



Firebase



Victor Alejandro Pinzón Ustate
Bases de datos
Maestría en Analítica de Datos
Abril 2024



¿Qué es Firebase?

Firebase de Google es una plataforma en la nube para el **desarrollo de aplicaciones web y móvil**. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo.

Fue creada en 2011 **pasó a ser parte de Google en 2014**, comenzando como una base de datos en tiempo real.

Sin embargo, se añadieron más y más funciones que, en parte, permitieron agrupar los SDK (kit de desarrollo de software) de productos de Google con distintos fines, facilitando su uso.

Nota: Un kit de desarrollo de software (SDK) es un conjunto de herramientas proporcionado usualmente por el fabricante de una plataforma de hardware, un sistema operativo (SO) o un lenguaje de programación.

¿Para qué sirve Firebase?

- Facilitar la creación de aplicaciones webs, móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad requerida.
- Simplificar las tareas de gestión a una **misma plataforma**.
- Es especialmente interesante para que los desarrolladores no necesiten dedicarle tanto tiempo al backend, tanto en cuestiones de desarrollo como de mantenimiento.



Funciones de Firebase

Firebase dispone de diferentes funcionalidades, que se pueden dividir básicamente en **3 grupos**:

Desarrollo (Develop): Como su nombre indica, incluye los servicios necesarios para el desarrollo de un proyecto de aplicación móvil o web. Estos contribuyen a que el proceso sea más rápido, puesto que se dejan determinadas actividades a mano de Firebase, mientras que otras permiten optimizar diversos aspectos para conseguir la calidad deseada.

Crecimiento (Grow): Contempla tanto la gestión de aquellos que ya son usuarios de esta, como herramientas para la captación de nuevas audiencias.

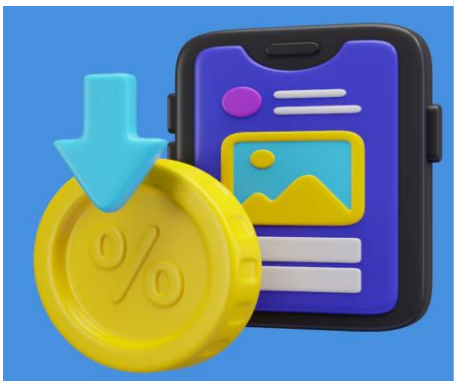
Monetización (Earn): En este caso, la búsqueda de ganancias viene ligada a la publicidad que se puede insertar en las aplicaciones, consiguiendo que los usuarios de estas reciban anuncios relevantes en función de la segmentación que se le haya dado a la campaña.



REALTIME DATABASE -
AUTENTICACIÓN DE USUARIOS
- ALMACENAMIENTO EN LA
NUBE - CRASH REPORTING -
TEST LAB - REMOTE CONFIG -
CLOUD MESSAGING - HOSTING



NOTIFICATIONS - APP
INDEXING - DYNAMIC LINKS,
INVITES - ADWORDS



AdMob



Firestore Analytics

Controla diversos parámetros, obtiene mediciones variadas desde un mismo panel de manera gratuita.

Es compatible con iOS, Android, C++ y Unity y, entre otras funciones, permite:

- Obtener mediciones y análisis de los eventos en la aplicación con informes ilimitados con hasta 25 atributos.
- Comprobar el rendimiento de eventos, notificaciones y campañas publicitarias en redes, basándose en el comportamiento de los usuarios.
- Conocer al usuario con información segmentada. Por ejemplo, se pueden obtener datos estadísticos de idioma, dispositivo de acceso, edad, género, ubicación, etc.

También se consiguen insights de uso y fidelidad hacia la app.

REALTIME DATABASE - CARACTERISTICAS



- Es una de las herramientas más destacadas y esenciales de Firebase son las **bases de datos en tiempo real**.
- Estas se alojan en la **nube**, son **No SQL** y almacenan los datos como **JSON**.
- Permiten alojar y disponer de los datos e información de la aplicación en tiempo real, manteniéndolos actualizados, aunque el usuario no realice ninguna acción.
- Firebase **envía automáticamente eventos** a las aplicaciones cuando los datos cambian, almacenando los datos nuevos en el disco.
- Aunque no hubiera conexión por parte de un usuario, sus datos estarían disponibles para el resto y los cambios realizados se sincronizarían una vez restablecida la conexión.
- Definición de reglas de seguridad para controlar el acceso.
- Compatibilidad multiplataforma IOS, ANDROID Y WEB.

