



UNIVERSIDAD ADVENTISTA DE CENTRO ÁMERICA
ESCUELA DE INGENIERÍA EN SISTEMAS

Tema: Implementación de Google Firebase en Android

Programación III

Profesor: Brian Funez

Estudiantes:

Brenda Maldonado Buendía

Eysell Acosta Sosa

Fecha: 27/03/2019

Contenido

Introducción.....	3
Antecedentes	4
Desarrollo.....	5
¿Cómo iniciar Firebase?	6
• Data:.....	6
• Security & Rules:	6
• Simulator:	6
• Analytics:	6
• Storage:.....	7
• Notificaciones:	7
¿Cómo funciona?	7
• Sistema de comprobación de los correos de los registrados:	8
• Analítica:.....	8
• Login & Auth:.....	8
• Hosting:	9
• Secrets:	9
Instalar Firebase en un proyecto web	9
Acceso a los datos por referencia	9
Breves notas sobre el uso de referencias.....	10
Evento value	12
Bibliografía	14
ANEXOS:	15
Consola de FireBase	15

Introducción

Sabemos lo importante que es la tecnología en la actualidad y es por esta razón que cada vez se desarrollan más aplicaciones para cubrir las necesidades de las personas que disfrutan de estas herramientas, debido a eso en esta ocasión hablaremos acerca de una tecnología que se ha robado la atención y los aplausos en los últimos años y estamos hablando de Firebase, esta aplicación comenzó a darse a conocer gracias a que permite implementar una base de datos en tiempo real para aplicaciones móviles de manera muy sencilla en pocas palabras cuentas con un backend disponible para las aplicaciones con tan solo un clic.

Antecedentes

En 2011, James Tampin y Andrew Lee fundaron un startup llamado Envelope. Esta plataforma proporcionó a los desarrolladores una API que permitía la integración de sistemas de chat en sus páginas web.

Tras lanzar el servicio de chat, Tampin y Lee observaron que dicho chat estaba siendo utilizado en gran escala por los desarrolladores para pasar paquetes de información de sus aplicaciones, como el estado de las partidas en el caso de los juegos. Fue entonces cuando decidieron separar ambas funcionalidades, el sistema de chat y el sistema de arquitectura en tiempo real que la propulsaba, dando como resultado la fundación de Firebase en abril de 2012.

En octubre de 2014, Firebase fue comprado por Google. En octubre de 2015, Google adquirió Divshot para fusionar su equipo con el equipo de Firebase. Desde su adquisición, Firebase ha crecido dentro de Google y ha expandido sus servicios para convertirse en una plataforma unificada para desarrolladores móviles. Firebase se integra actualmente con otros servicios de Google para poder ofrecer productos a mayor escala para los desarrolladores. En enero de 2017 Google adquirió Tejido y Crashlytics de Twitter para unir sus servicios al equipo de Firebase.

Desarrollo



Es la nueva plataforma de desarrollo móvil que permite desarrollar apps multiplataforma (Android, iOS y web).

Firebase es un claro ejemplo de las posibilidades de desarrollo en la nube, a partir de un servicio web nos ofrece la posibilidad de programar aplicaciones con

datos que se sincronizan en tiempo real a través de múltiples dispositivos, evitando muchas de las tareas difíciles en las que tendríamos que dedicar mucho tiempo al programar.

En pocas palabras Firebase es una base de datos remota, alojada en la nube y capaz de ser accedida desde navegadores y apps para dispositivos que tiene como principal característica que responde en tiempo real a los cambios realizados en los datos. En la práctica, mediante Firebase podemos escribir datos en una base de datos y que estos datos se comuniquen a todos los clientes conectados a esa misma fuente de datos.

Firebase cuenta con una serie de librerías mediante las cuales podemos conectarnos y mantenernos suscritos a los cambios de los datos, compatibles con los sistemas más comunes como iOS, Android y web, pero también a varios lenguajes de programación del lado del servidor como podrían ser Python o PHP.

Otro servicio que nos ofrece es el de la autenticación, de modo que podremos realizar login en el sistema y crear aplicaciones multiusuarios sin tener que programar toda la parte de la autenticación. Permite login con usuario y clave, así como con distintos conectores sociales como Facebook, Twitter, Google, etc.

