



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

Carrera de Ingeniería en Electrónica y
Telecomunicaciones

Desarrollo de un Gateway Server para decodificar y retransmitir mensajes del sistema de alerta temprana en servicios de televisión digital terrestre

Autor: José Alejandro Salas Ladino

Director: Ing. Gonzalo Fernando Olmedo Cifuentes, PhD.

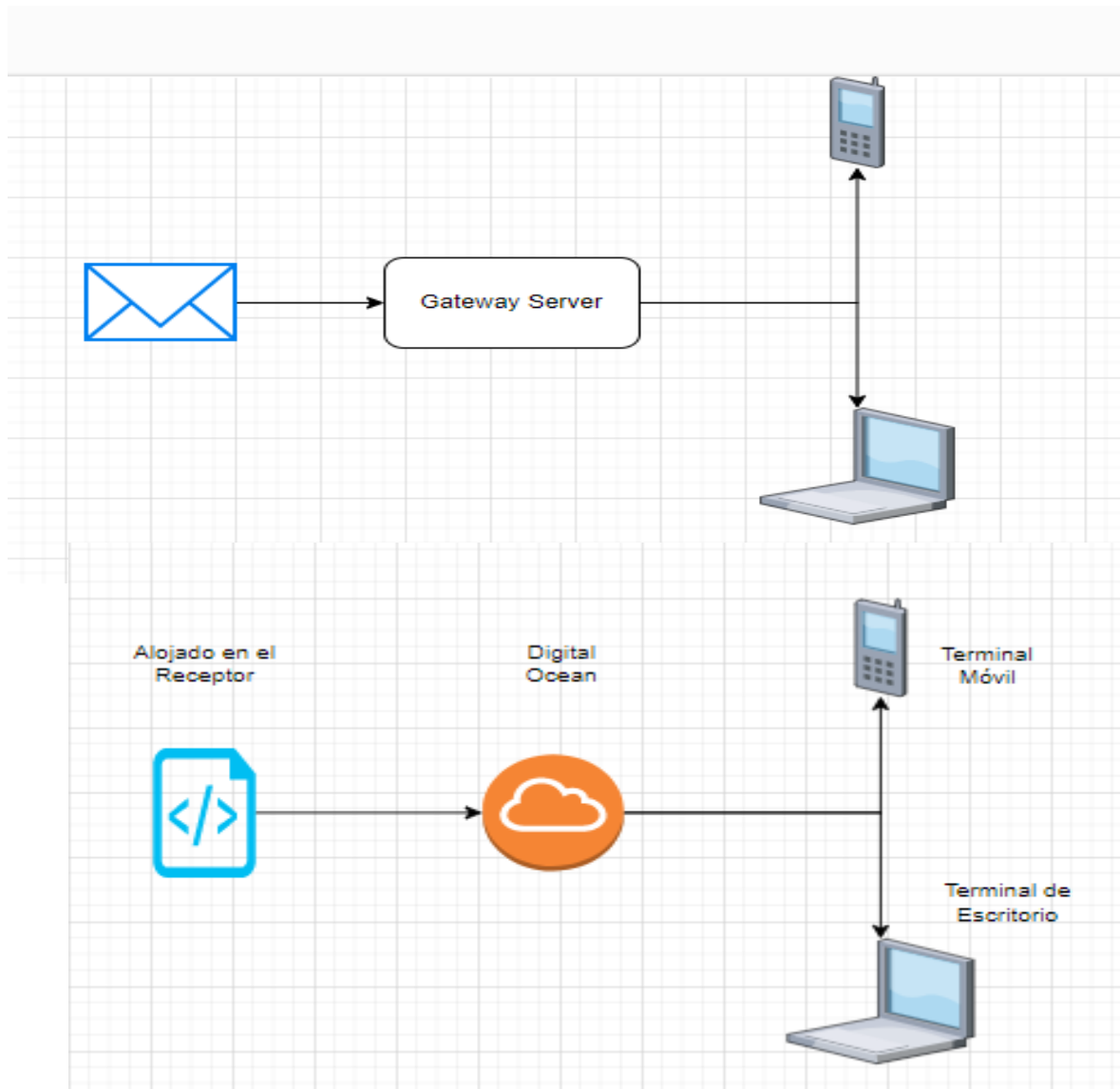
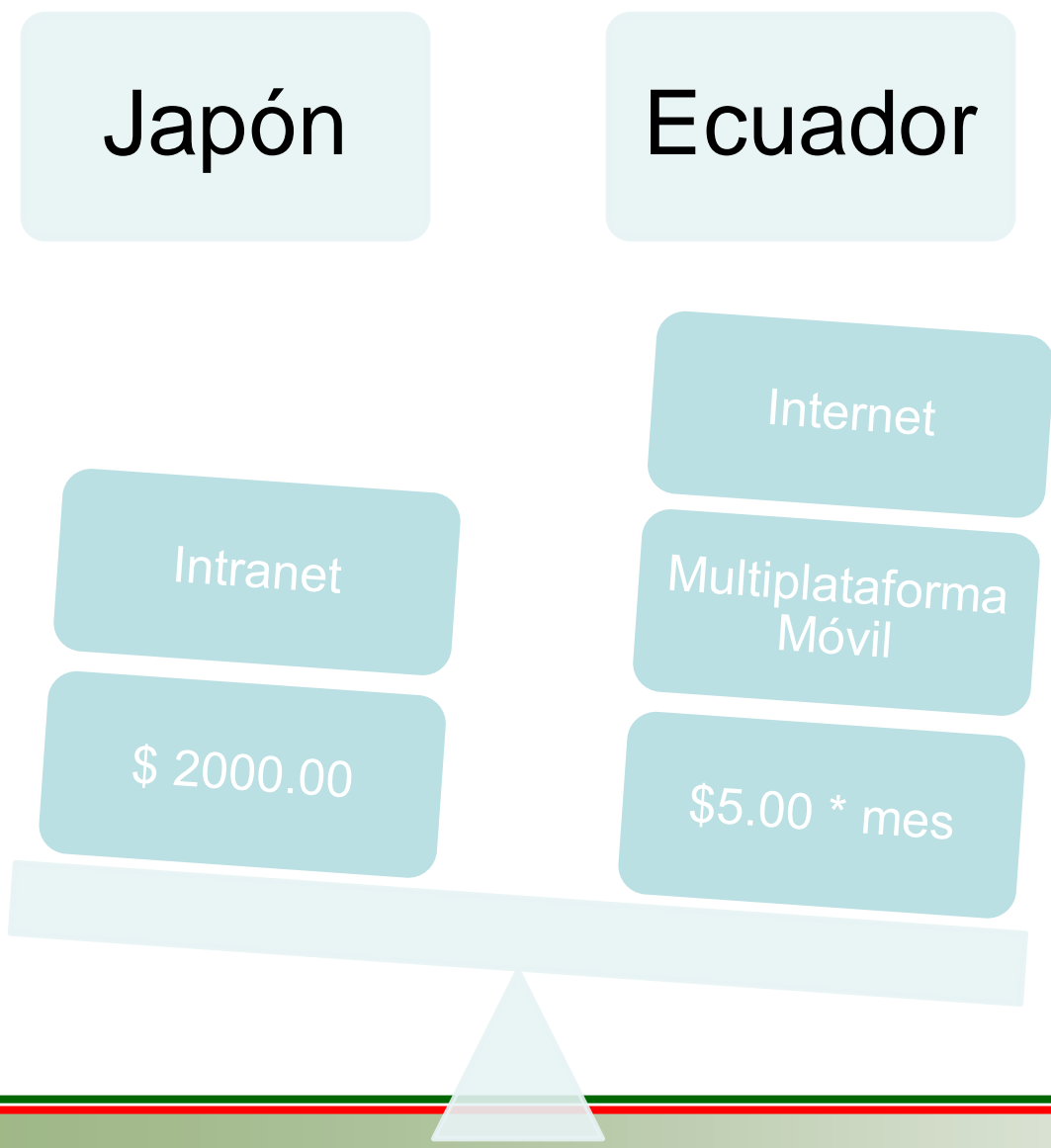
2023



CONTENIDO

- 1.- INTRODUCCIÓN
- 2.- OBJETIVOS
- 3.- MARCO TEÓRICO
- 4.-METODOLOGÍA
- 5.- PRUEBAS Y RESULTADOS
- 6.- CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS





Objetivo General

Desarrollar un Gateway Server para decodificar y retransmitir mensajes del sistema de alerta temprana en servicios de televisión digital terrestre.

Objetivo Específicos

Desarrollar el estado del arte sobre el protocolo del sistema de mensajes de alerta temprana y Gateway server.

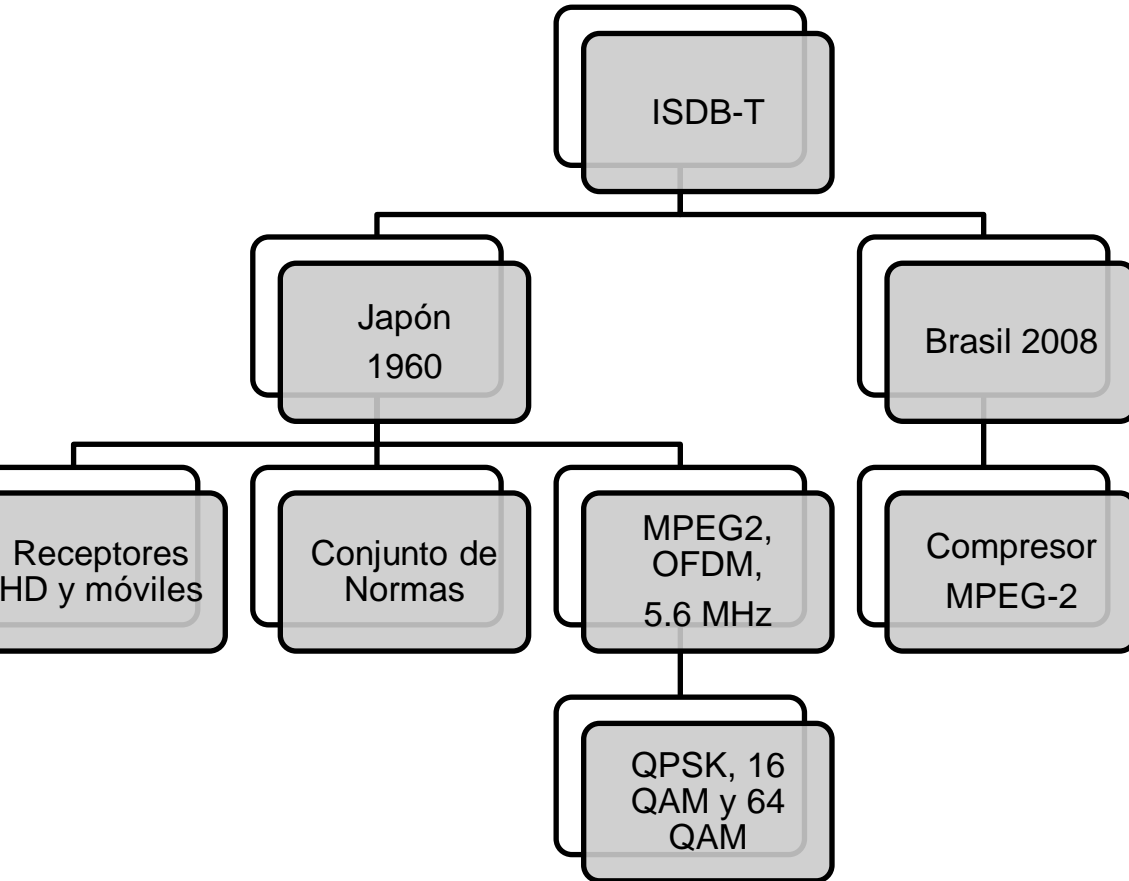
Desarrollar un Gateway server que sirva como medio de autenticación y almacene mensajes e identificadores de dispositivo (Device ID).

Desarrollar dos aplicaciones para clientes móviles y de escritorio para retransmitir el mensaje de alerta temprana EWBS.

Realizar pruebas de verificación del correcto funcionamiento del sistema.

