

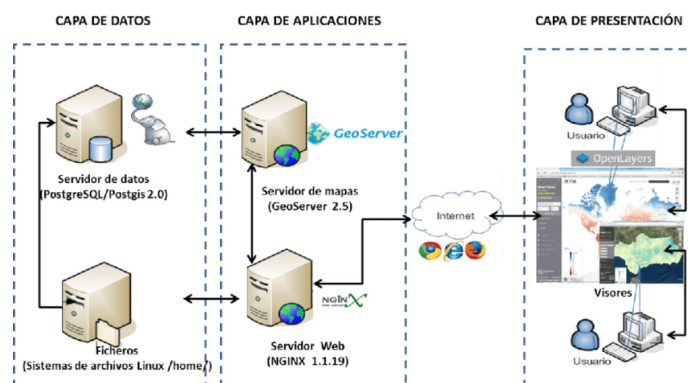
## DISEÑO DE LA SOLUCIÓN

### 1. Arquitectura del sistema

En los inicios de la informática, la programación era bastante difícil para la mayoría de las personas, pero con el tiempo se han ido descubriendo y desarrollando formas y guías generales para resolver problemas. A las mismas se les ha denominado arquitectura de software, por la semejanza con los planos de un edificio, debido a que la arquitectura indica la estructura, funcionamiento e interacción entre las distintas partes del software<sup>1</sup>.

Abordada desde el punto de vista técnico, la arquitectura de software es el diseño de más alto nivel de la estructura de un sistema:

- También denominada arquitectura lógica, integra un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software.
- Se selecciona y diseña con base en requisitos y restricciones.
  - Objetivos: Prefijados por el sistema, no solo son de tipo funcional, sino que también incluyen otros aspectos como el mantenimiento, la auditoría, flexibilidad e interacción con otros sistemas.
  - Restricciones: Limitaciones derivadas de las tecnologías disponibles.
- Define de manera abstracta los componentes que llevan a cabo tareas de cómputo, sus interfaces y la comunicación entre los mismos. La arquitectura general del sistema especifica las distintas particiones físicas del mismo, su descomposición lógica en subsistemas de diseño y la ubicación de cada subsistema en cada partición<sup>2</sup>.



<sup>1</sup> [https://es.wikipedia.org/wiki/Arquitectura\\_de\\_software](https://es.wikipedia.org/wiki/Arquitectura_de_software)

<sup>2</sup> <https://manuel.cillero.es/doc/metodologia/metrica-3/procesos-principales/dsi/actividad-1/>

## 2. Arquitectura del sistema Instadroid

### 2.1. Decisión de la arquitectura y justificación

Como bien he descrito a la hora de mostrar el diseño de la persistencia de datos, Instadroid necesita que los datos sean mostrados en tiempo real, por lo que cualquier cambio respecto a ellos deberá reflejarse automáticamente en la interfaz del cliente de la aplicación.

En este sentido, como ya he descrito, utilizaré Firebase ya que me otorga las herramientas necesarias a tal fin (se explicará Firebase de forma detallada en el apartado de herramientas de la implementación del proyecto).

Al utilizar Firebase, la arquitectura de Instadroid será una arquitectura en tiempo real que además utiliza arquitecturas basadas en servicios web y servicios.

Una **arquitectura en tiempo real** es un software cuyo correcto funcionamiento depende de los resultados producidos por el mismo y del instante de tiempo en el que se producen esos resultados<sup>3</sup>. Hay dos tipos:

- Soft: Es aquel que se degrada si los resultados no se producen correctamente de acuerdo con los requisitos especificados.
- Hard: Su funcionamiento será incorrecto si los resultados no se producen de acuerdo con la especificación temporal.

Firebase ofrece estos servicios de consulta de datos en tiempo real, por lo que, en nuestra implementación, cuando un usuario comenta una foto, otro usuario que esté viendo los comentarios de la misma foto verá el comentario aparecer al momento, sin necesidad de “refrescar” la pantalla.

