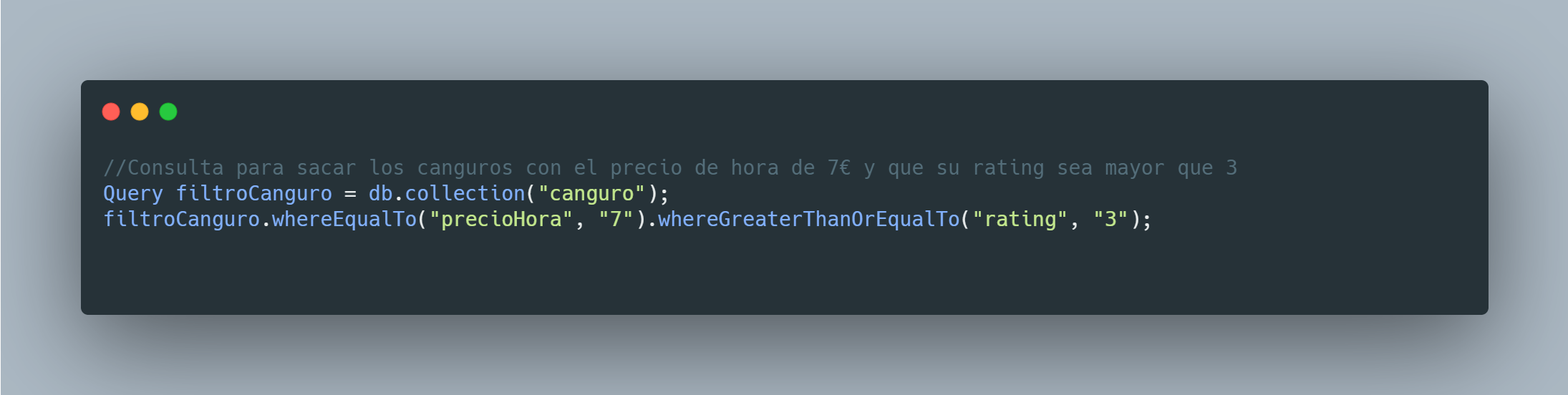


Esta consulta es lo más simple que podemos hacer, accede a la colección de la base de datos llamada “usuarios” y nos saca todos los documentos que tenga dentro.

Podemos hacer consultas más complejas como estas:



También tenemos la opción de crear consultas compuestas, como si de un AND se tratara.