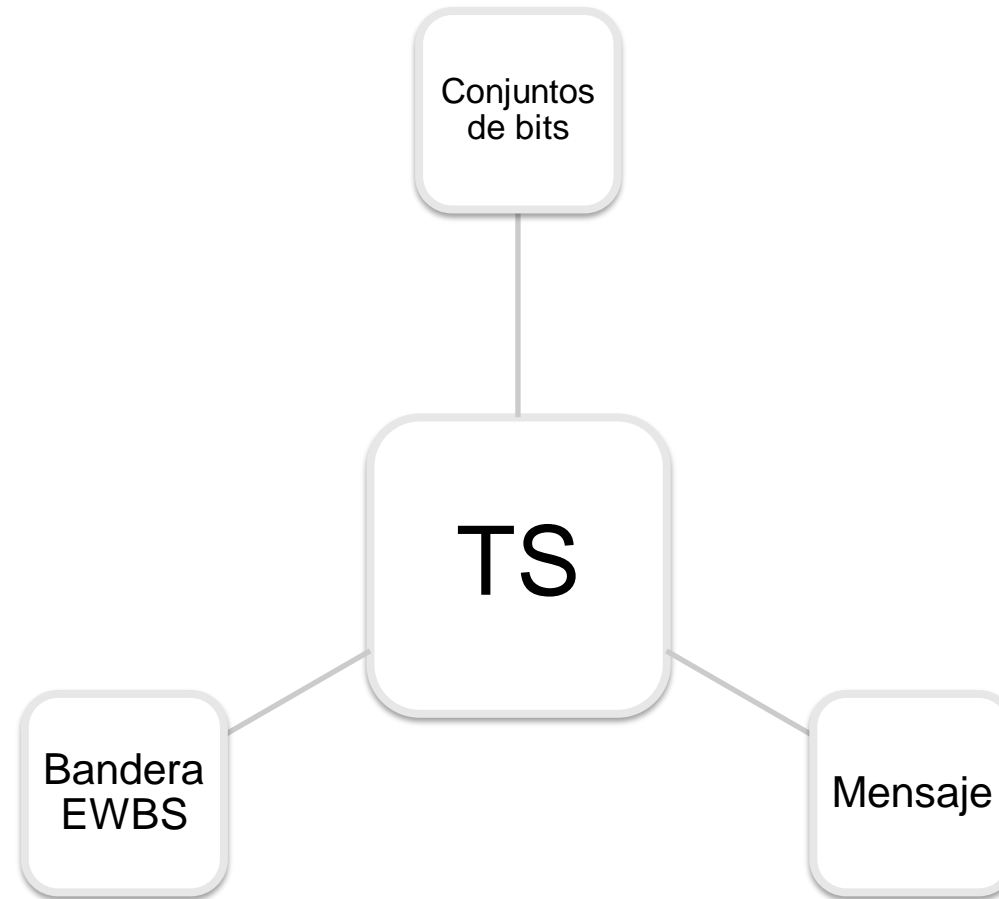


Estándar ISDB-T

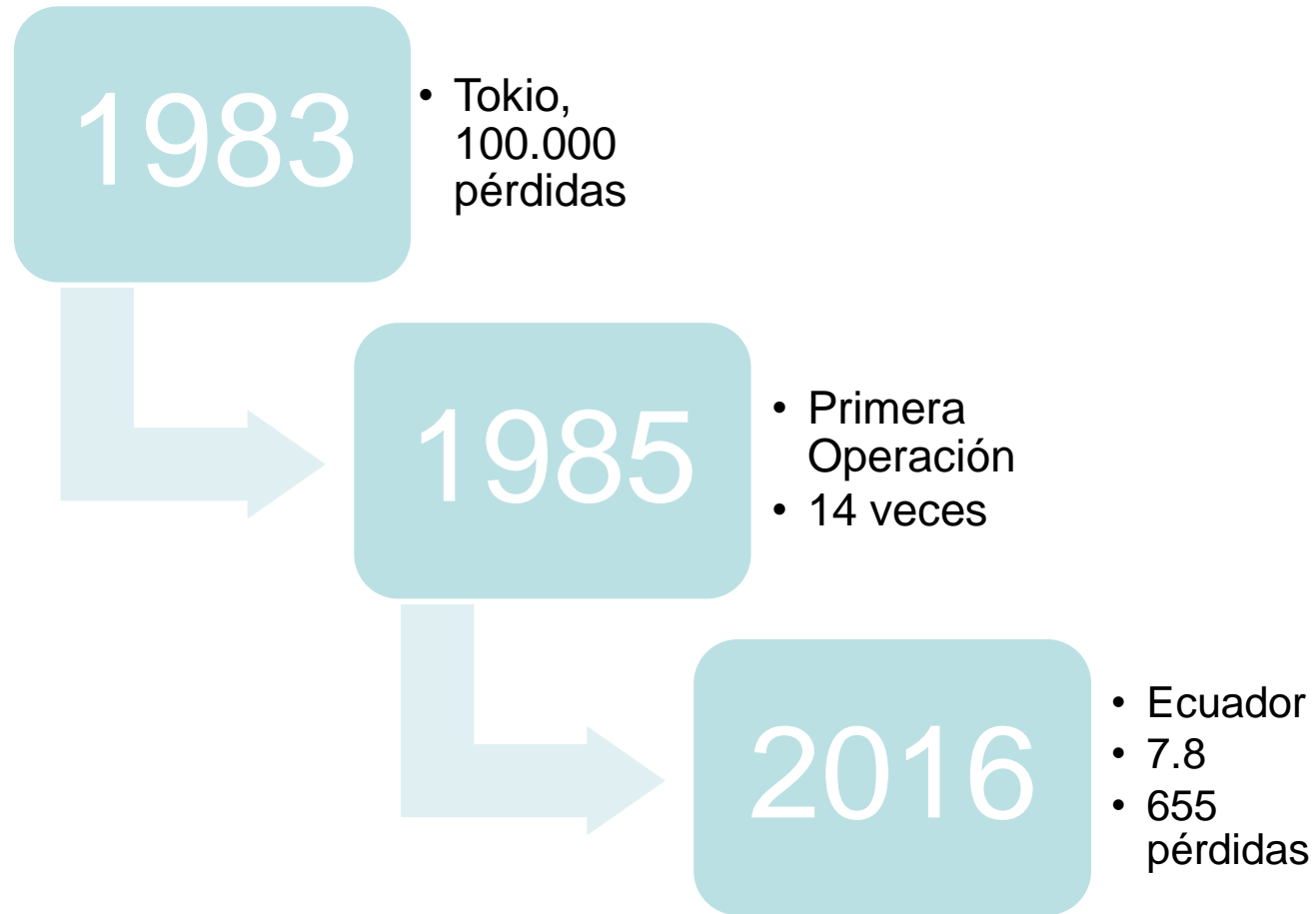


Nota. La figura muestra el diagrama de bloques de un Sistema de transmisión ISDB-Tb. Tomado de *Análisis del Transport Stream para el estándar de televisión digital ISDB-TB* (p. 29), por Granja y Olmedo, 2011.

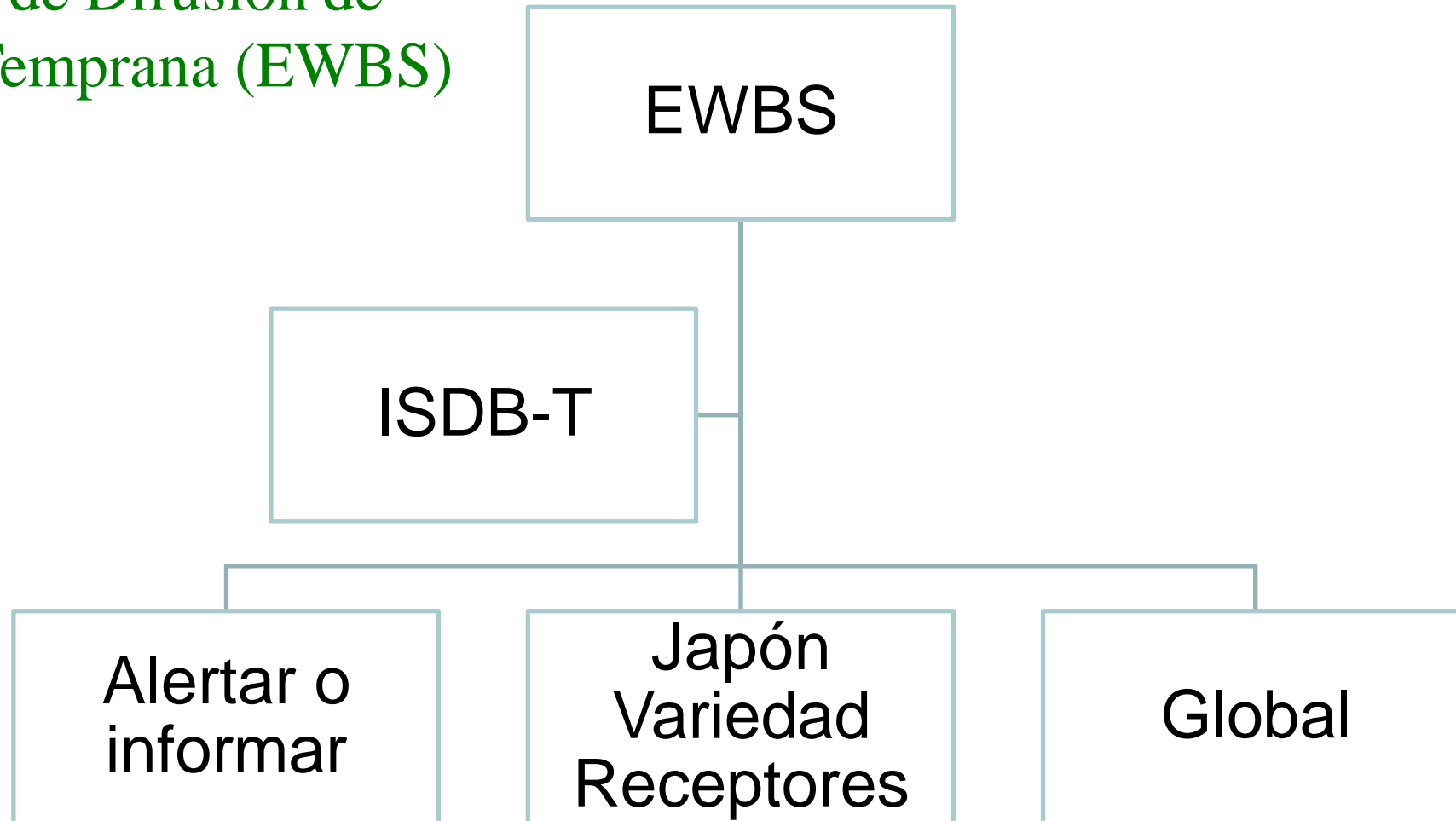
Flujo de Transporte TS (Transport Stream)



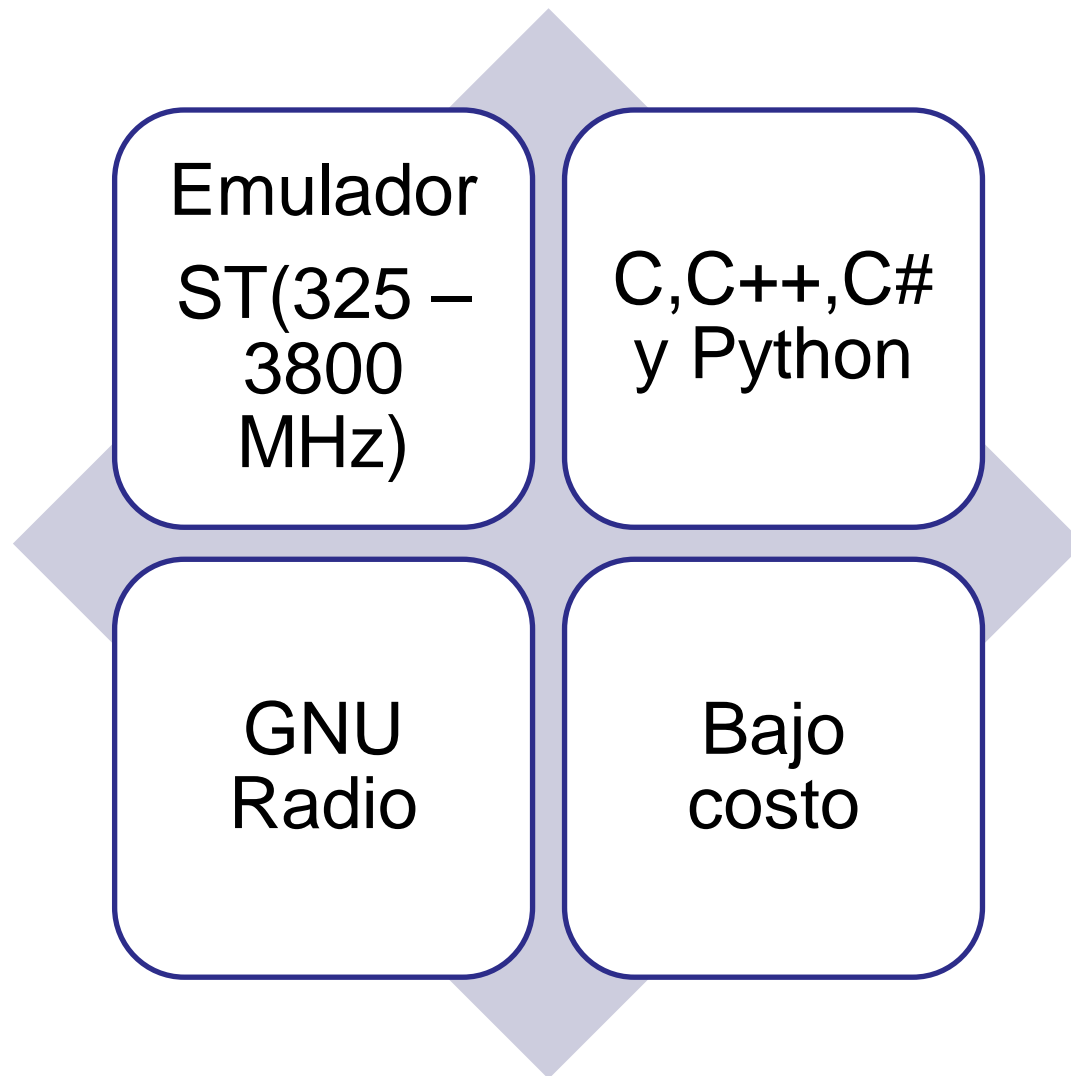
Sistema de Difusión de Alerta Temprana (EWBS)



