

UNIVERSIDAD DON BOSCO



Tema:

Foro 1.

Materia:

Desarrollo de Software para Móviles.

Titular de Catedra:

Ing. Alexander Alberto Sigüenza Campos.

Integrantes:

Rivas Serrano, Omar Alexander

RS060867

Fecha de entrega:

29 de octubre de 2023

Contenido

| | |
|---|----|
| Historia | 3 |
| Firebase | 3 |
| SQLite | 3 |
| Conceptos | 5 |
| Firebase: | 5 |
| SQLite: | 5 |
| Ventajas y desventajas | 6 |
| Firebase: | 6 |
| SQLite: | 7 |
| Mejor Opción para Implementar en Android: | 10 |
| Comparación y Conclusiones: | 12 |
| Conclusiones | 14 |
| Bibliografía | 15 |

Historia

Firestore

Esta herramienta fue fundada en el año 2011 por Firebase, Inc., como un SDK de chat para móviles que llevaba por nombre “Envolv”, el cual no tuvo el éxito esperado. Por este motivo, posteriormente se convertiría en otro producto llamado “Firestore Realtime Database”, que durante dos años consecutivos (2012 – 2013) logró recaudar grandes cifras de dólares (Muradas, 2021).

Debido a su éxito en el año 2014 salió al mercado “Firestore Hosting” y “Firestore Authentication”, este mismo año la compañía fue comprada por Google, que entendiendo la necesidad y/o demanda de los desarrolladores de aplicaciones móviles maduraron la idea, aunque no fue hasta el año 2016 que lograron convertir a Firestore en una plataforma unificada y con una serie de productos para cubrir el objetivo principal de apoyar y facilitar sus labores a los desarrolladores. En ese lanzamiento Firestore incluyó:

- Google Cloud Platform
- Admob
- Google Ads

Y Firestore Cloud Messaging pasó a reemplazar a Google Cloud Messaging, permitiendo enviar notificaciones push a iOS, Android y Web. En el año 2017 se incluyen en la suite Firestore a Crashlytics y Fabric dos empresas de Twitter que Google compró en este momento. En este mismo orden de ideas, para el último trimestre del 2017 llega Cloud Firestore quien sustituyó a Firestore Realtime Database como un novedoso producto en tiempo real de base de datos.

SQLite

Cuando D. Richard Hipp trabajaba desarrollando Software para la fuerza naval de los Estados Unidos, comenzó a desarrollar SQLite, según él por una necesidad personal para su

uso personal. En enero de 2000 D. Richard Hipp estaba trabajando con su equipo de la General Dynamics en la Fuerza naval de los Estados Unidos, en un proyecto de software, el cual se conectaba a una base de datos Informix, el motor funcionaba muy bien, pero habían tenido problemas para hacer una reconfiguración cuando el sistema se reiniciaba. Luego cambiaron a PostgreSQL, pero administrar la base de datos era un poco más complejo. Fue en ese momento cuando surgió la idea de escribir un simple motor de base de datos SQL que permitiera leer los archivos del disco duro, y luego ser llamados en diferentes solicitudes.

Cinco meses más tarde comenzó a escribir las primeras versiones de lo que se conoce hoy como SQLite, con el pensamiento de que sería útil en algún problema similar. A diferencia de los sistemas de gestión de base de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción (EcuRed, s.f.).

En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB. El autor de SQLite ofrece formación, contratos de soporte técnico y características adicionales como compresión y cifrado.

Conceptos

Firestore:

Firestore básicamente es una plataforma móvil diseñada y creada por Google, teniendo como principal función desarrollar y facilitar la creación de aplicaciones para dispositivos móviles que cuenten con una alta calidad a pesar de su rápida elaboración; esto con la finalidad de que se pueda incrementar la base de datos de usuarios y de esta manera incrementar la monetización de dicha app (ganar más dinero).

Esta plataforma se encuentra alojada en la nube y, por ende, está disponible para diferentes plataformas como Android, iOS, y web. Así mismo, cuenta con diversas funciones para que cualquier desarrollador pueda combinar y adaptar la plataforma a medida de sus necesidades (Muradas, 2021).

