

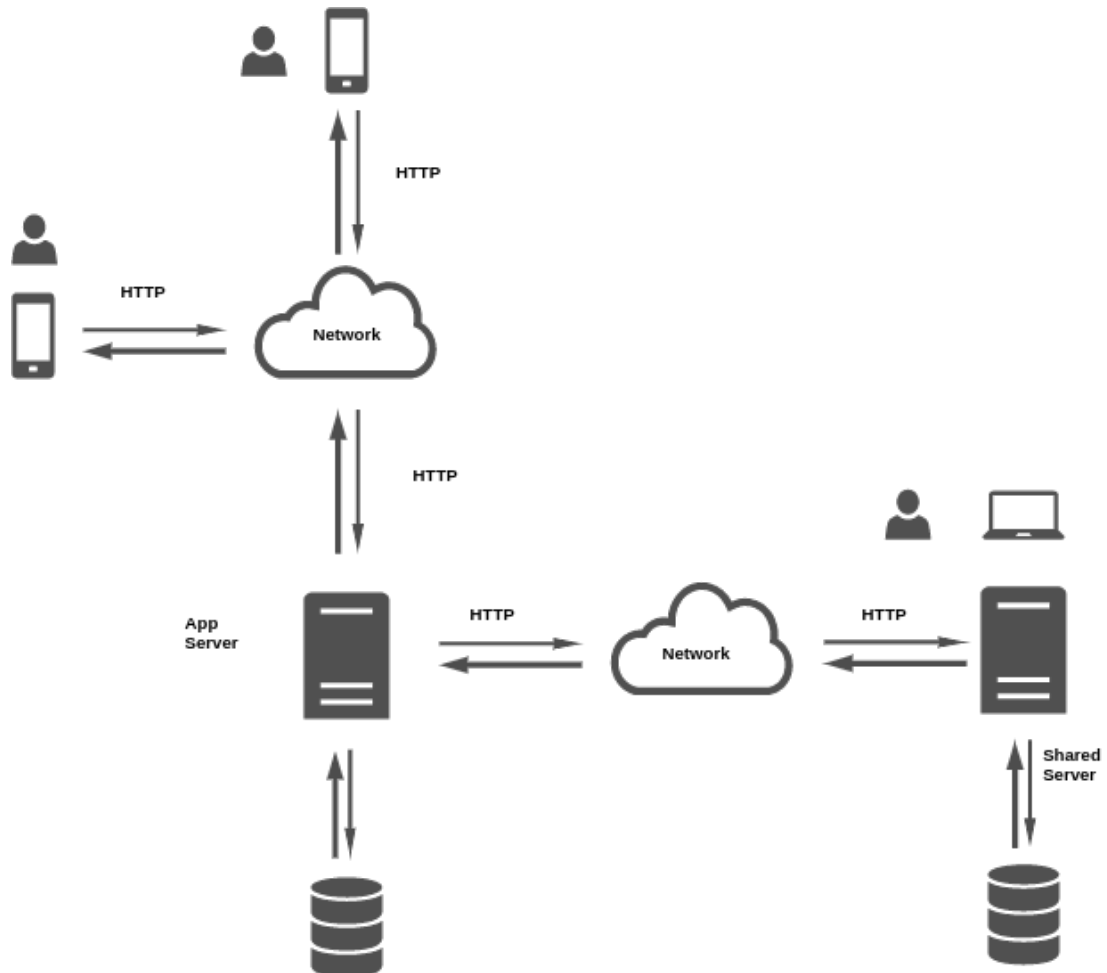
# App Server: Definición de Arquitectura

# Índice

1. Definición de Arquitectura	2
2. Diseño de aplicación	3

# 1. Definición de Arquitectura

El app server implementa una API REST utilizada para un sistema de venta de artículos. Esta API es consumida por una app de Android. A su vez se comunica con otro servidor que administra los pagos, envíos y cotización de estos últimos. se puede ver un esquema de comunicación en la siguiente imagen:



La comunicación entre los dispositivos móviles y el servidor se inicia mediante Google Firebase, una plataforma Backend-as-a-Service (BaaS), mediante el cual provee muchos servicios. Entre estos, se encuentra el servicio de autenticación y el de cloud messaging, que fueron utilizados en este proyecto.

Para implementar este servidor se utilizó el lenguaje de programación Python y Flask como web framework. Se utilizaron librerías para Flask/Python para realizar la autenticación de los usuarios, la serialización y deserialización de los datos y otras librerías comunes para este tipo de desarrollo.

La aplicación inicia ejecutando procesos de configuración del mismo. Luego, antes de empezar a recibir peticiones, se registra en el Shared Server (hay varios App Server que se van a comunicar con un solo Shared Server) y obtiene un id y un

token de refresco. Este id y token los guarda en un archivo para utilizarlos a partir de su primera conexión en adelante. Una vez que se registró con el shared server empieza a recibir peticiones de los dispositivos móviles. El proceso de registro de estos es el siguiente:

- El dispositivo móvil se autentica ante Firebase, obteniendo un token.
- El dispositivo móvil se registra entrando al endpoint de registro del app server, obteniendo un id
- Con este id y el token de Firebase accede al endpoint de login.
- El app server valida ante Firebase que el token sea válido, así como el id.
- Se obtiene un token que se utilizará para acceder al resto de los servicios del app server

Para la autenticación del app server se utilizan tokens del tipo JWT.

Se utiliza como base de datos MongoDB, una base de datos no relacional que otorga un modelo de datos flexible. Se utilizó un servicio cloud de MongoDB.

## 2. Diseño de aplicación

El diseño de la aplicación cuenta con tres capas principales:

- Controllors: Es el encargado de proveer el enrutamiento de la aplicación así como recibir las peticiones y preparar las respuestas.
- Services: Implementa la lógica de negocio así como las validaciones.
- Data: Esta capa se encarga de las consultas e inserciones a la base de dato.