Sistema de Difusión de Alerta Temprana (EWBS)

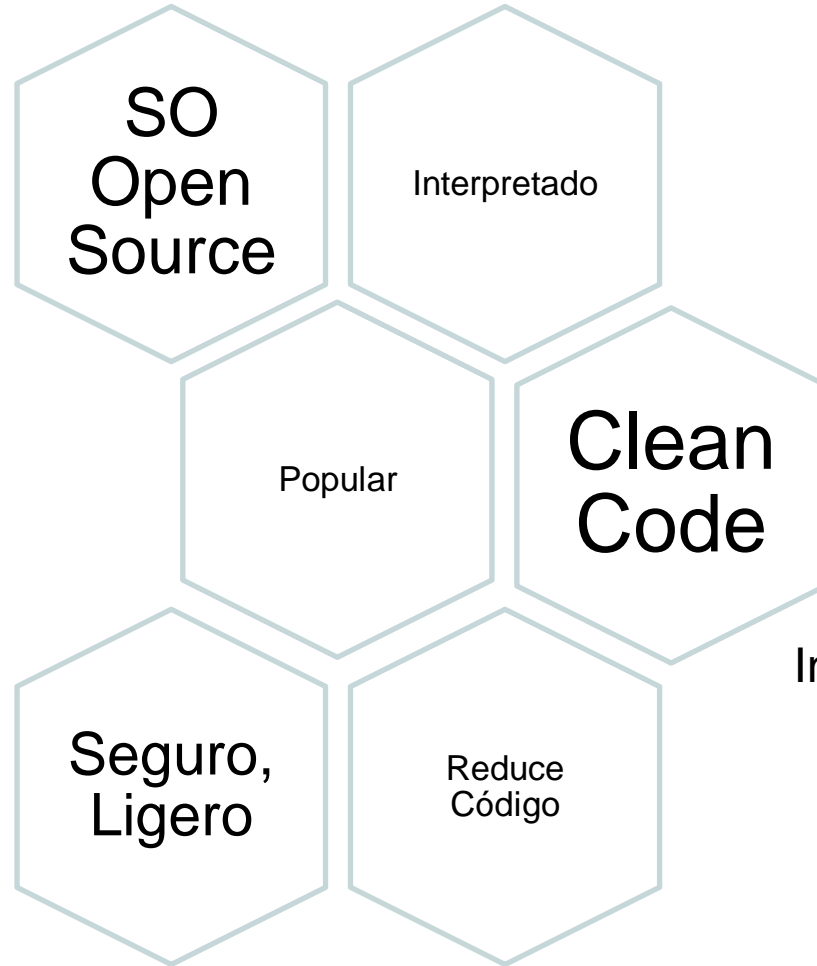


SDR

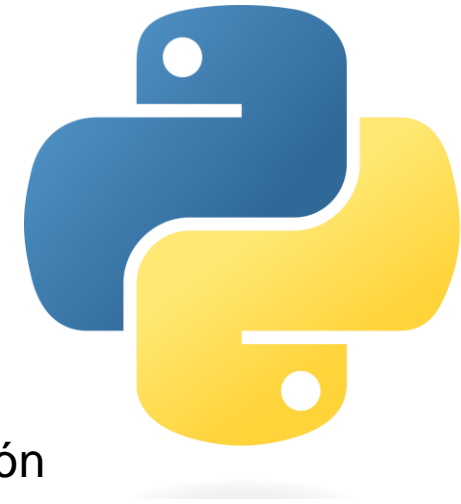


Nota. La figura muestra un SDR Adlam Pluto. Tomado de *Adalm – Pluto SDR Active Learning Module* (p. 2), por Analog Devices, 2017.

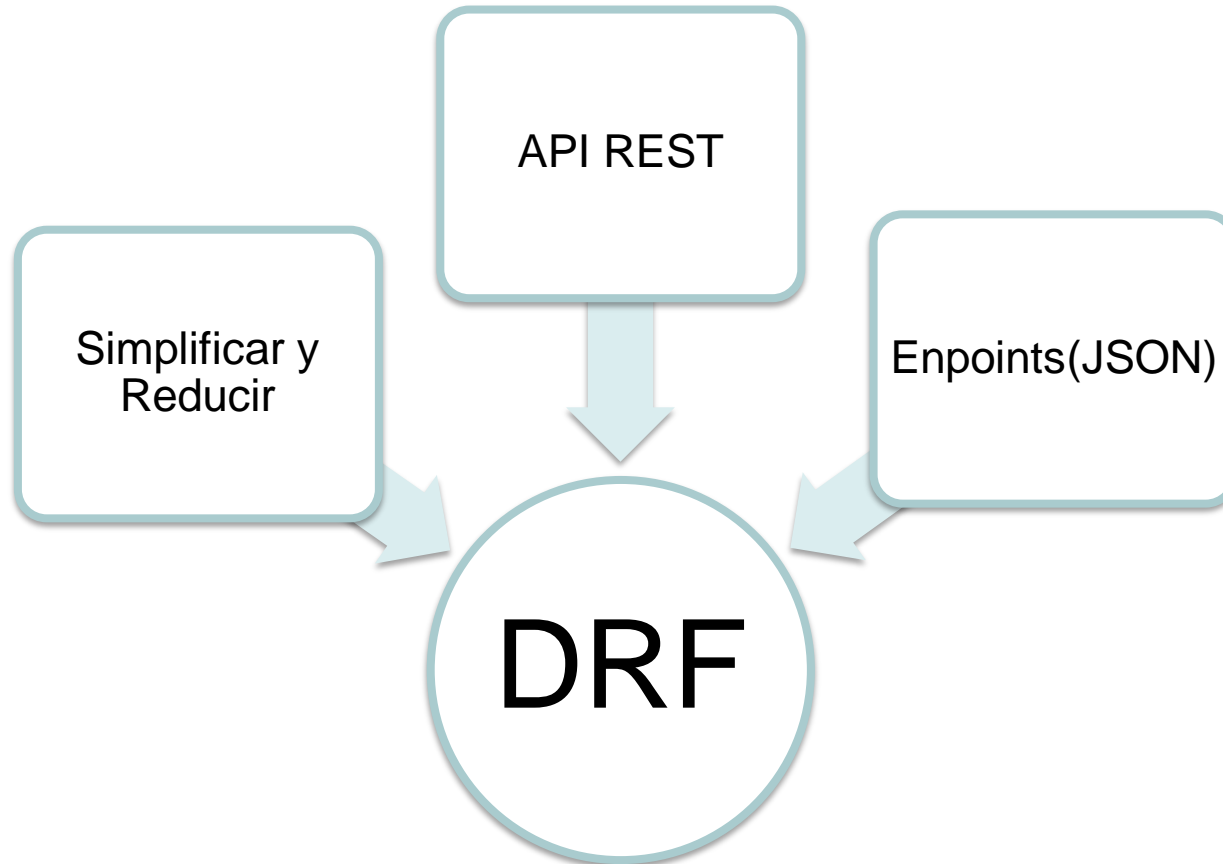
Ubuntu 20.04 – Python 3.8



Indentación

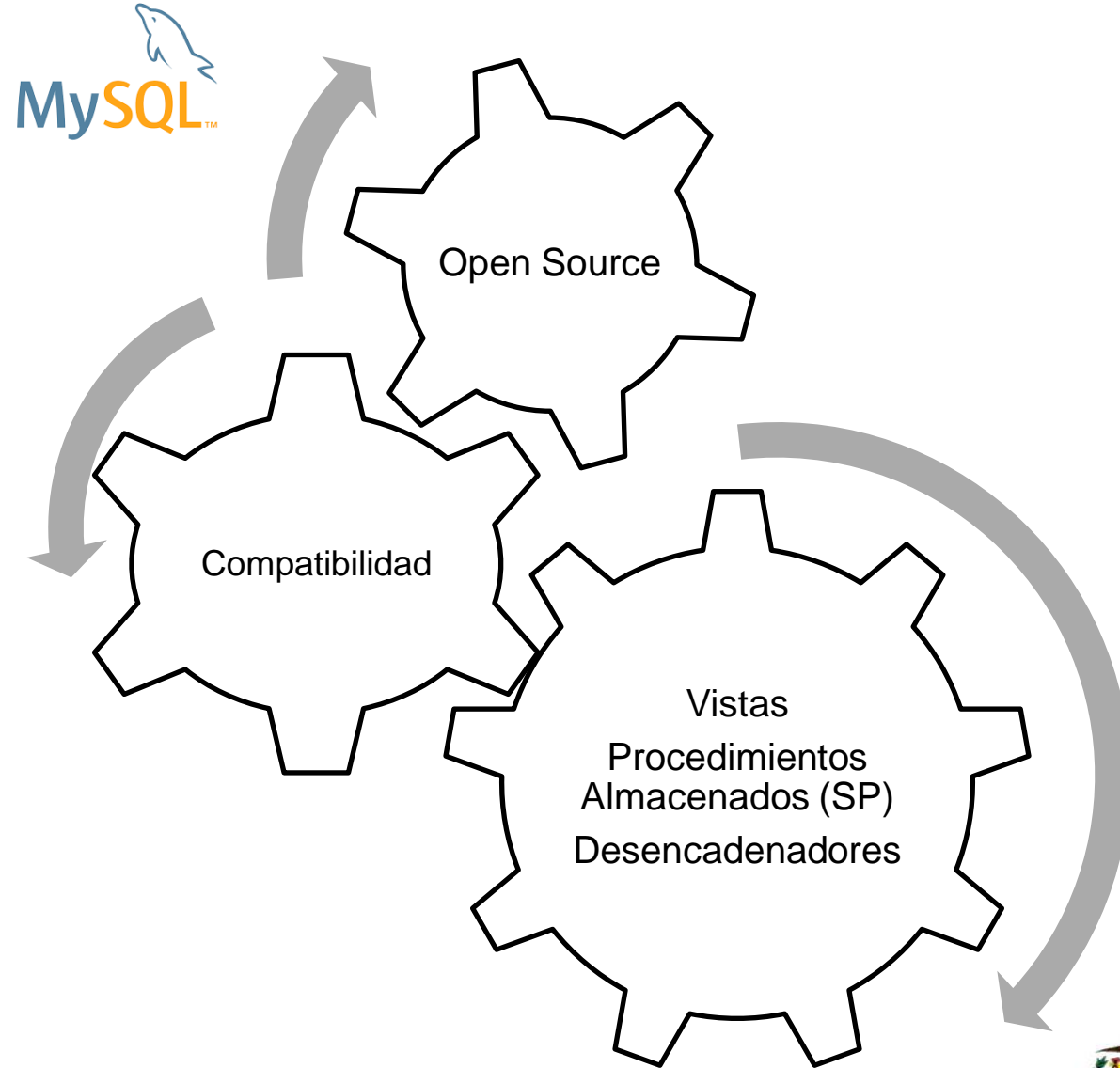


Django Rest Framework (DRF)