### Modelo de datos

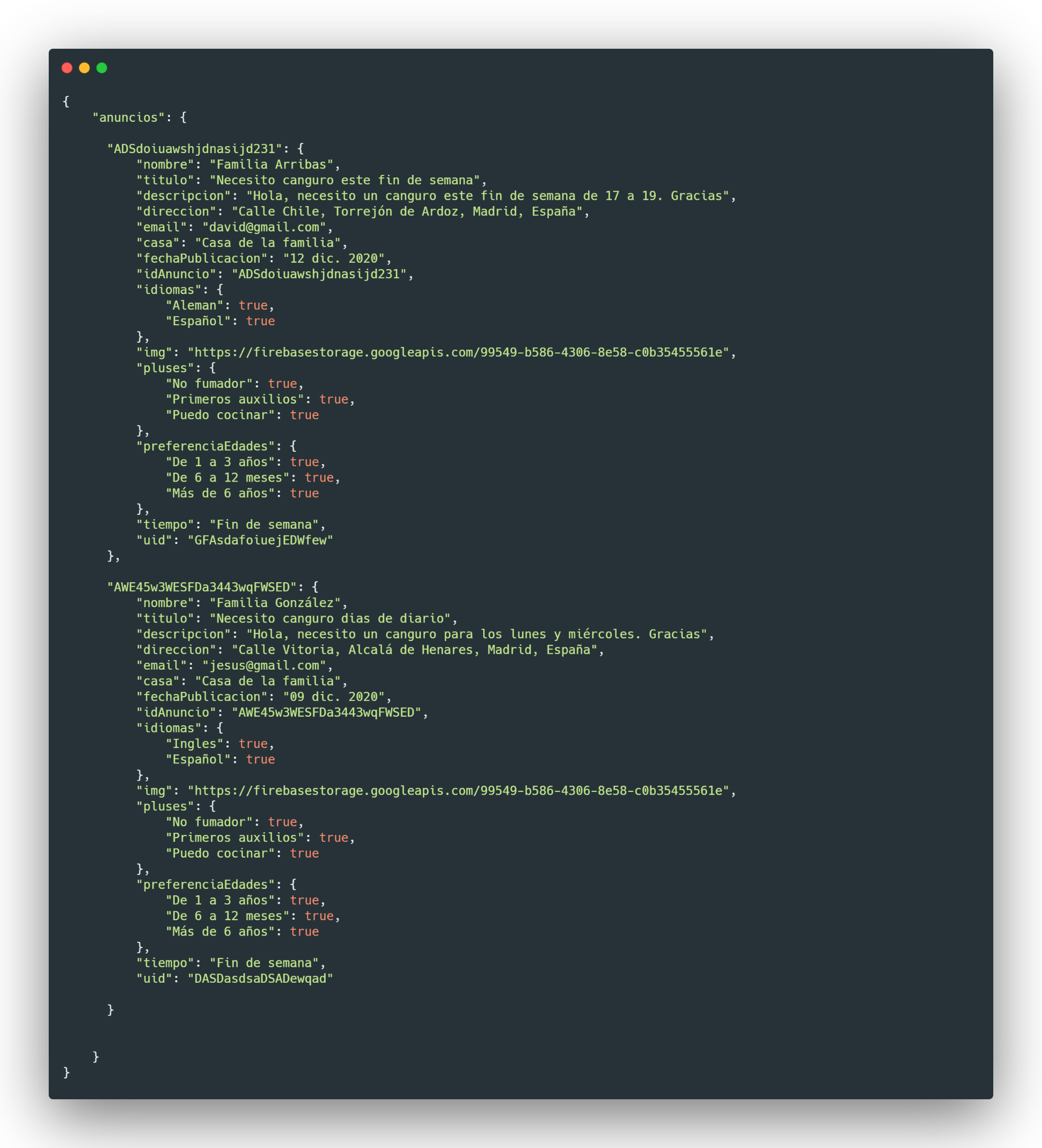
Colección Usuarios



Colección Familias

Colección hijos

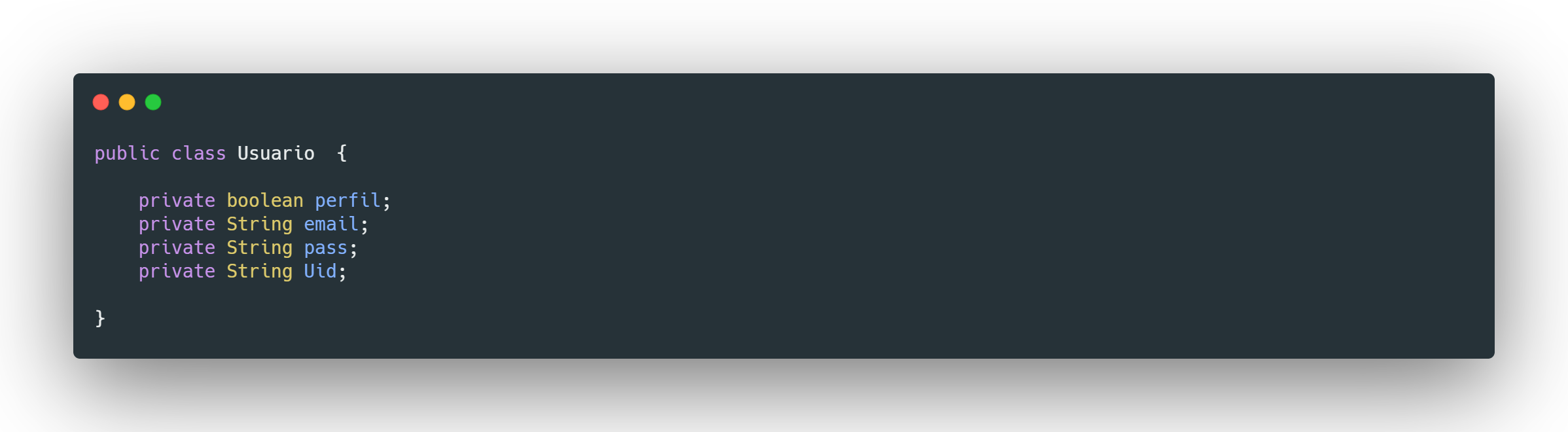


Colección Anuncios