Nos ofrece la posibilidad de realizar aplicaciones de funcionalidades avanzadas programando solamente la parte del lado del cliente. Nos permite programar una aplicación centrándonos únicamente en la parte del frontend, dejando el backend gestionado por completo con Firebase.

¿Cómo iniciar Firebase?

Para iniciar un proyecto con Firebase tenemos que registrarnos en el servicio. Es gratuito siempre que nuestra aplicación este en fase de desarrollo. Cuando queramos poner el proyecto en producción tendremos que suscribir un plan, con valores a partir de los 5 euros mensuales.

Una vez creada nuestra aplicación podemos administrarla desde un panel de control que nos ofrece las opciones que resumimos a continuación:

- Data: nos ofrece el acceso para visualizar y editar nuestro modelo de datos. Básicamente lo que editamos es un JSON, ya que el sistema de Firebase es de base de datos documental.
- Security & Rules: Nos permite el acceso a una sección donde podemos escribir las reglas de acceso a los recursos dentro de Data, así como la validación que se aplicará para decidir si se permite o no una escritura de información.
- Simulator: desde donde podemos simular accesos en diversas situaciones para verificar si tendrían o no permisos para realizar ciertas operaciones.
- Analytics: Una serie de reportes estadísticos con los que podemos monitorear el estado de la aplicación y la cantidad de recursos que están siendo destinados para su funcionamiento (usuarios concurrentes, transferencia, etc.).

- **Storage:** Un sistema de almacenamiento de archivos de los usuarios. Los usuarios pueden subir archivos desde el navegador y desde dispositivos, gracias al nuevo SDK, sin tener que tocar ni una línea de código backend. Los archivos pueden cargarse sin importar las condiciones de red, en background, y si una subida se interrumpe, se reanuda automáticamente cuando vuelve la conexión, continuando por donde se había parado. Todo esto lo hace especialmente indicado para usuarios con pobres conexiones de red. Además, usa Google Cloud Storage, por lo que la escalabilidad es casi infinita.
- **Notificaciones:** Ahora somos capaces de enviar notificaciones a los móviles del usuario, usando servidores de notificaciones de Google.

Es un servicio gratuito para desarrolladores de apps para dispositivos móviles que permite enviar notificaciones orientadas a los usuarios.

Basada en Firebase Cloud Messaging y en el SDK FCM , The Notifications composer (the Notifications composer) ofrece una opción para desarrolladores y organizaciones que buscan una plataforma de notificación flexible que requiera codificación mínima para comenzar, y una consola gráfica para enviar mensajes. Usando la GUI de la consola the Notifications composer , puedes reactivar y retener tu base de usuarios, impulsar el crecimiento de la app y respaldar campañas de comercialización.

¿Cómo funciona?

Usa la GUI de la consola the Notifications composer para redactar y enviar notificaciones a todos los destinos de mensajes admitidos. Firebase Cloud Messaging se ocupa del enrutamiento y la entrega a los dispositivos de destino.

Cuando tu app está en segundo plano en el dispositivo de un usuario, las notificaciones se entregan en la bandeja del sistema. Cuando un usuario hace clic en la notificación, el lanzador de apps abre tu app. Si quieres, también puedes agregar administración de mensajes del cliente para recibir notificaciones en tu app cuando ya esté en primer plano en el dispositivo del usuario.

Ruta de implementación

1	Configura el SDK FCM.	Agrega una línea de código para agregar la dependencia FCM a tu app.
2	Envía notificaciones desde la consola Notifications.	Abre la consola Notifications y comienza a enviar notificaciones a segmentos de usuarios.
3	(Opcional) Agrega manejo de mensajes.	Agrega lógica de manejo de mensajes a tu app cliente para recibir notificaciones en tu app cuando ya se encuentre en primer plano en el dispositivo del usuario.

Agrega el SDK FCM como una dependencia en tu app e intenta enviar una notificación al segmento de usuarios predefinido **version** en iOS o Android. Selecciona la versión de la app que estás desarrollando, y la notificación se entregará a dispositivos de prueba.

Ejecuta el inicio rápido de FCM para iOS o Android.

- Sistema de comprobación de los correos de los registrados: Anteriormente no había posibilidad de confirmar los correos de los usuarios registrados y aunque esto depende del servicio de autenticación que ya existía, es una de las novedades más demandadas.
- Analítica: de uso de la aplicación, principalmente enfocado al uso móvil.
- Login & Auth: desde aquí se puede controlar cuáles son los mecanismos que estarán disponibles para la autenticación de usuarios, tanto email y clave como por redes sociales, junto con los datos de configuración.