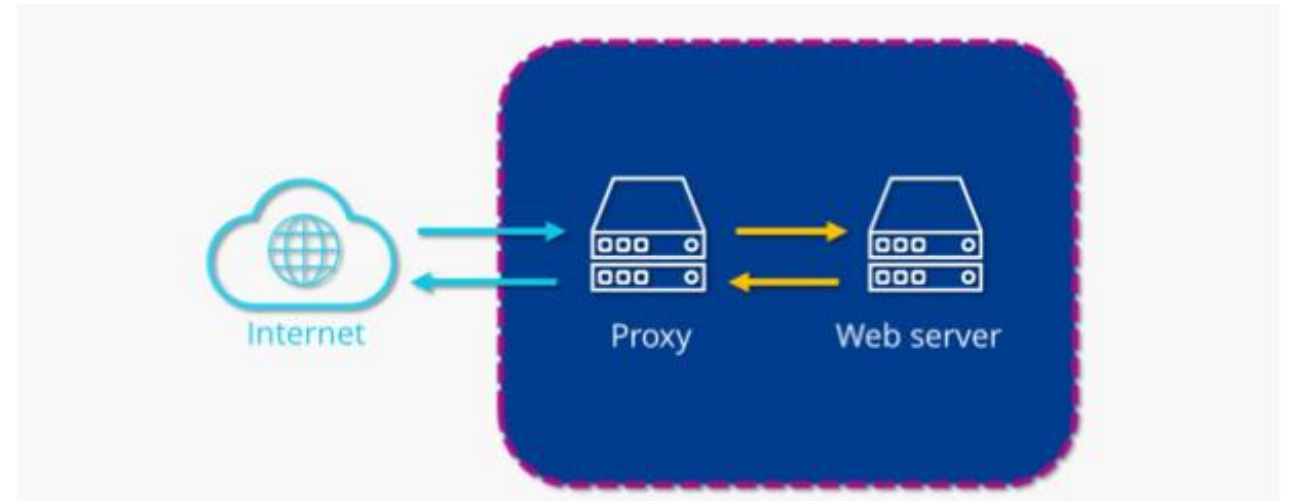
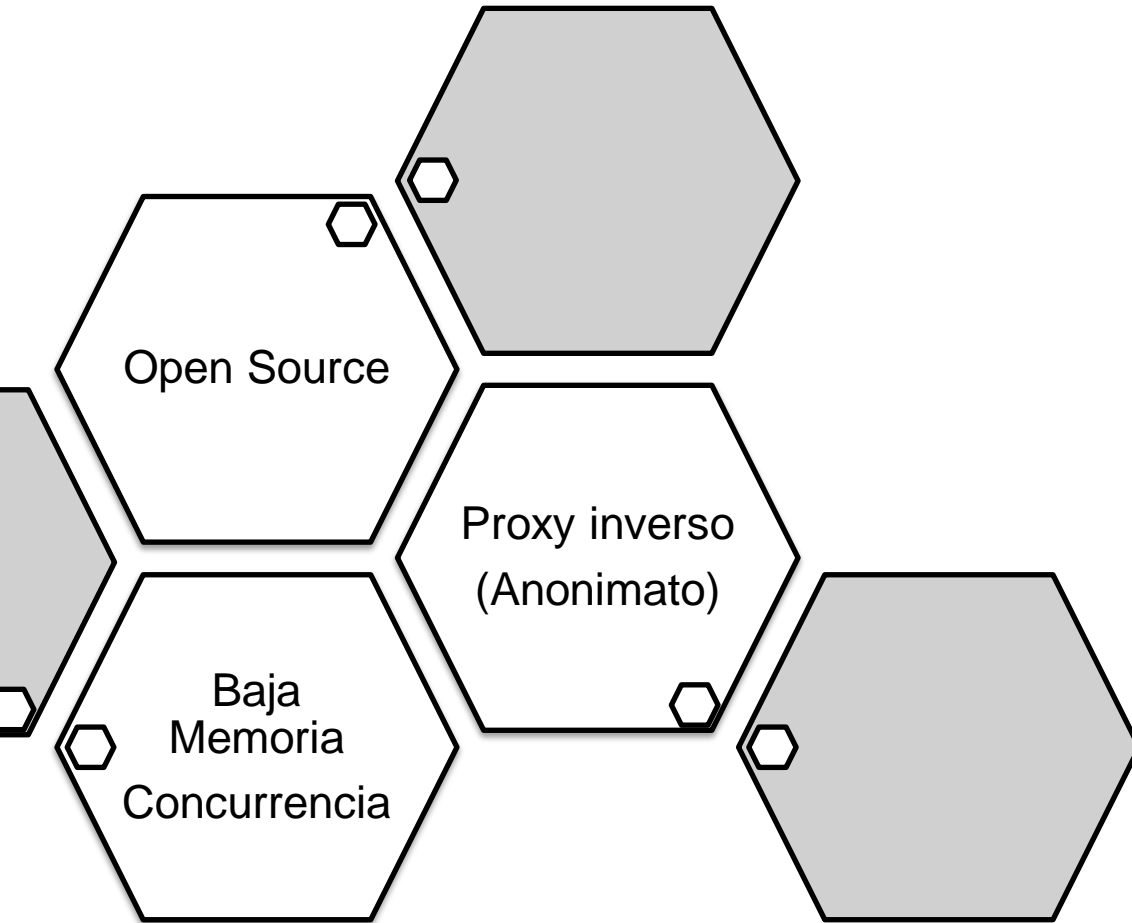


<https://alertaewbs.site/SaveTokens/>

MySQL Server



Nginx



(Alonso, 2019).

Java



3000 millones

Open
Source

Multi
plataforma
(Win –
Linux)

Robusto
Compilado
Tipado

Java

- `public int sumar(int a, int b)`

PHP

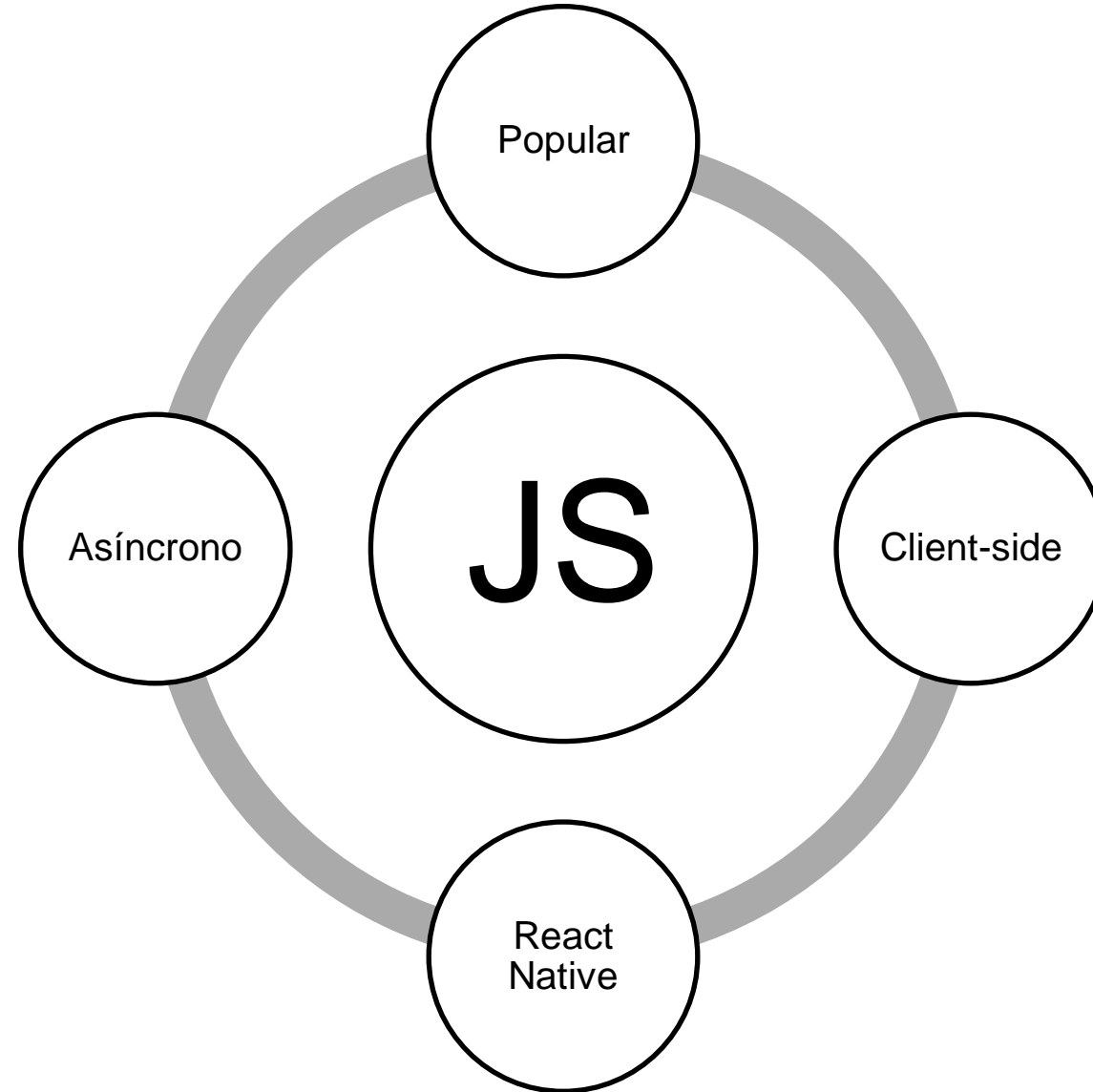
- `Function sumar (a, b)`



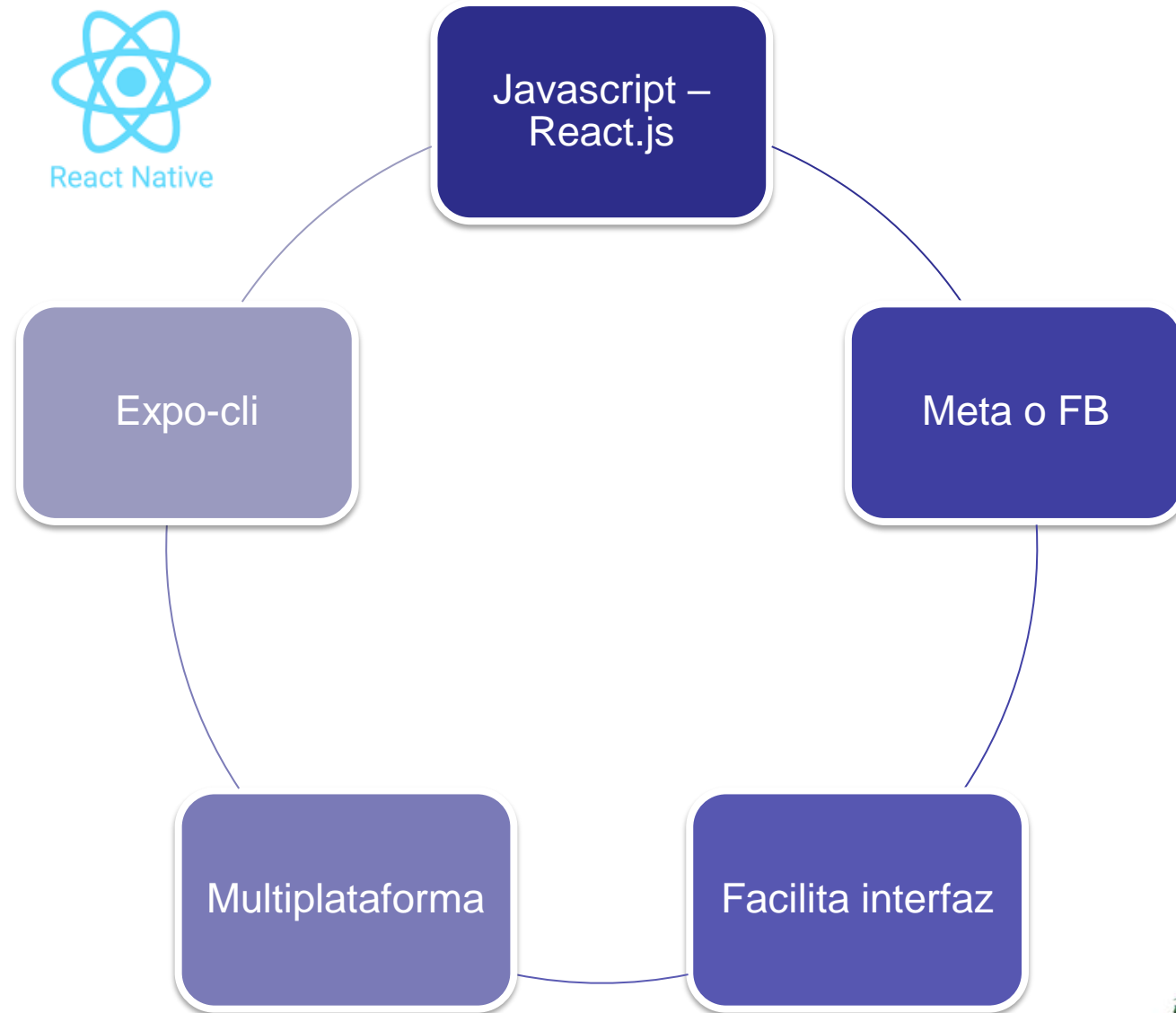
ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Javascript