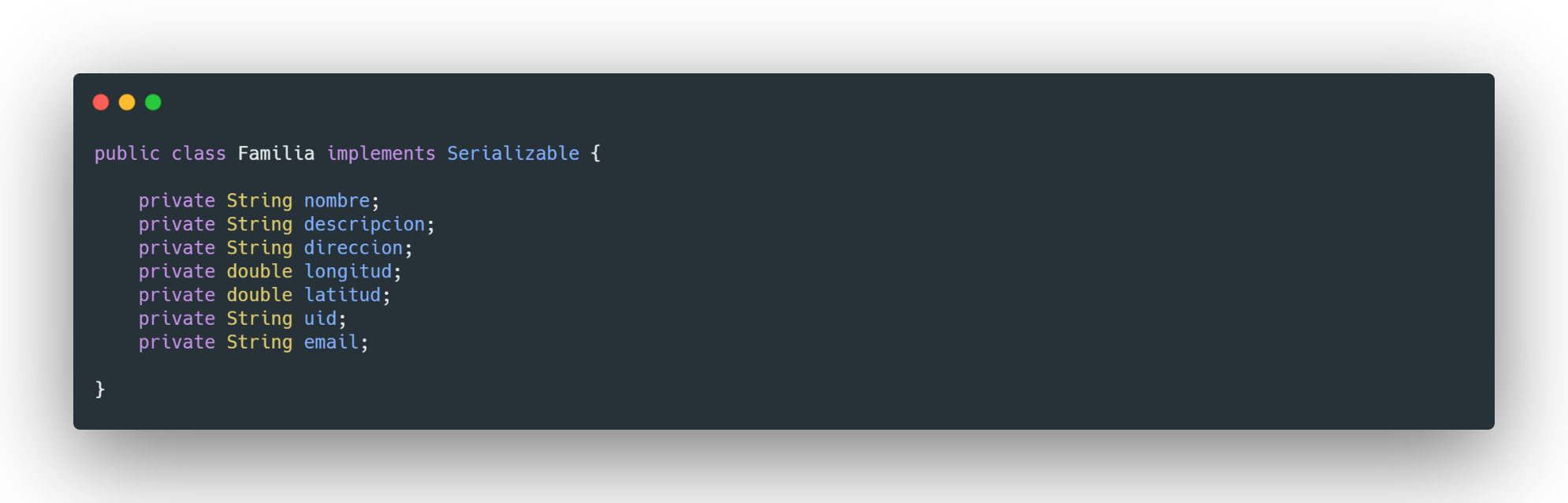
Colección Canguro

Como podemos observar el modelo de datos es no relacional. Cada colección es una clase en nuestro proyecto.