- Hosting: Inicialmente con una ayuda para aprovechar lo que sería un espacio de alojamiento donde desplegar la app realizada. Una vez activado también se pueden controlar aquí las configuraciones del hosting.
- Secrets: por último, una página desde donde se administran claves del API de nuestra App que serán utilizadas para accesos del lado del servidor y autenticación por medio de JSON Web Tokens.

Instalar Firebase en un proyecto web

Para hacer funcionar a Firebase se tiene que cargar una librería JavaScript que contiene el código del API que este sistema nos proporciona para acceder a los datos.

Lo más cómodo es que la instales usando Bower con el correspondiente comando:

Bower install Firebase

Una vez descargadas las dependencias, verás el archivo firebase.js dentro de bower_components/Firebase.

Ahora en la página .HTML que quieras usar Firebase debes colocar el script firebase.js en la cabecera. Eso nos permitirá usar el API de Firebase.

```
<script src="bower_components/Firebase/firebase.js"></script>
```

Acceso a los datos por referencia

Para acceder a información de nuestra aplicación Firebase usa lo que se conoce como referencias. Son enlaces con tu modelo de datos que apuntan a un nodo en concreto del JSON. A través de esas referencias podemos acceder a un documento completo o navegar por los hijos para acceder a otros nodos dependientes. Para hacer todas estas operaciones de recorrido de los datos Firebase nos expone una nutrida API.

Estas referencias se crean a través de una llamada a la función `Firebase()`, indicándole como parámetro el nodo de nuestro modelo que queremos referenciar. Los nodos están compuestos por una ruta definida con el nombre de tu aplicación, algo como `https://ejemplo.firebaseio.com` y luego una ruta. Si no indicas ninguna ruta simplemente accederás a la raíz de tus datos y si especificas una ruta podrás acceder a un nodo más interior.

```
var ref = new Firebase("https://juegosdemesa.firebaseio.com/");
```

Sin embargo, aunque nuestra referencia apunte a la raíz, puedo dirigirme a uno de sus hijos con el método `child()`.

```
ref.child("appName")
```

Breves notas sobre el uso de referencias

Los datos de la base de datos de Firebase se organizan en un árbol, algo parecido a lo que conocemos como JSON. Así, cuando estás trabajando con datos, unos pueden estar dentro de otros y otros dentro de otros, para conseguir cualquier estructura de almacenamiento que necesites. Lo importante ahora es que, a nivel de referencias, todos esos niveles de anidación se separan mediante el carácter barra `"/"`.

Por ejemplo, si tienes una lista de usuarios y cada uno tiene un identificador y una serie de datos como su nombre o email, tendríamos una estructura como esta:

```
users: {  
  iduser1: {  
    nombre: 'Miguel Angel Alvarez',
```

```
    email: 'malvarez@desarrolloweb.com'
  },
  iduser2: {
    nombre: 'Alvaro Martínez',
    email: 'alvaro@escuela.it'
  }
}
```

Si mediante Firebase necesito acceder a la lista de usuarios usaré una referencia como esta:

```
var referencia = databaseService.ref('users');
```

Si ahora quiero acceder al primer usuario de la lista, entonces podré construir mi referencia expresando los niveles de anidación con la "/".

```
var referencia = databaseService.ref('users/iduser1');
```

Si quiero acceder únicamente al nombre del usuario primero, entonces mi referencia quedaría así:

```
var referencia = databaseService.ref('users/iduser1/nombre');
```

Como alternativa, puedes navegar entre referencias por medio del método `child ()` del API de Firebase:

```
var referencia = databaseService.ref('users').child('iduser1').child('nombre');
referencia.Set('Miguel Angel Alvarez Sánchez');
```

Evento value

Este sirve para sincronizarse con el dato.

En Firebase cuando accedes a una información puede tener un estado determinado, pero sin embargo al ser una aplicación en tiempo real, ese estado puede cambiar en cualquier instante. Firebase avisará en el mismo momento en el que se ha producido el cambio facilitando el nuevo estado de manera automática.

Por tanto, mientras que en otro tipo de sistemas consultarías un dato una vez, en Firebase debemos estar preparados para los cambios posteriores.