JS



React Native





<https://alertaewbs.site>



Transporte



Cloud Computing Digital Ocean

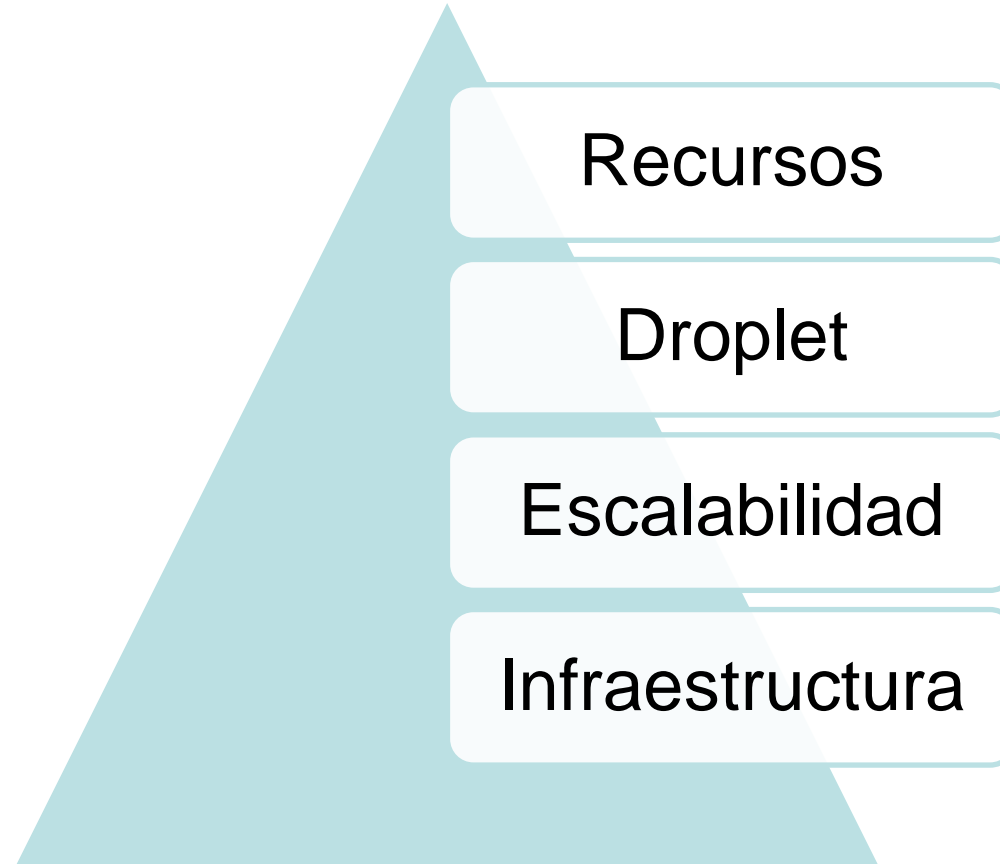
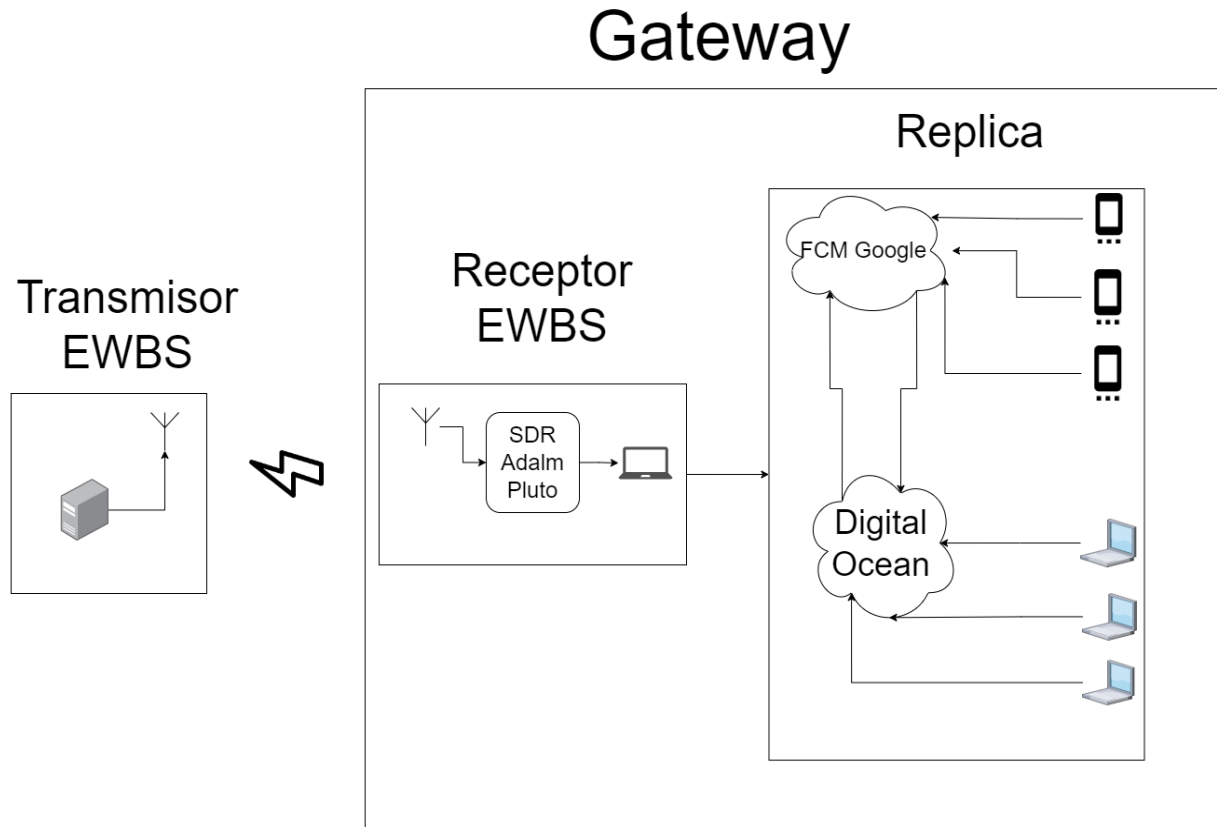


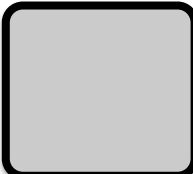


Diagrama de Bloques Completo



-  **Transmisor**
 - Ordenador (Stream Xpress)
 - Tarjeta Decktek (Modulador)
-  **Receptor**
 - Ordenador (Ubuntu – GNU Radio)
 - Adalm Pluto
-  **Replica**
 - Digital Ocean
 - Aplicaciones Multiplataforma

División de la Sección

Droplet

- MySQL
- Python
- Gunicorn
- Nginx
- DRF

Script Python

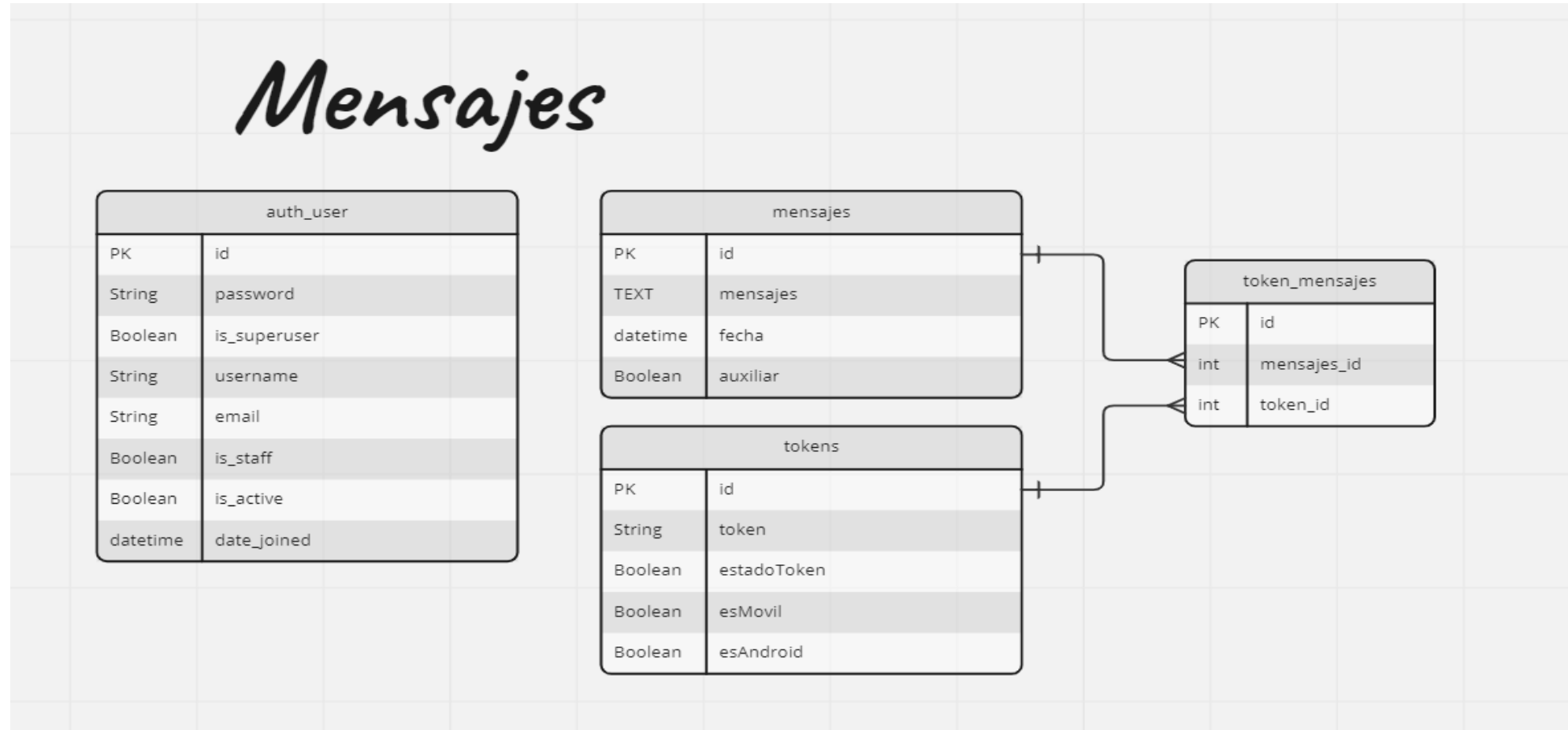
- Recupera el mensaje
- Guarda en el Droplet
- Difunde Mensaje

Aplicaciones

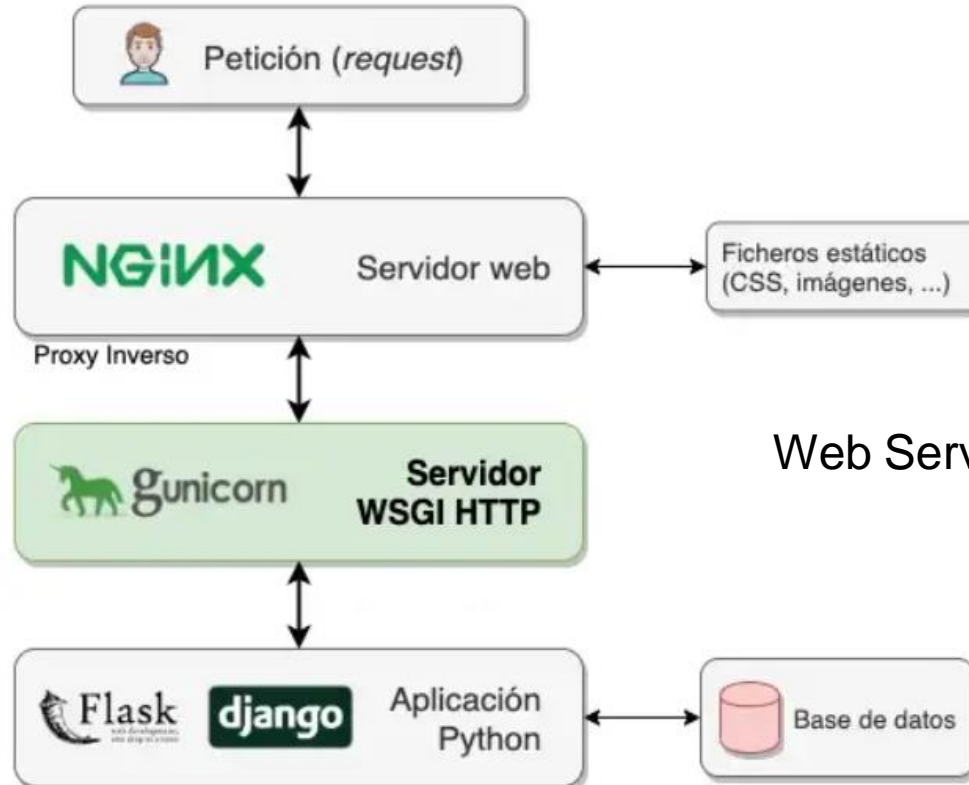
- Móvil
- Escritorio



Base de Datos (Mysql)



Gunicorn



Web Server Gateway Interface

Nota. La figura muestra el diagrama de bloques del funcionamiento de Nginx como servidor web de python con Django. Tomado de *¿Qué es un WSGI?*, por Ignacio Alonso, 2019.