SQLite:

SQLite es una herramienta de software libre, que permite almacenar información en dispositivos empujados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular (Guru, s.f.).

SQLite implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, y lo más importante es que se puede usar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente.

Ventajas y desventajas

Firestore:

Las ventajas de la tecnología de Firestore son (SEIDOR, 2023):

- Inicio gratuito y escalable: uno de los aspectos más destacables de Firestore es que la mayoría de sus módulos son de uso gratuito siempre y para cualquier tamaño. Hay algunos que limitan la capacidad en el plan gratuito, pero no debemos preocuparnos por ello, puesto que, si a futuro nuestra app tiene éxito y es necesario aumentar las capacidades de nuestro servidor, podemos gestionarlo fácilmente con Firestore.
- Es multiplataforma, ya que se encuentra disponible para las principales plataformas móviles (Android e iOS) y para la web. Además, otras tecnologías comienzan a tener también soporte como Node, Flutter, Unity...
- Ofrece un gran poder de crecimiento al gestionar fácilmente todos los usuarios de las aplicaciones, y potenciar el crecimiento de estos a través del envío de notificaciones e invitaciones.
- Se trata de una herramienta ágil, sencilla y nos permite olvidarnos de la infraestructura. Firestore nos proporciona APIs intuitivas integradas todas en un sólo SDK, lo que nos permite centrarnos en resolver los problemas de nuestros clientes en vez de focalizarnos en crear una infraestructura compleja. Podemos ganar dinero: sí sí, has leído bien, mediante el uso de AdMob podemos dotar a nuestra app con anuncios y publicidad de manera muy sencilla.
- Crecimiento continuo: no perdáis de vista las novedades de Firestore, ya que van creando nuevos módulos y éstos cada vez son compatibles con nuevas tecnologías.

Firebase también tiene algunas desventajas que debemos tener en cuenta a la hora apostar por esta plataforma (SEIDOR, 2023):

- No es open source: Firebase no es una opción de código abierto, por lo que los desarrolladores no pueden hacer adaptaciones si lo desean.
- No dispone de bases de datos relacionales: sus dos opciones, Firestore y Realtime Database tienen una estructura NoSQL. Además, se hace complicado a veces ejecutar consultas complejas.
- No todos los servicios son gratuitos: si bien como hemos comentado antes hay un conjunto interesante de ellos que sí lo son, hay otros que comienzan siendo gratuitos y tan pronto el usuario exceda el límite establecido, se efectuarán los cargos bajo el Plan de pago. Por ello, es necesario tener una previsión de cómo escalará nuestra aplicación para tener en cuenta que en algún momento podemos tener costes por el uso de alguno de sus módulos.
- Se ejecuta sólo el Google Cloud: al ser parte de Google no podía ser de otra forma. No tenemos ninguna opción de ejecutar Firebase en otros proveedores en la nube como AWS o Azure.
- Requiere tiempo de aprendizaje: como con toda tecnología, debemos aprender cómo usar Firebase. Si bien proporciona gran documentación sobre cómo integrar cada módulo y ejemplos de cada uno de ellos, tenemos que dedicar tiempo a conocer cómo integrarlos en nuestra aplicación.

SQLite:

SQLite es muy popular debido a las múltiples ventajas que ofrece a la hora de desarrollar aplicaciones con base de datos incorporada. Sin embargo, presenta algunos inconvenientes que es conveniente conocer y que le impiden ser el motor ideal para muchos tipos de proyectos.

Veamos cuales son los puntos fuertes y los inconvenientes que presenta SQLite (Caro, s.f.):