Nota: JSON es un formato de texto que forma parte del sistema de JavaScript y que se deriva de su sintaxis, pero no tiene como objetivo la creación de programas, sino el acceso, almacenamiento e intercambio de datos. Usualmente es conocido como una alternativa al lenguaje XML.

REALTIME DATABASE – FUNCIONAMIENTO

Funciona utilizando una arquitectura cliente-servidor que permite la sincronización en tiempo real de los datos entre los clientes y la base de datos en la nube.



1. **Cliente:** Cada cliente de la aplicación se conecta utilizando una biblioteca de Firebase específica para la plataforma que esté utilizando.
2. **Base de datos en la nube:** Es una base de datos en la nube alojada por Google. Almacena los datos de la aplicación en una estructura de árbol JSON.
3. **Solicitud de datos:** Cuando un cliente necesita acceder a datos en la base de datos (por ejemplo, para leer o escribir datos), envía una solicitud al servidor de Firebase.
4. **Actualización en tiempo real:** Cuando se realiza un cambio en los datos de la base de datos desde cualquier cliente, Realtime Database envía actualizaciones a todos los clientes conectados en tiempo real. Esto garantiza que todos los clientes tengan los datos más recientes.
5. **Reglas de seguridad:** Antes de permitir cualquier operación de lectura o escritura en la base de datos, Realtime Database evalúa las reglas de seguridad definidas por el desarrollador para asegurarse de que el cliente tenga permiso para realizar esa operación.
6. **Sincronización bidireccional:** La sincronización en tiempo real es bidireccional, lo que significa que los cambios realizados en cualquier cliente se reflejan automáticamente en todos los demás clientes conectados, y viceversa.

En resumen, Realtime Database proporciona a los desarrolladores una forma sencilla de sincronizar datos en tiempo real entre múltiples clientes y la base de datos en la nube, lo que facilita la creación de aplicaciones colaborativas y en tiempo real.

FORTALEZAS	DEBILIDADES
<ul style="list-style-type: none">1. Sincronización en tiempo real: de datos entre clientes y la base de datos en la nube, lo que facilita la creación de aplicaciones en tiempo real y colaborativas.2. Fácil integración: con otras herramientas de Firebase, como Firebase Authentication, Firebase Hosting y Firebase Cloud Messaging, lo que facilita la creación de aplicaciones completas y multifuncionales.3. Escalabilidad es capaz de manejar una gran cantidad de usuarios simultáneos y escalarse automáticamente según las necesidades del proyecto.4. Base de datos en la nube: no es necesario administrar la infraestructura subyacente, lo que reduce la carga operativa para los desarrolladores.5. Compatibilidad multiplataforma: incluidas iOS, Android y la web, lo que permite a los desarrolladores crear aplicaciones multiplataforma con la misma base de datos en tiempo real.	<ul style="list-style-type: none">1. Estructura de datos limitada: está estructurado como un árbol JSON plano, lo que puede resultar limitante para ciertos tipos de datos o consultas complejas.2. Reglas de seguridad complejas: Aunque son flexibles y potentes, pueden ser complejas de configurar correctamente, especialmente para aplicaciones con requisitos de seguridad avanzados.3. Costos variables: opera con un modelo de precios basado en el uso, lo que significa que los costos pueden aumentar con el crecimiento de la aplicación y el uso intensivo de la base.4. Limitaciones de consultas: tienen ciertas limitaciones en comparación con las bases de datos SQL, lo que puede dificultar la implementación de ciertas consultas complejas.5. Dependencia de la infraestructura de Google: los desarrolladores están vinculados a la infraestructura y las políticas de Google, lo que puede ser un factor a considerar en términos de privacidad y soberanía de datos.

Opción sólida para muchas aplicaciones que requieren sincronización en tiempo real de datos entre múltiples clientes, pero es importante evaluar cuidadosamente sus fortalezas y debilidades en relación con los requisitos específicos del proyecto.