Por eso para leer un dato lo que vamos a hacer es suscribirnos a un evento. Nada más suscribirnos obtendremos el valor actual y cada vez que el dato cambie se disparará ese mismo evento indicando el nuevo valor alojado en ese dato. Igual que en librerías como jQuery, nos suscribimos a eventos con el método `on ()`. Por ejemplo, nos vamos a suscribir en "value".

```
ref.child("appName").on("value",function(snapshot){  
  console.log(snapshot.Val());  
  document.getElementById("titular").textContent=snapshot.Val();  
});
```

El método `on ()` me permite suscribirme indicando el nombre del evento que me interesa, en este caso "value" y la función que queremos registrar como manejadora del evento. En esa función recibiremos el "snapshot" que es una instantánea del dato que tenemos en Firebase. Cuando cambie se disparará de nuevo el método recibiendo la siguiente instantánea.

Para acceder al valor de la instantánea usamos el método `val ()`.

En este caso dentro del manejador de evento mostramos el valor por `console.log ()` y luego usamos ese mismo valor para cargarlo como contenido en un elemento del DOM que tiene el id "titular".

Conexiones por Sockets

En Firebase las conexiones se realizan mediante sockets, por eso es tan rápido en recibir los cambios que hayan realizado en el modelo. Cuando un cliente abre una página donde se usa Firebase se mantiene abierto un canal de comunicación con el servidor bidireccional, que permite comunicarse desde el cliente al servidor y desde el servidor a todos los clientes.

Firebase también responde a las operaciones por medio de una interfaz "REST", como cualquier API REST que puedas conocer. De hecho, las conexiones mediante REST son las que utilizarás desde lenguajes de servidor como PHP o Python. Sin embargo, las comunicaciones por sockets son mucho más rápidas debido a que no tienen que hacer toda la operativa de una request al servidor con HTTP, con todas las etapas del protocolo. El canal de comunicación permanece abierto y los datos fluyen de un lado a otro sin más. Además, los sockets permiten una comunicación que ahorra llamadas en aplicaciones al servidor en tiempo real, porque el cliente no tiene necesidad de hacer constantes request para recuperar los datos que hayan cambiado, sino que simplemente tiene que escuchar el socket, a la espera que Firebase le mande los cambios cuando se estén produciendo.

Bibliografía

Desarrolloweb.com. (s.f.). Obtenido de Desarrolloweb.com:

<https://desarrolloweb.com/manuales/manual-firebase.html>

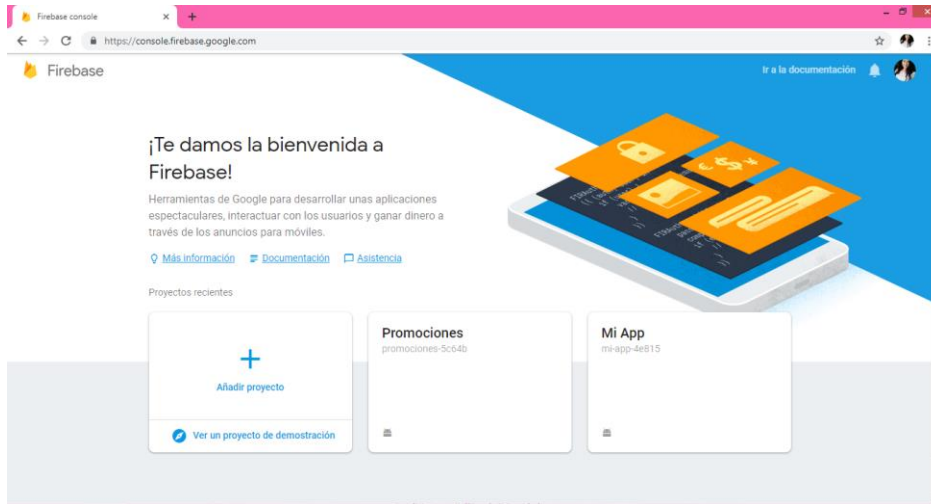
El android libre. (s.f.). Obtenido de El android libre:

<https://elandroidelibre.lespanol.com/2016/05/firebase-plataforma-desarrollo-android-ios-web.html>

FireBase. (s.f.). Obtenido de FireBase: <https://console.firebase.google.com/>

ANEXOS:

Consola de FireBase



FireBase Notifications