- Multiplataforma: SQLite funciona en cualquier plataforma por lo que es ideal para desarrollar aplicaciones, apps y webs que funcionen en Linux, Windows, Android o sistemas Apple.
- Es muy ligero: SQLite es muy ligero, integrándose con la propia aplicación. La base de datos se guarda en un solo archivo y su peso es muy pequeño en comparación al de otras alternativas existentes. Este peso ligero de la base de datos la convierte en una alternativa ideal para utilizar en dispositivos que tienen capacidad limitada de almacenamiento como Smart TV, cámaras, celulares, tablets...
- Reducción de costes: El código de SQLite es gratuito y público, pudiendo utilizarse de forma libre en cualquier tipo de proyecto. Además, se puede modificar para adaptarlo a las necesidades propias de cada proyecto.
- Sencillo de utilizar: Utilizar SQLite es muy sencillo al no tener que montar un servidor para poder gestionar la base de datos. Esto facilita mucho el proceso de desarrollo al no requerir una compleja instalación y configuración. Además, está basado en el lenguaje de consultas SQL, por lo que si domina se podrá utilizar SQLite sin ninguna dificultad (utiliza la sintaxis propia de SQL con unas pequeñas variaciones sencillas de aprender).
- Alto rendimiento: SQLite se caracteriza por funcionar de forma muy rápida y eficiente. Cualquier consulta u operación que se realice se resuelve de manera eficiente y casi instantánea, consiguiendo que el funcionamiento de la app o web ofrezca un gran rendimiento.

Entre sus desventajas, podemos encontrar (Caro, s.f.):

- Tipos de datos limitados: SQLite solo trabaja con cuatro tipos de datos, INTEGER, REAL, BLOB y TEXT lo que supone una gran limitación para muchos proyectos.
- No es escalable: Quizás, uno de los grandes hándicaps de SQLite es que escala fácilmente, es decir, que si los datos aumentan de forma considerable la aplicación deja de ser eficiente, o si se quiere ampliar la capacidad o funciones de la base de datos será complicado realizarlo.
- Deficiente gestión de usuarios: Uno de los puntos clave en la gestión de una base de datos es la gestión y permiso de usuarios, algo que SQLite no posee y que lo limita de forma severa.
- No cuenta con funciones de seguridad: Al carecer de funciones de seguridad no es una gran alternativa en un entorno donde la seguridad cada vez tiene un papel más importante.
- Monousuario: No permite que más de un usuario interactúe al mismo tiempo con la base de datos.

Mejor Opción para Implementar en Android:

Para el proyecto de la aplicación móvil para la gestión de registros en la clínica veterinaria, sería recomendable utilizar Firebase como base de datos. Esto por las siguientes razones:

- **Autenticación de Usuarios:** Firebase proporciona un sistema de autenticación robusto y seguro que permite la creación y gestión de cuentas de usuario. Esto es esencial para cumplir con el requisito de acceso a la aplicación mediante usuario y contraseña.
- **Tiempo Real:** Firebase ofrece una base de datos en tiempo real que es ideal para aplicaciones que requieren actualizaciones instantáneas y sincronización en tiempo real. Esto es muy útil en una clínica veterinaria donde es crucial tener acceso inmediato a la información actualizada de pacientes, doctores y citas.
- **Capacidad de Almacenamiento:** Además de la base de datos en tiempo real, Firebase proporciona opciones de almacenamiento en la nube que permiten guardar y recuperar archivos como imágenes o documentos, lo que puede ser útil para almacenar historiales de citas o cualquier otro tipo de archivo relacionado.
- **Seguridad y Control de Acceso:** Firebase ofrece opciones de autenticación y control de acceso que permiten definir quién tiene acceso a qué partes de la base de datos. Esto es esencial para mantener la confidencialidad de la información de los pacientes y garantizar que solo personal autorizado pueda acceder a ciertos datos.
- **Escalabilidad:** Firebase es escalable y puede manejar un gran número de usuarios y datos, lo que es importante si la clínica veterinaria tiene un crecimiento futuro previsto.

- Fácil Integración con Android: Firebase está completamente integrado con el ecosistema de desarrollo de Android y proporciona bibliotecas y herramientas específicas para facilitar la implementación.

SQLite, a pesar de ser una excelente opción para aplicaciones locales o de un solo usuario, no sería la mejor elección para el proyecto de la clínica veterinaria. Esto se debe a sus limitaciones en cuanto a acceso múltiple y sincronización en tiempo real, así como su incapacidad para proporcionar autenticación y control de acceso avanzados. Además, SQLite puede enfrentar desafíos en términos de escalabilidad y no es la opción más adecuada para situaciones donde se requiere sincronización de datos entre múltiples dispositivos.