Una **arquitectura orientada a servicios (SOA)** es una forma de pensar en servicios, su construcción y resultados. Cada servicio es una representación lógica de una actividad de negocio que tiene un resultado de negocio específico, por ejemplo, comprobar el crédito de un cliente<sup>4</sup>.

Firebase, ofrece distintos servicios, si un usuario quiere autenticarse lo hará contra el servicio de autenticación de Firebase. Si un usuario quiere subir una publicación los datos

---

<sup>3</sup> <https://ittgweb.wordpress.com/2016/05/29/3-7-diseno-de-software-de-arquitectura-de-tiempo-real/>

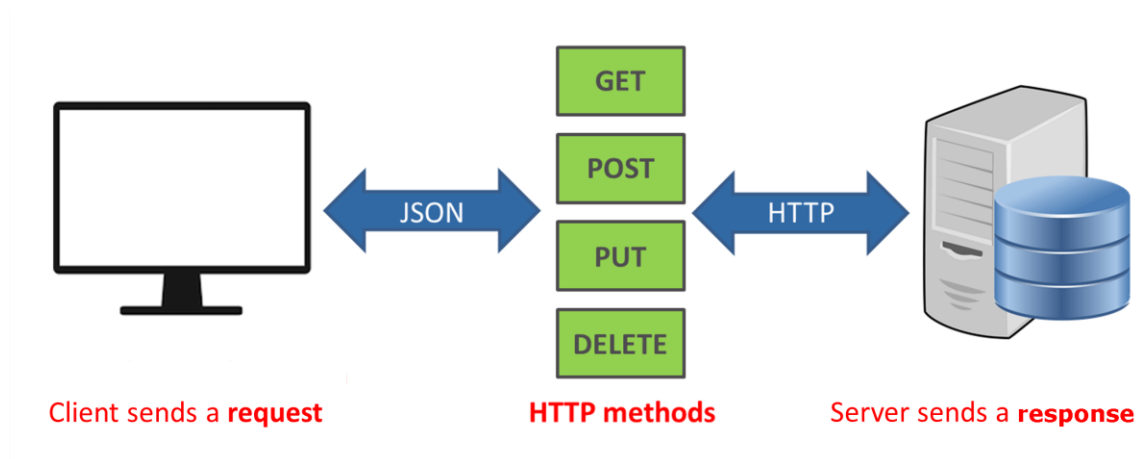
<sup>4</sup> [https://es.wikipedia.org/wiki/Arquitectura\\_orientada\\_a\\_servicios](https://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios)

de esta se subirán al servicio de base de datos (Firestore) y la imagen se subirá a el servicio de almacenamiento (Storage).

Finalmente, una **arquitectura orienta a servicios web** es muy similar a la anterior, utilizando un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Nos permiten intercambiar datos entre aplicaciones programadas en distintos lenguajes. Estos estándares utilizados han sido muchos, pero principalmente en los últimos años se utiliza el estándar REST (Representational State Transfer), que utiliza los verbos del protocolo HTTP proporcionando una API que utiliza los verbos para poder realizar distintas operaciones entre la aplicación que ofrece el servicio web y el cliente<sup>5</sup>.

Todo el conjunto de herramientas de Firebase se puede mirar como un servicio Web<sup>6</sup>, que expone una API Rest a la que nosotros le enviamos peticiones, solo que Firebase facilita librerías a los distintos lenguajes de programación con métodos que se encargan de realizar estas peticiones. Por lo que, en realidad, la comunicación con Firebase es con el protocolo HTTP, aunque nosotros no lo “escribamos” específicamente.