REALTIME DATABASE – CASOS DE EXITO



The
Economist



Firestore
Google

The New York Times



todoist

REALTIME DATABASE – CASOS DE EXITO

1. **WhatsApp** que se basa en la sincronización en tiempo real para permitir a los usuarios enviar y recibir mensajes instantáneos.
1. **The New York Times** es uno de los diarios estadounidenses con mayor credibilidad que utiliza Firebase. Con aplicación móvil y sitio web, también ha lanzado la aplicación Times Reader para lectores de escritorio con la colaboración de Microsoft. The New York Times utiliza con éxito Firebase, principalmente Cloud Firestore para utilizar Oak Editor sin problemas. Firebase también ayuda a este periódico a realizar fácilmente modificaciones en la base de datos, recopilar selecciones y limpiar el cursor cuando un usuario cierra una página de edición.
1. **The Economist** periódico británico semanal se publicó inicialmente en 1843 en forma impresa, pero ahora tiene presencia digital en todo el mundo. Depende de Firebase para respaldar sus pilas tecnológicas de backend. Sí, los productos de Firebase ayudan a The Economist a realizar sus operaciones del lado del servidor para continuar con su presencia digital sin problemas.
1. **Todoist** con más de 187.000 reseñas en Google Play y una calificación de 4,8 estrellas en la App Store, Todoist muestra otra historia de éxito de Firebase. Este es básicamente un software que ayuda a los equipos a organizar sus tareas y prioridades de manera eficiente. Todoist ha atendido más de 150 millones de proyectos. Este software se beneficia de Firebase, sus desarrolladores pueden conectar Todoist con Firebase para acceder a Firestore y Realtime Database para continuar con las actualizaciones y modificaciones de datos sin problemas.
1. **AliExpress** es una tienda minorista en línea con sede en China que se hizo pública en 2010. Los clientes suelen comprar artículos electrónicos y de moda en esta tienda minorista. Firebase ayuda a AliExpress con su servicio de mensajería en la nube. De hecho, con la ayuda de Firebase Cloud Messaging (FCM), la tasa de apertura de la aplicación web de AliExpress se ha mejorado hasta el 93,4%.
1. **Rubbish** es básicamente una aplicación de iOS que ayuda a las personas a retratar las tendencias de la basura y cómo estos datos pueden ayudar a las comunidades. Esta toma una foto de cada material de basura recolectado por un voluntario. Solo necesita conectar esto con su teléfono inteligente. Los cofundadores Elena Guberman y Emin Israfil compartieron que confiaron completamente en Firebase para crear esta interesante aplicación en 2018.



REALTIME DATABASE – CASOS DE EXITO



7. **Halfbrick** es una empresa de videojuegos con sede en Australia que nació en 2001. Con una duración mínima de 90 segundos, Halfbrick ha lanzado numerosos videojuegos populares. Los videojuegos populares de Halfbrick son Fruit Ninja 2, Monster Dash, Battle Racing Stars y Dan the Man. Ellos confían en Firebase para conocer el comportamiento de los usuarios objetivo y obtener la configuración remota. Esta empresa notó un aumento del 20% en el índice de retención de usuarios activos semanales cuando utilizó Firebase Predictions.
7. **Fabulous** es una aplicación de autocuidado que hace hincapié en mejorar los hábitos para lograr objetivos. Esta aplicación tiene 1,2 millones de usuarios activos al mes debido a sus increíbles características. Este éxito no fue posible sin Firebase. Fabulous utiliza Firebase para una fácil autenticación e incluso para invitar a amigos. Firebase envía atractivos correos electrónicos personalizados a sus amigos para que comience a usar esta aplicación. Del mismo modo, los procesos de incorporación personalizados también ayudaron a Fabulous a lograr una tasa de retención dos veces mayor.
7. **OneFootball** es una empresa de medios alemana que ofrece noticias de fútbol a los lectores. Está disponible tanto en formato web como de aplicación móvil para consultar resultados en vivo, noticias de fútbol y estadísticas. Según el jefe de productos de OneFootball, Maxime Blanc-Strauss, la empresa experimentó un aumento del 5% en las interacciones diarias de los usuarios después de empezar a usar Firebase.
7. **Yummly** es una plataforma delicada que muestra sugerencias de recetas de comida para mejorar el sabor de diferentes platos. Esta plataforma de recomendación de recetas utiliza Firebase para invitar a los usuarios. Afortunadamente, con App Invites, Yummly obtuvo un incremento del 11% en el índice de instalación de aplicaciones, y el índice de aceptación mensual de estas invitaciones fue del 37%.
7. **Playbuzz** con 13.000 editores, es otra plataforma que ha utilizado Firebase la cual brindó asistencia efectiva en el nombre de la frecuencia de anuncios y notificaciones automáticas al equipo de Playbuzz. Con Firebase, el equipo de monetización de Playbuzz obtuvo el control total de las campañas en solo 2 días.