DRF (Endpoints)

Ruta	Tipo	Acción
/Show/	GET	Lista mensajes.
/Insert/	POST(Autenticable)	Inserta mensajes.
/tokens/	POST	Guarda Tokens
/ShowToken/	POST(Autenticable)	Listar Tokens

Autenticación

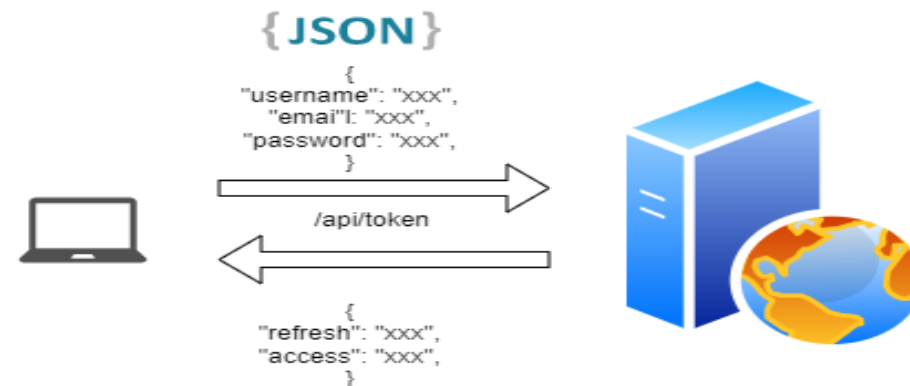
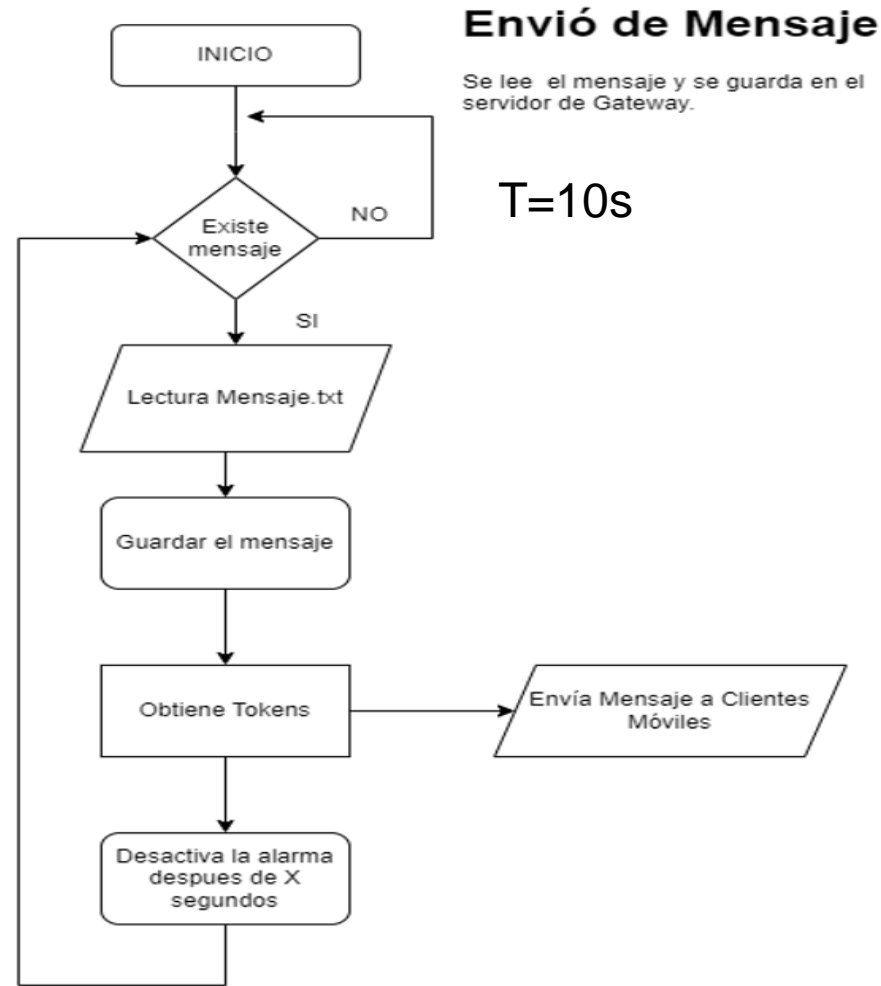