Clase Usuario



Clase Familia



Clase Anuncio



Clase Canguro

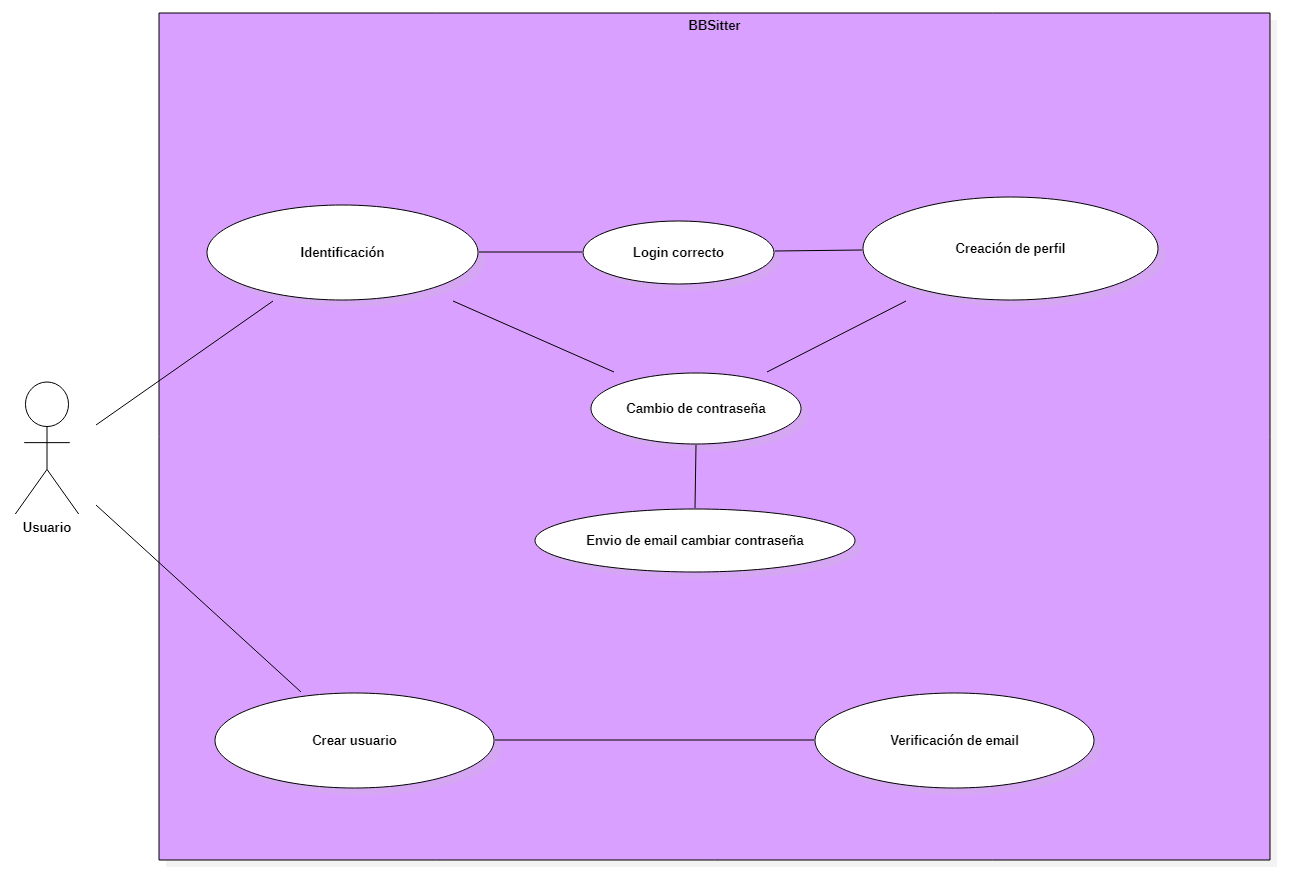


## 7. Casos de Uso

Tenemos en cuenta unos requisitos claves para la utilización de nuestra aplicación. Queremos que el usuario pueda tener una experiencia lo más rápida y eficiente posible, para ello el usuario debería poder hacer estos pasos:

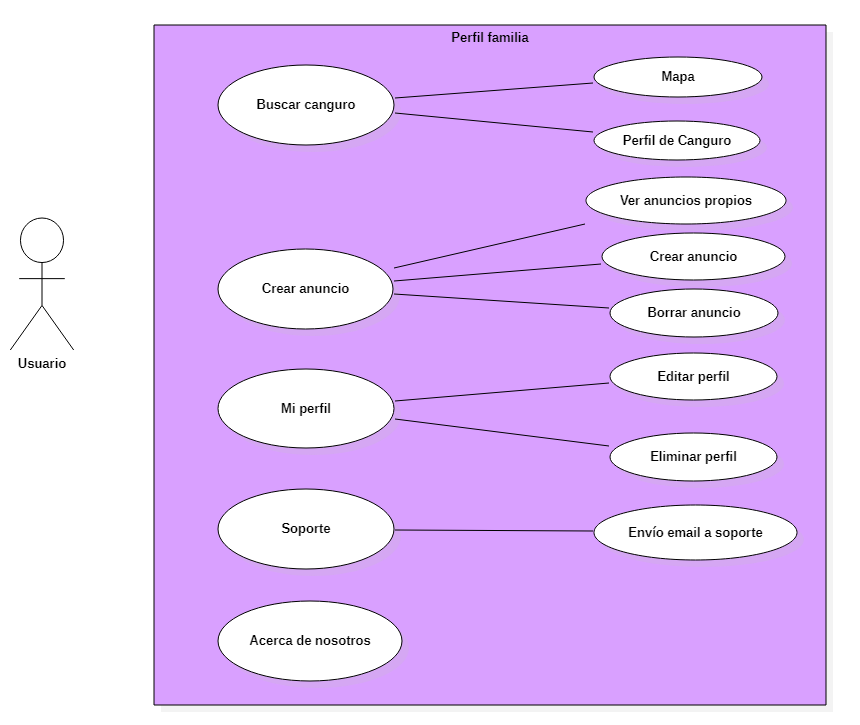
* Usuario
  + Verificación de email
  + Creación de perfil
  + Cambiar contraseña
* Familia
  + Crear un anuncio
  + Visualizar los canguros disponibles filtrados
  + Visualizar mapa con canguros
  + Llamar y mandar un email a los canguros
  + Añadir Hijos
  + Contactar con Soporte
  + Eliminar Anuncios
  + Eliminar Hijos
  + Eliminar su perfil
  + Editar su perfil
  + Cerrar sesión
* Canguro
  + Creación de perfil con sus datos
  + Visualizar anuncios de las familias
  + Editar su perfil
  + Eliminar su perfil
  + Contactar con Soporte
  + Cerrar sesión

Una vez establecido los requisitos que queremos que tenga nuestra aplicación procedemos a hacer los casos de uso.



En esta pantalla podemos hacer tres cosas: Identificación, Registro y cambio de contraseña.

Cuando te identificas puede ser que te equivoques de contraseña, por eso tenemos un control de validación de campos, tanto para errores, para existencia de usuarios en Firebase como si el email esta verificado o no (Esto lo explicamos en el apartado de usabilidad).