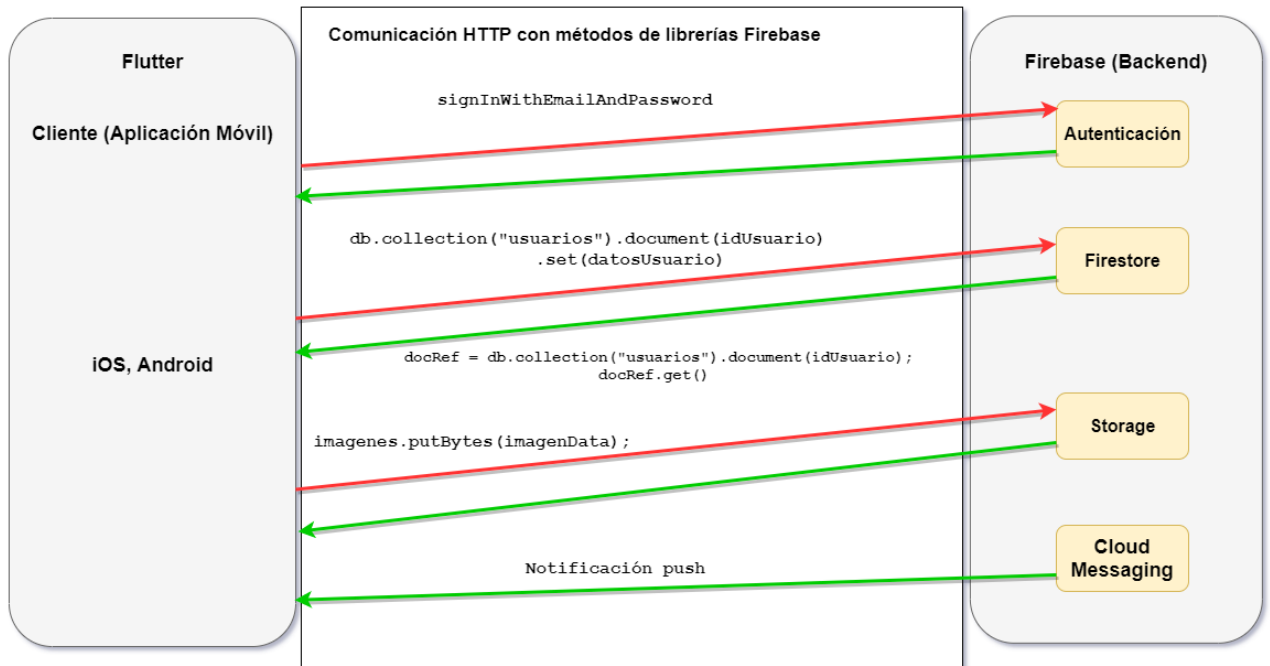
Así, por ejemplo, cuando hacemos un set a una colección de datos en realidad estamos mandando esos datos vía POST.



<sup>5</sup> [https://es.wikipedia.org/wiki/Servicio\\_web](https://es.wikipedia.org/wiki/Servicio_web)

<sup>6</sup> <https://es.stackoverflow.com/questions/267497/explicaci%C3%B3n-comunicaci%C3%B3n-cliente-android-con-la-servidor-en-firebase>

## 2.2. Diagrama de la arquitectura de Instadroid



Instadroid está conformado por un cliente (aplicación móvil), válido para ambos sistemas operativos (Android y iOS) y que utiliza como Backend Firebase. Como podemos ver, la parte de Firebase está dividida entre los distintos servicios que utilizaremos en el desarrollo de nuestra aplicación:

### ■ Autenticación

Cuando el usuario se autentica lo hace contra el servicio de autenticación de Firebase, como por ejemplo `signInWithEmailAndPassword`, que autenticará a un usuario con su email y contraseña.

### ■ Firestore

Se trata de la base de datos en tiempo real (ya entraremos en detalles en el punto oportuno). Cuando el usuario suba una publicación o actualice su información de usuario se registrará en esta base de datos NoSQL.

### ■ Storage

Es el servicio de almacenamiento de ficheros. Aquí almacenaremos todas las imágenes de la aplicación en vez de guardarlas en los propios documentos de la base de datos como cadena codificada a base 64.

La comunicación utilizada por las dos partes del sistema es el protocolo HTTP, como ya expliqué antes. He añadido una serie de métodos que utiliza Firebase en sus librerías para mostrar como cada uno de ellos es en realidad una petición HTTP.