Diagrama de Bloques Script Python



Aplicaciones de Escritorio

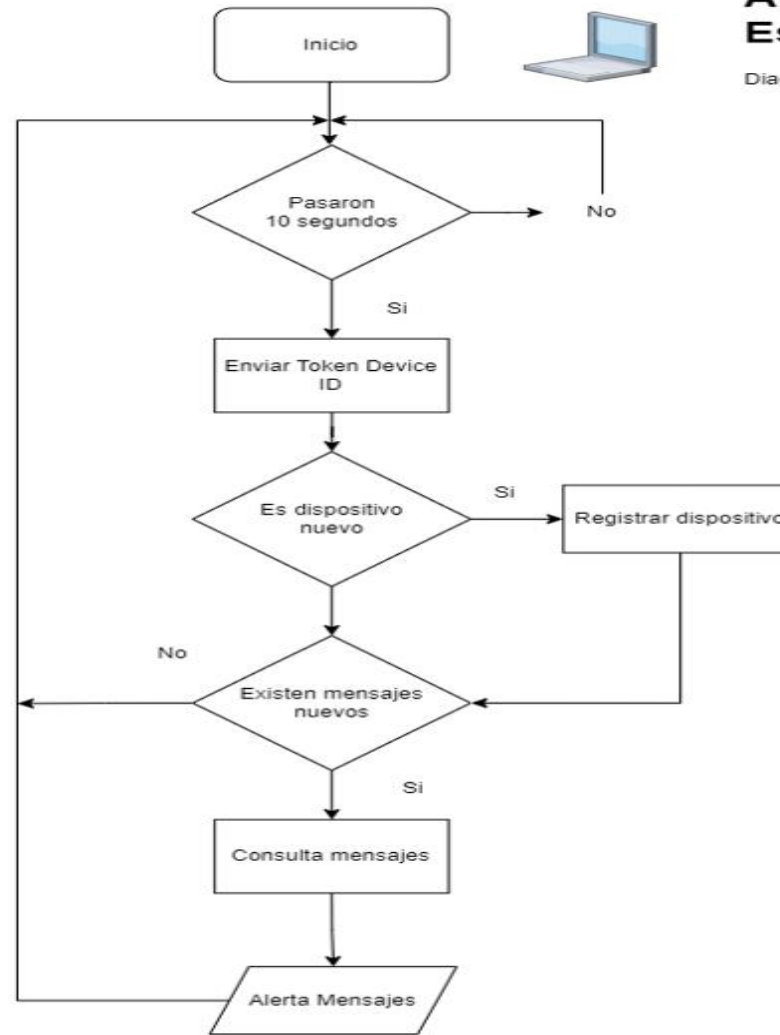


28080



Aplicación de Escritorio

Diagrama de Flujo



Aplicaciones Móviles

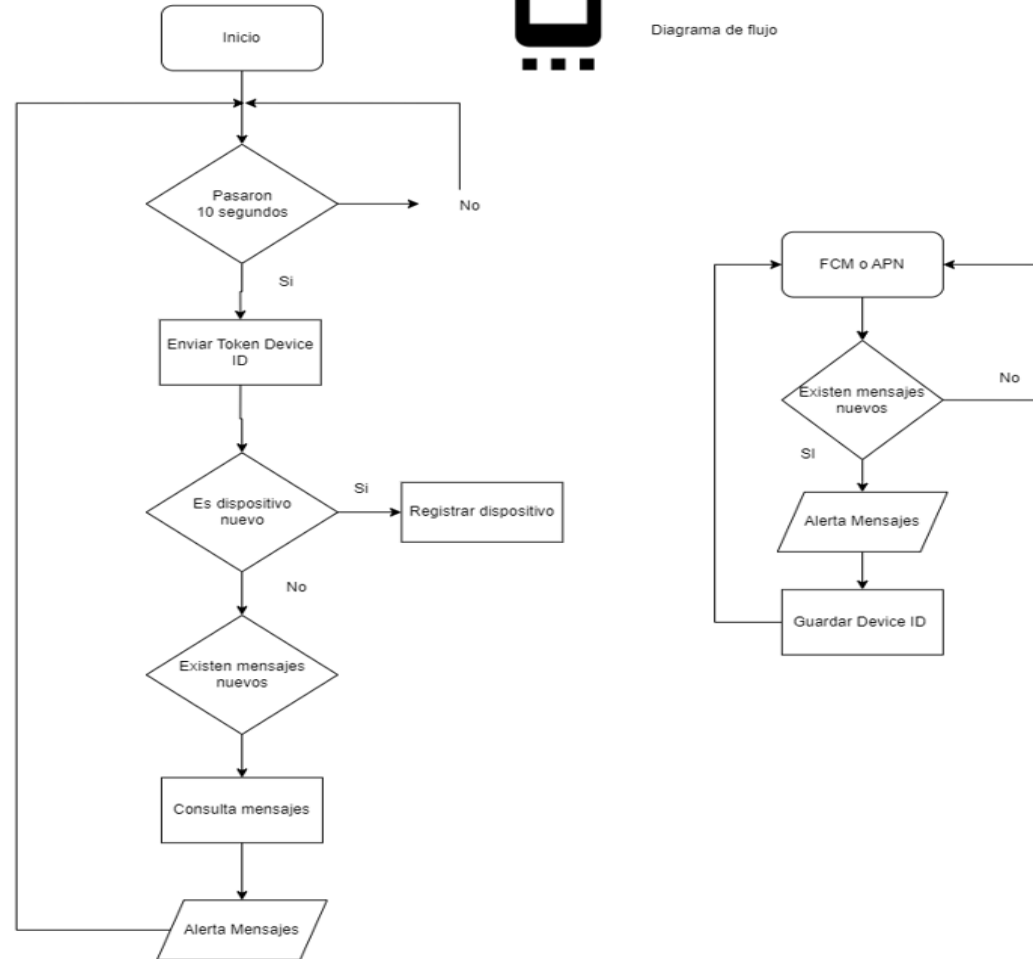
 Expo

 React Native

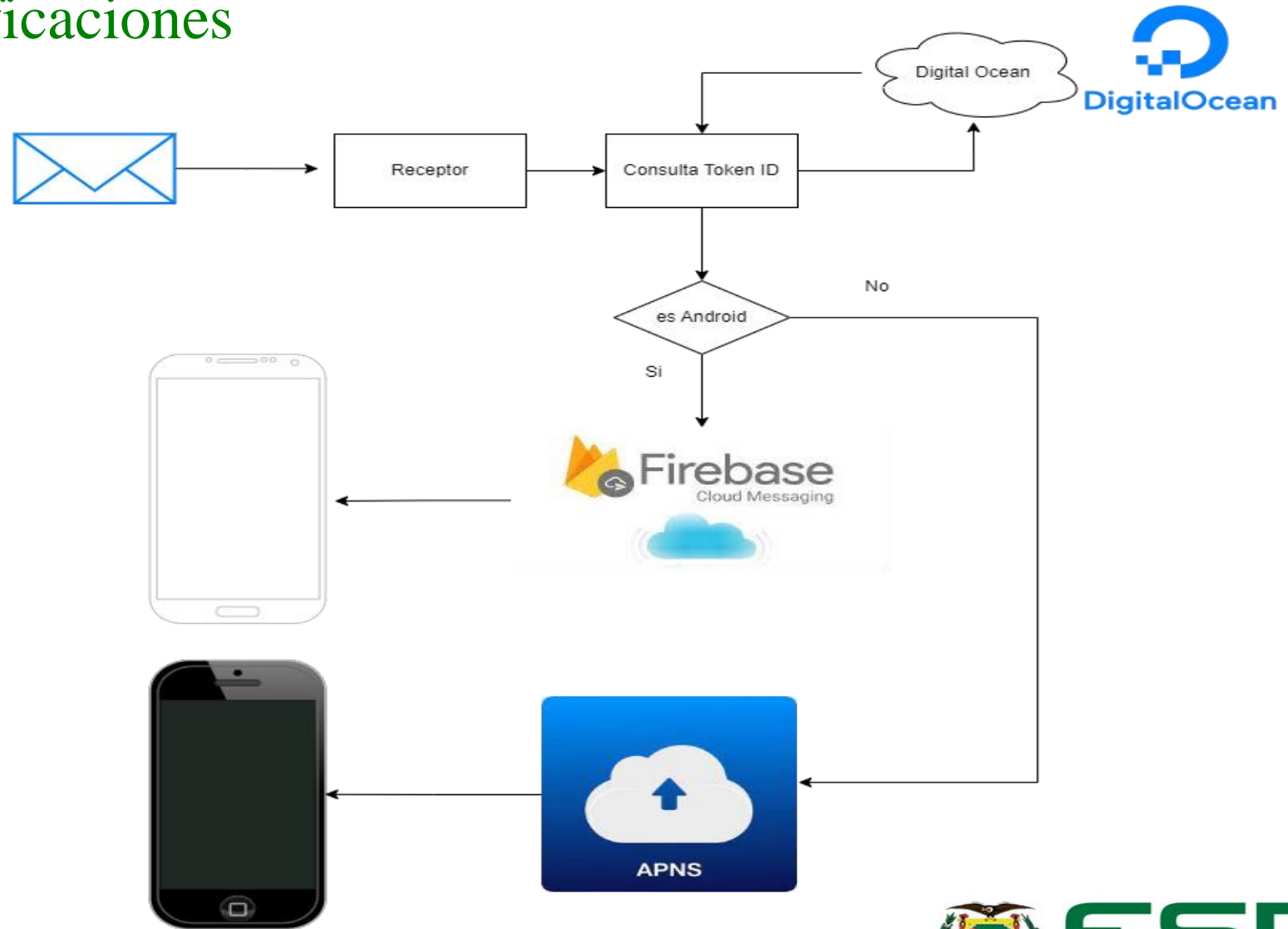


Aplicación Móvil

Diagrama de flujo

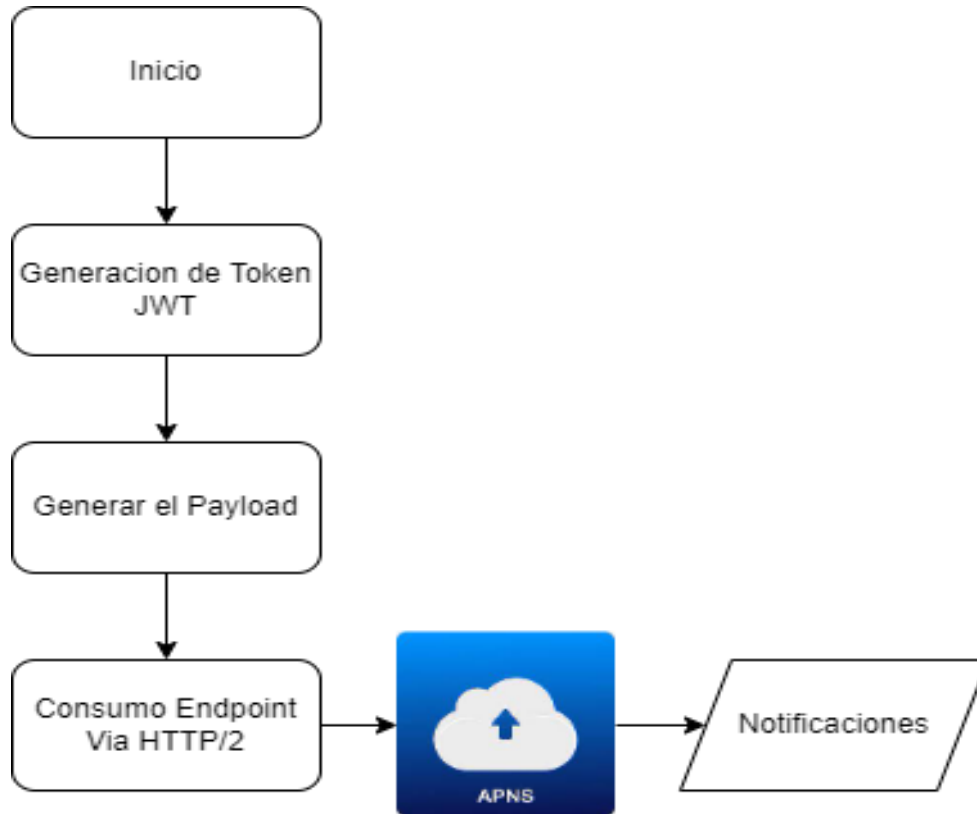


Funcionamiento Notificaciones



APNs

Parámetros



HYPER
Python 3.8

- APNS_KEY_ID: Generado desde la consola de Apple. <https://developer.apple.com/account/resources/> como se muestra en la Figura 26
- APNS_AUTH_KEY: Es un archivo de extensión p8 que se descarga desde la consola el panel de desarrollador de Apple. <https://developer.apple.com/account/resources/>
- TEAM_ID: Creado al pagar la suscripción.
- BUNDLE_ID: Creado al ejecutar el comando de empaquetado ***eas build -p ios --auto-submit***

Digital Ocean

Dominio	IP
alertaewbs.site	143.244.157.58

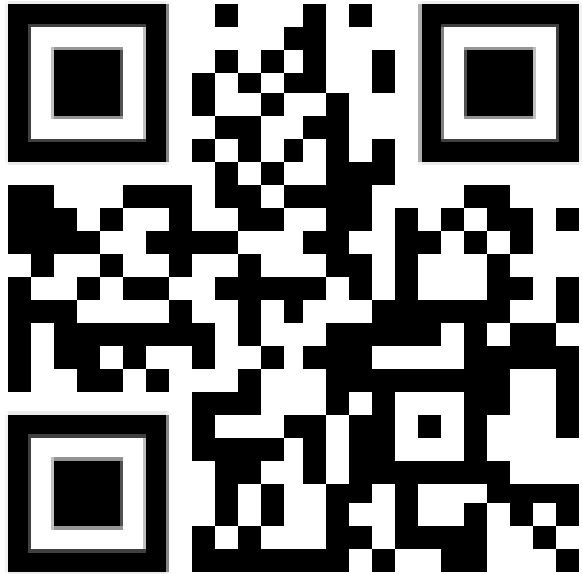
SO	vCore	RAM	SSD	Peticiones
	[Unidad]	[GB]	[GB]	[Unidad]
Ubuntu	1	2 GB	200	400
20.04 LTS				

Registros DNS

Los registros de DNS definen cómo se comporta tu dominio, como por ejemplo, mostrar el contenido de tu sitio web y entregar tu correo electrónico.

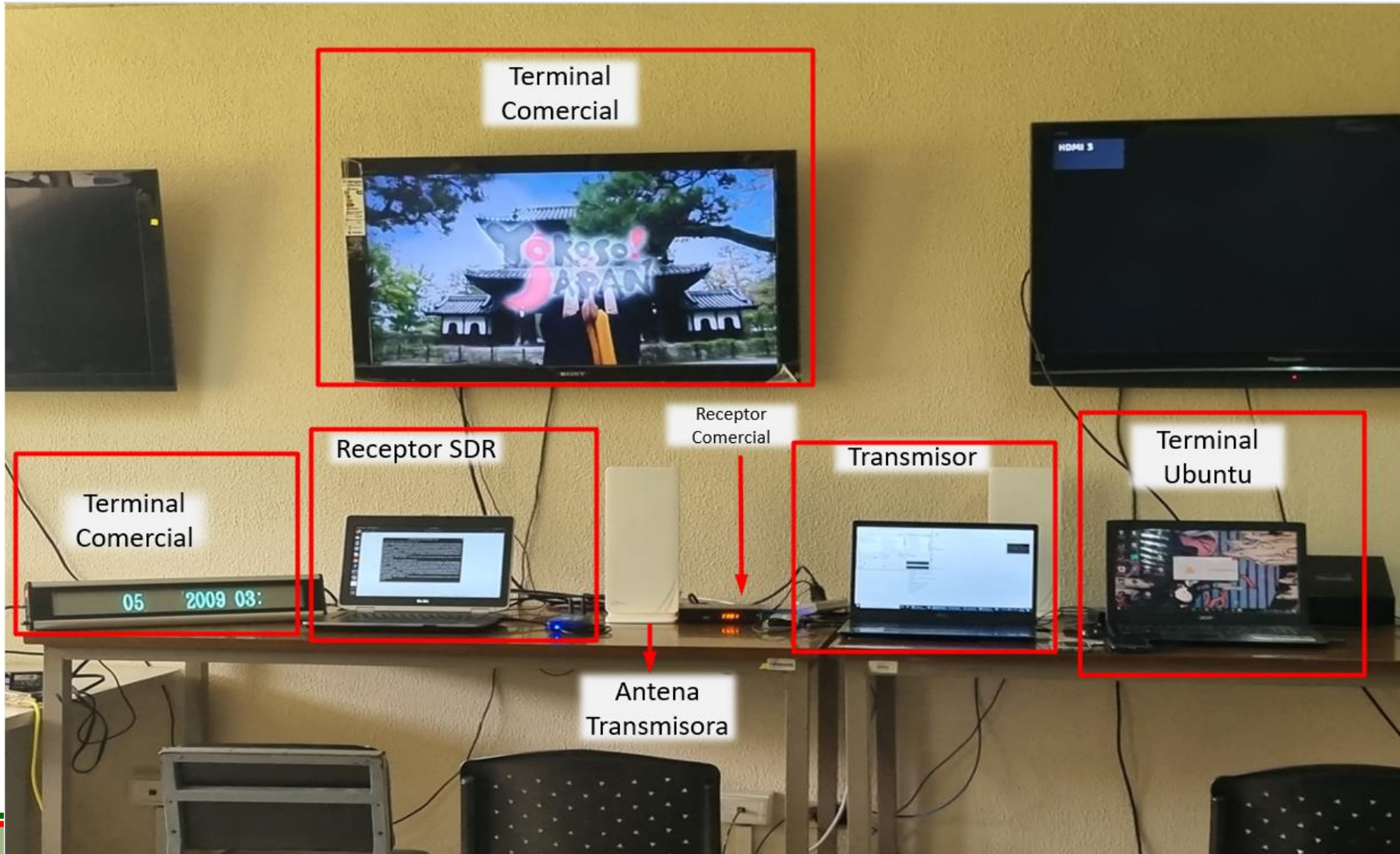
Eliminar		Copiar		Filtrar	Agregar	...
Tipo	Nombre	Datos	TTL	Eliminar	Editar	
<input type="checkbox"/>	A	@	143.244.157.58	600 segundos		

Digital Ocean (Puesta en Marcha)



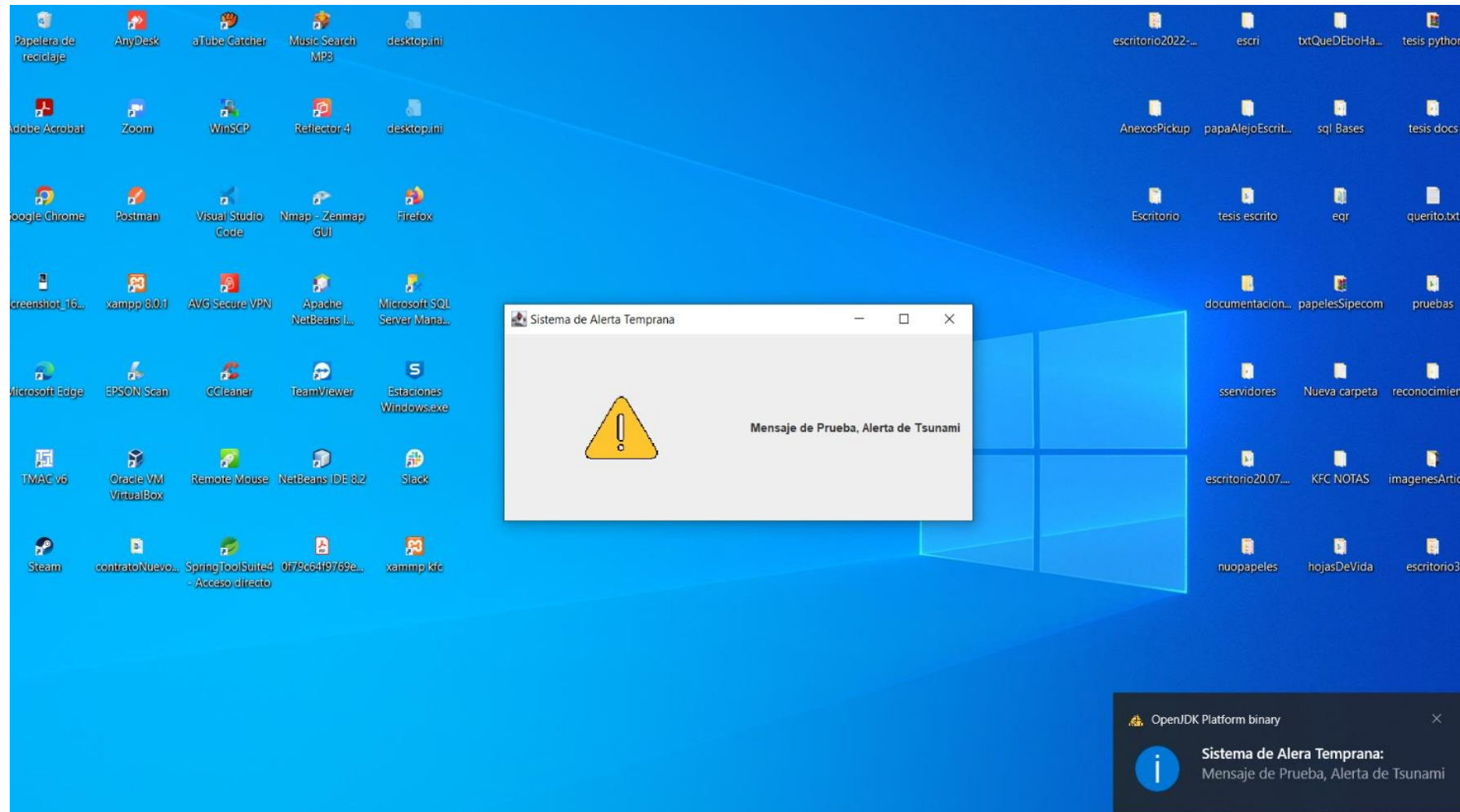
Herramienta	Versión
Python	3.8.10
Django RF	3.13.1
MySQL	8.0.31
Nginx	1.18.0
Gunicorn	20.1.0