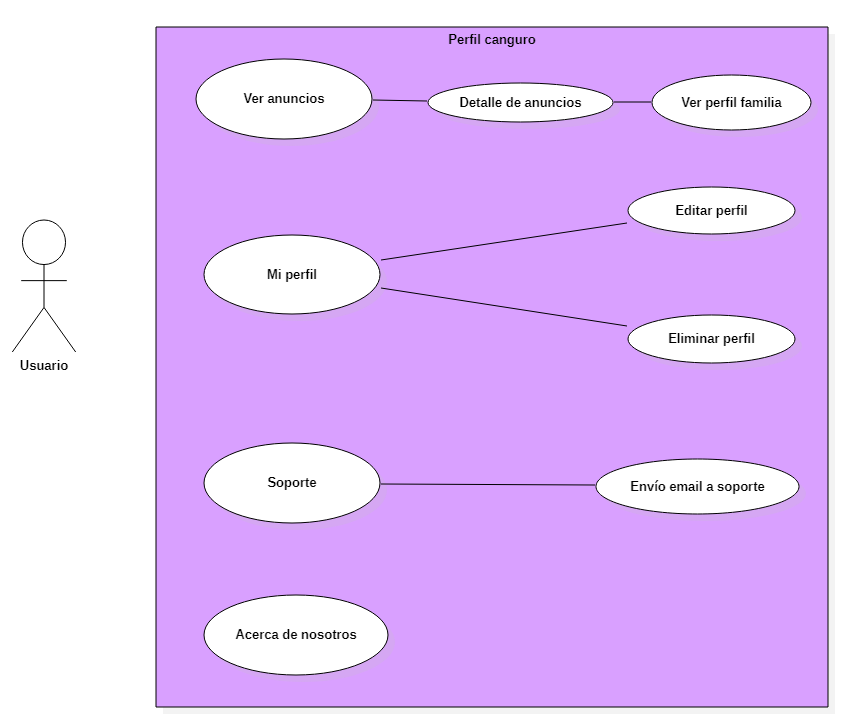


Una vez creado el perfil de familia tenemos diversas opciones dentro de la aplicación.

Podemos ver a todos los canguros tanto en una lista como en un mapa con sus marcadores y dentro de ellos podemos ver su perfil y contactar con ellos tanto por email como por teléfono.

En mi perfil podemos editar nuestro perfil, añadir hijos y eliminar nuestro perfil

En Acerca de nosotros podrá ver nuestros datos y nuestro GitHub



En el perfil de canguro podemos ver los anuncios de las familias en forma de lista. Cuando nos metemos en el detalle del anuncio podemos contactar con la familia por email y podemos ver su perfil para ver sus datos y sus hijos

En Mi perfil podemos modificar nuestro perfil por si tenemos más pluses o más idiomas y podemos eliminar nuestro perfil

En soporte podemos contactar con el soporte de nuestra aplicación por su tuviera algún problema

En Acerca de nosotros podrá ver nuestros datos y nuestro GitHub

## 8.Conclusiones

## 9. Mejoras de la aplicación

Nuestra aplicación la pensamos con idea de que en un futuro pudiera ampliarse para que fuera más completa. Las mejoras que tenemos en mente son:

* Usuario
  + Inicio de sesión con otros proveedores (Google, Facebook, …)
  + Cambiar correo electrónico
* Perfiles
  + Crear chat
  + Calificaciones a los canguros
  + Comentarios a los canguros
  + Filtros personalizables
  + Filtro por distancia en perfil de familia
  + Calificación a las familias
  + Comentarios a las familias
* Base de datos
  + Mejorar la eficiencia de la base sin repetir datos
  + Seguridad en las contraseñas, teléfonos, emails…
  + Compresión de las imágenes para cargas más rápidas

## 10. Bibliografía

[Android Studio](https://es.wikipedia.org/wiki/Android_Studio)

[Firebase](https://es.wikipedia.org/wiki/Firebase)

[Adobe Photoshop](https://es.wikipedia.org/wiki/Adobe_Photoshop)

[Java (lenguaje de programación)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

[Documentación Firebase](https://firebase.google.com/docs?authuser=1)

[Firebase Authentication](https://firebase.google.com/docs/auth?authuser=1)

[Autentica con Firebase mediante un vínculo de correo electrónico en Android](https://firebase.google.com/docs/auth/android/email-link-auth?authuser=1)

[Autentica mediante el Acceso con Google en Android](https://firebase.google.com/docs/auth/android/google-signin?authuser=1)

[APIs de geolocalización  |  Google Maps Platform  |  Google Cloud](https://cloud.google.com/maps-platform)

[Mapas personalizados  |  Google Maps Platform  |  Google Cloud](https://cloud.google.com/maps-platform/maps)

[Places  |  Google Maps Platform  |  Google Cloud](https://cloud.google.com/maps-platform/places)

[Kotlin Programming Language (kotlinlang.org)](https://kotlinlang.org/)

[Cloud Firestore](https://firebase.google.com/docs/firestore?authuser=1)