Comparación y Conclusiones:

En la siguiente tabla podemos observar una comparativa de características entre SQLite y Firebase.

| Característica | SQLite | Firebase |
|---|--|--|
| Tipo de Base de Datos | Relacional | NoSQL (Basada en Documentos) |
| Acceso Múltiple y Sincronización en Tiempo Real | Limitado | Sí (Base de Datos en Tiempo Real) |
| Autenticación y Control de Acceso | Requiere Implementación | Integrado y Configurable |
| Escalabilidad | Limitada (Adecuada para Aplicaciones de un solo Usuario o Locales) | Alta (Adecuada para Aplicaciones con Muchos Usuarios o Escenarios de Escala) |
| Almacenamiento de Datos | Local (Archivo en el Dispositivo) | En la Nube (Accesible desde Múltiples Dispositivos) |
| Seguridad | Depende de la Implementación | Integrada con Opciones Avanzadas de Configuración |
| Integración con Android | Compatible, pero requiere configuración adicional | Totalmente Integrada con Herramientas y Bibliotecas Específicas |
| Respaldo Automático y Restauración de Datos | No | Sí (Capacidades de Respaldos Automáticos) |

Las diferencias clave entre SQLite y Firebase radican en su arquitectura y capacidades. SQLite es una base de datos relacional que se almacena localmente en el dispositivo, lo que lo hace adecuado para aplicaciones de un solo usuario o entornos locales. Sin embargo, su acceso múltiple y sincronización en tiempo real son limitados, lo que podría no ser ideal para un proyecto como el de la clínica veterinaria, donde múltiples doctores y personal necesitan acceder y actualizar registros simultáneamente. Además, SQLite requiere la implementación manual de autenticación y control de acceso, lo que puede añadir complejidad al desarrollo.

Por otro lado, Firebase ofrece una base de datos en tiempo real basada en documentos, lo que facilita la sincronización instantánea de datos entre múltiples dispositivos. Esto es fundamental para una clínica veterinaria donde la información debe estar siempre actualizada. Firebase también integra un sistema de autenticación y control de acceso avanzado, lo que garantiza la seguridad y confidencialidad de los datos de los pacientes. Su capacidad de escalabilidad y la posibilidad de respaldar automáticamente los datos son características que hacen de Firebase la opción más adecuada para este proyecto.

Conclusiones

En conclusión, para el proyecto de desarrollo de software destinado a la clínica veterinaria Santa Bárbara, la elección entre SQLite y Firebase se inclina claramente hacia Firebase. Esta plataforma de desarrollo de aplicaciones móviles y web de Google ofrece una serie de ventajas cruciales para abordar las necesidades específicas de la clínica. La capacidad de proporcionar autenticación de usuarios, acceso en tiempo real, seguridad avanzada y escalabilidad hacen de Firebase la opción más indicada. Además, su integración perfecta con el ecosistema de desarrollo de Android facilitará la implementación y mejora la eficiencia en el desarrollo.

Por otro lado, SQLite, aunque es una opción valiosa para aplicaciones locales y de un solo usuario, presenta limitaciones significativas para este proyecto. Su incapacidad para manejar accesos múltiples y sincronización en tiempo real, así como la necesidad de implementar manualmente autenticación y control de acceso, lo relegan a aplicaciones con requisitos menos complejos y exigentes en términos de interconexión y seguridad. Firebase, en cambio, proporciona una solución integral que se alinea perfectamente con los objetivos y requisitos de la clínica veterinaria Santa Bárbara.

Bibliografía

Caro, C. (s.f.). Obtenido de Ventajas y desventajas de SQLite: <https://ecohosting.cl/ventajas-y-desventajas-de-sqlite/3437/>

EcuRed. (s.f.). Obtenido de SQLite: <https://www.ecured.cu/SQLite>

Guru, S. (s.f.). Obtenido de SQLite: La Base de Datos Embebida: <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>

Muradas, Y. (22 de Junio de 2021). Obtenido de Qué es Firebase: Conoce la plataforma de Google: <https://openwebinars.net/blog/que-es-firebase-de-google/>

SEIDOR. (20 de Febrero de 2023). Obtenido de ¿Qué es Firebase? ¿Qué ventajas ofrece en 2023 a nuestras apps?: <https://www.seidor.com/blog/firebase-que-es>