REALTIME DATABASE – REQUISITOS TÉCNICOS



Para utilizar Firebase Realtime Database son relativamente simples y están diseñados para ser accesibles para una amplia gama de desarrolladores.

1. Conexión a Internet: Es necesario que los dispositivos que ejecutan la aplicación tengan acceso a Internet ya que la base de datos está alojada en la nube.
2. Plataforma de desarrollo compatible: Entorno de desarrollo adecuado para cada plataforma, como Xcode para iOS, Android Studio para Android y un navegador web para aplicaciones web.
3. SDK de Firebase: Correspondiente a la plataforma de desarrollo utilizada. Firebase proporciona SDKs para iOS, Android, JavaScript (para aplicaciones web) y otras plataformas.
4. Cuenta de Firebase: Para crear un proyecto y configurar Firebase Realtime Database. Esta cuenta es gratuita y se puede crear en el sitio web de Firebase.
5. Conocimientos de programación: adecuado para la plataforma de desarrollo elegida (por ejemplo, Swift o Objective-C para iOS, Java o Kotlin para Android, JavaScript para aplicaciones web).
6. Capacidad de gestión de datos: tener un entendimiento básico de cómo gestionar datos y estructuras de datos en una base de datos en tiempo real.

Una vez cumplidos estos requisitos, los desarrolladores pueden comenzar a integrar Firebase Realtime Database en sus aplicaciones para agregar funcionalidades de base de datos en tiempo real.

REALTIME DATABASE – EJEMPLO CÓDIGO



APP web utilizando JavaScript:

Aplicación de lista de tareas donde los usuarios pueden agregar nuevas tareas y verlas en tiempo real.

Para esto, primero necesitamos configurar Firebase y luego interactuar con la base de datos para realizar operaciones como agregar tareas y escuchar cambios en la lista de tareas.

Este código HTML crea una interfaz simple con un campo de entrada para agregar tareas, un botón para agregar tareas, y una lista de tareas que se actualiza en tiempo real.

En el bloque `<script>`, primero inicializamos Firebase con nuestra configuración. Luego, creamos una referencia a la colección de tareas en la base de datos. La función `addTask()` agrega una nueva tarea a la base de datos cuando se hace clic en el botón "Add Task".

Finalmente, utilizamos `tasksRef.on('value', ...)` para escuchar cambios en la colección de tareas y actualizar la lista de tareas en tiempo real.

```
// Initialize Firebase
const firebaseConfig = {
  apiKey: "YOUR_API_KEY",
  authDomain: "YOUR_AUTH_DOMAIN",
  databaseURL: "YOUR_DATABASE_URL",
  projectId: "YOUR_PROJECT_ID",
  storageBucket: "YOUR_STORAGE_BUCKET",
  messagingSenderId: "YOUR_MESSAGING_SENDER_ID",
  appId: "YOUR_APP_ID"
};
firebase.initializeApp(firebaseConfig);

// Reference to the tasks collection in the database
const tasksRef = firebase.database().ref('tasks');

// Function to add a new task to the database
function addTask() {
  const taskInput = document.getElementById('taskInput');
  const taskName = taskInput.value.trim();
  if (taskName !== '') {
    tasksRef.push({
      name: taskName,
      completed: false
    });
    taskInput.value = '';
  }
}

// Function to render tasks
function renderTasks(snapshot) {
  const taskList = document.getElementById('taskList');
  taskList.innerHTML = '';
  snapshot.forEach(childSnapshot => {
    const task = childSnapshot.val();
    const li = document.createElement('li');
    li.textContent = task.name;
    if (task.completed) {
      li.style.textDecoration = 'line-through';
    }
    taskList.appendChild(li);
  });
}

// Listen for changes in the tasks collection
tasksRef.on('value', snapshot => {
  renderTasks(snapshot);
});
</script>
</body>
</html>
```