[Primeros pasos con Cloud Firestore](https://firebase.google.com/docs/firestore/quickstart?authuser=1)

[Modelo de datos de Cloud Firestore](https://firebase.google.com/docs/firestore/data-model?authuser=1)

[Elige una base de datos: Cloud Firestore o Realtime Database](https://firebase.google.com/docs/firestore/rtdb-vs-firestore?authuser=1)

[SDK y bibliotecas cliente](https://firebase.google.com/docs/firestore/client/libraries?authuser=1)

[Elige una estructura de datos](https://firebase.google.com/docs/firestore/manage-data/structure-data?authuser=1)

[Agrega datos a Cloud Firestore](https://firebase.google.com/docs/firestore/manage-data/add-data?authuser=1)

[Transacciones y escrituras en lotes](https://firebase.google.com/docs/firestore/manage-data/transactions?authuser=1)

[Borra datos de Cloud Firestore](https://firebase.google.com/docs/firestore/manage-data/delete-data?authuser=1)

[Administra Cloud Firestore con Firebase console](https://firebase.google.com/docs/firestore/using-console?authuser=1)

[Obtén datos con Cloud Firestore](https://firebase.google.com/docs/firestore/query-data/get-data?authuser=1)

[Obtén actualizaciones en tiempo real con Cloud Firestore](https://firebase.google.com/docs/firestore/query-data/listen?authuser=1)

[Realiza consultas simples y compuestas en Cloud Firestore](https://firebase.google.com/docs/firestore/query-data/queries?authuser=1)

[Cloud Storage](https://firebase.google.com/docs/storage?authuser=1)

[Comienza a usar Cloud Storage en Android](https://firebase.google.com/docs/storage/android/start?authuser=1)

[Crea una referencia de Storage en Android](https://firebase.google.com/docs/storage/android/create-reference?authuser=1)

[Sube archivos en Android](https://firebase.google.com/docs/storage/android/upload-files?authuser=1)

[Descarga archivos en Android](https://firebase.google.com/docs/storage/android/download-files?authuser=1)

[Borra archivos en Android](https://firebase.google.com/docs/storage/android/delete-files?authuser=1)

[Material Design](https://material.io/)

[Design - Material Design](https://material.io/design)

[Components - Material Design](https://material.io/components)

[Develop - Material Design](https://material.io/develop)

[LottieFiles](https://lottiefiles.com/)

[Iconos](https://icon-icons.com/es/)

[Carbon | Codigo](https://carbon.now.sh/)

[Download Android Studio and SDK tools  |  Android Developers](https://developer.android.com/studio?hl=zh-cn)

[Android Studio release notes  |  Android Developers](https://developer.android.com/studio/releases)

[SDK Tools release notes  |  Android Developers](https://developer.android.com/studio/releases/sdk-tools)

[Emulator release notes  |  Android Developers](https://developer.android.com/studio/releases/emulator)

[Java 8+ APIs available through desugaring  |  Android Developers](https://developer.android.com/studio/write/java8-support-table)

[Stack Overflow en español](https://es.stackoverflow.com/)

[GitHub - jesusgm94/Proyecto-BBSitter](https://github.com/jesusgm94/Proyecto-BBSitter/)

[Git - gittutorial Documentation (git-scm.com)](https://git-scm.com/docs/gittutorial)

[Instant Chat Messenger with Cloud Firestore | Eric Decanini](https://www.ericdecanini.com/2019/12/16/instant-chat-messenger-with-cloud-firestore/)

[This Person Does Not Exist](https://thispersondoesnotexist.com/)

[MoureDev by Brais Moure - YouTube](https://www.youtube.com/channel/UCxPD7bsocoAMq8Dj18kmGyQ)

[Firebase - YouTube](https://www.youtube.com/user/Firebase)

[Fireship - YouTube](https://www.youtube.com/channel/UCsBjURrPoezykLs9EqgamOA)

[FalconMasters - YouTube](https://www.youtube.com/channel/UCJl1YajcPWTeJNsQhGyMIMg)