Escenario Prueba



*Video demostrativo
ambiente real en el
Laboratorio de
Televisión Digital
UFA - ESPE*

Interfaz Escritorio



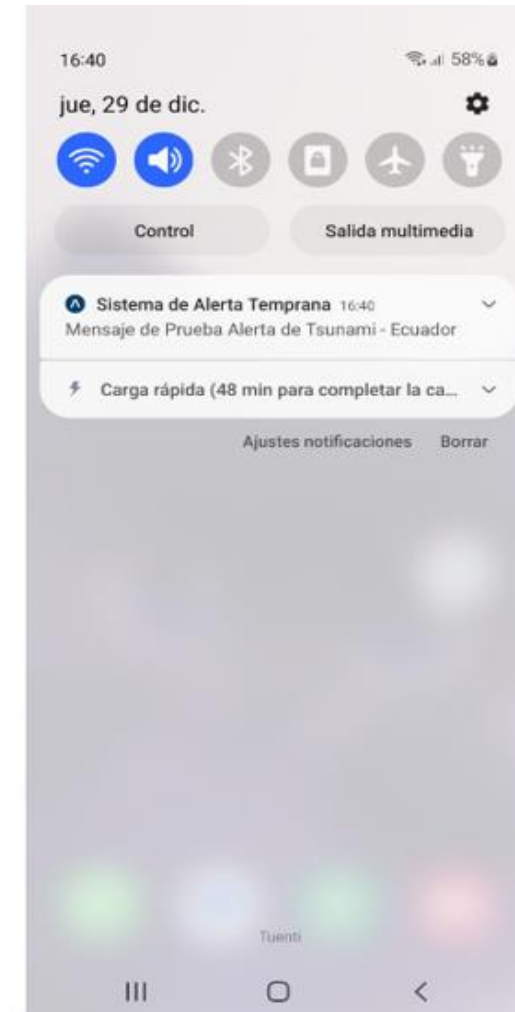
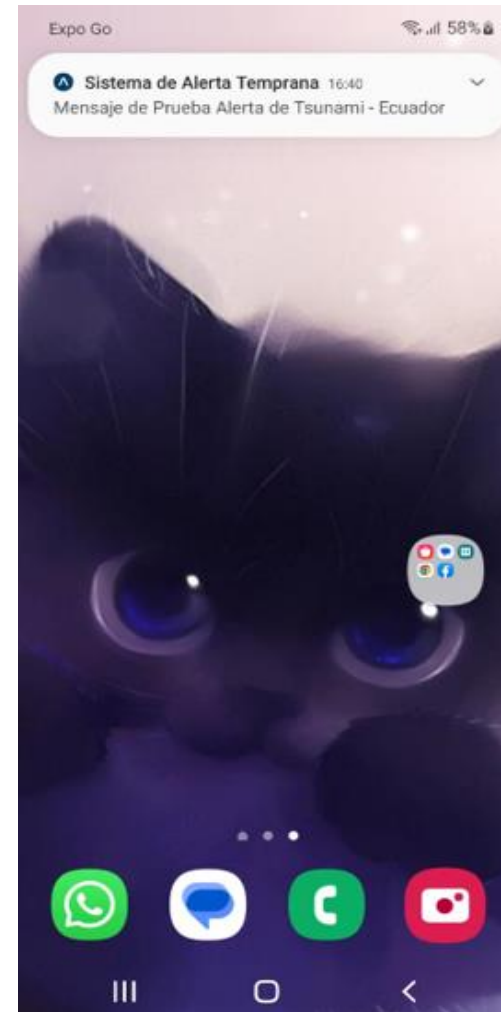
Video demostrativo ambiente Windows



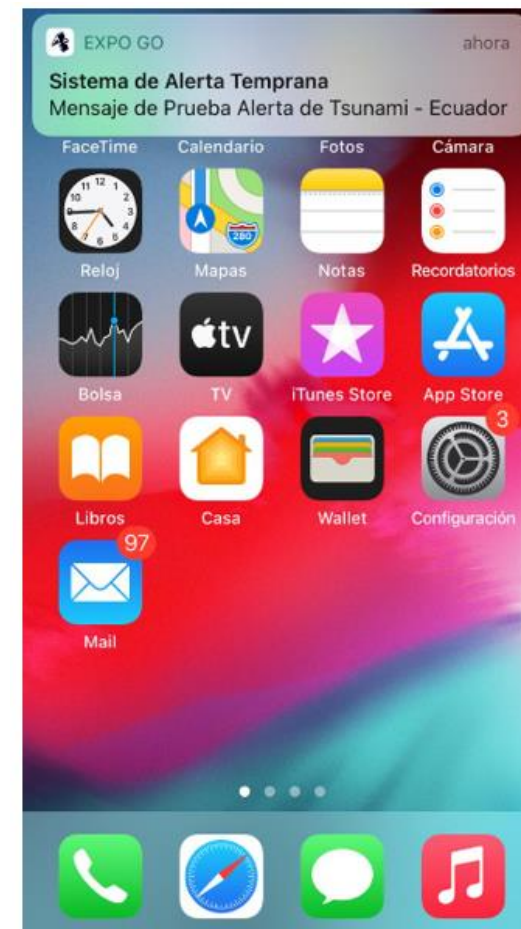
Video demostrativo ambiente Linux



Interfaz Móvil (Android)



Interfaz Móvil (iOS)



Videos Demostrativos



Video demostrativo ambiente móvil Android

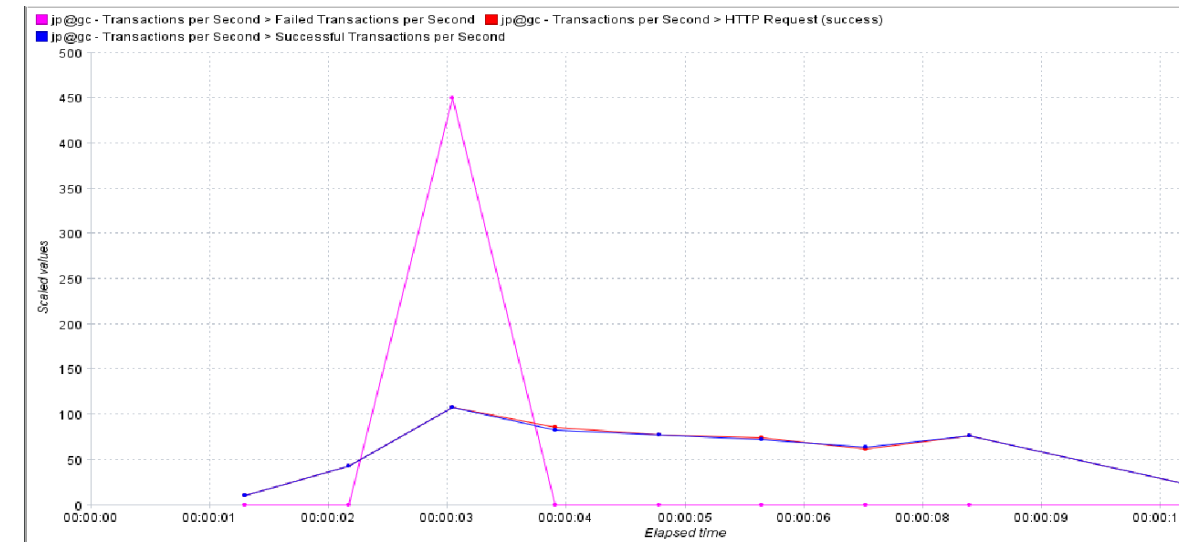
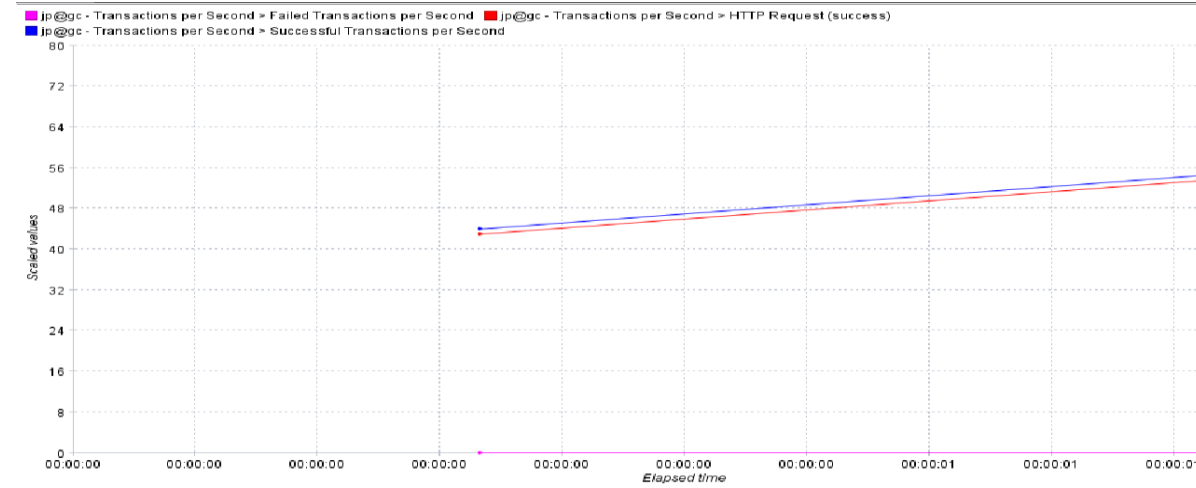


Video demostrativo ambiente iOS



Pruebas de Estrés

Número de peticiones	Porcentaje de error [%]	Tiempo [s]
100	0	1
200	0	3
300	0	4
400	0	5
500	3.2	7
600	10.67	9
700	35,86	8
800	44,75	8
900	46,44	7
1000	45,4	11



CONCLUSIONES

Se desarrolló el estado del arte sobre el funcionamiento un Gateway Server para poder retransmitir a dispositivos móviles y computadores el mensaje de alerta temprana.

Se desarrolló una aplicación de escritorio multiplataforma (Windows, Linux) en el lenguaje de programación Java y una aplicación móvil multiplataforma (iOS, Android) recibir los mensajes de alerta temprana, garantizando que se reciba el mensaje y el usuario se percate de inmediato de la alerta siempre y cuando el cliente esté conectado a Internet, si no lo está recibirá la notificación unos segundos después de tener la conexión.

Se utilizaron las librerías de desarrollo de JavaScript para interfaces React Native y *expo cli* debido la facilidad de crear con el mismo código aplicaciones para ambas plataformas o sistemas operativos como iOS (iPhone) y Android entre otros, además de la ventaja de no necesitar un ordenador MAC para codificar la aplicación en iOS.

El rendimiento y uso de recursos de una aplicación es menor cuando se codificada bajo su lenguaje nativo Android - Java o iOS – SWIFT, el peso de la aplicación disminuye considerablemente cuando es codificada en lenguajes nativos, sin embargo, mediante el uso de librerías que se compilan nativamente es posible desarrollar más rápida y eficientemente la interfaces y funcionalidad de una aplicación.

Se logró integrar el sistema de transmisión de televisión digital terrestre y difundir satisfactoriamente el mensaje de alerta temprana, mediante las pruebas de estrés se garantiza 500 usuarios de escritorio concurrentes con un servidor muy modesto y haciendo referencia al número de usuarios móviles no existe un límite de terminales en iOS y en Android se pueden enviar 1000 notificaciones por bloque de manera ilimitada.



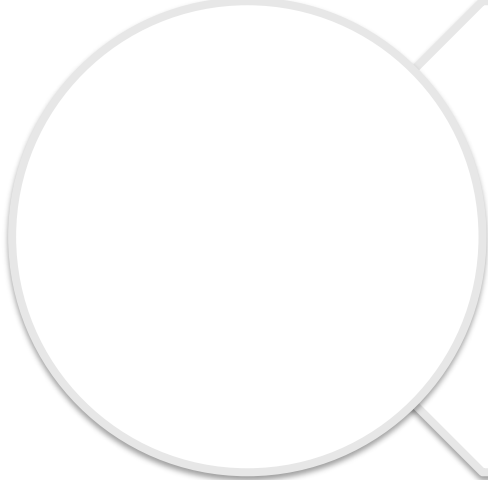
RECOMENDACIONES

Se recomienda el uso de React Native con expo cli y otras librerías y/o frameworks de programación si su aplicación no requiere demasiado configuración nativa o módulos nativos propios de iOS y Android, ya que aceleran mucho la programación de interfaces con muchos plugins o dependencias listas para el uso y por el gran soporte que tienen estas tecnologías por los usuarios.

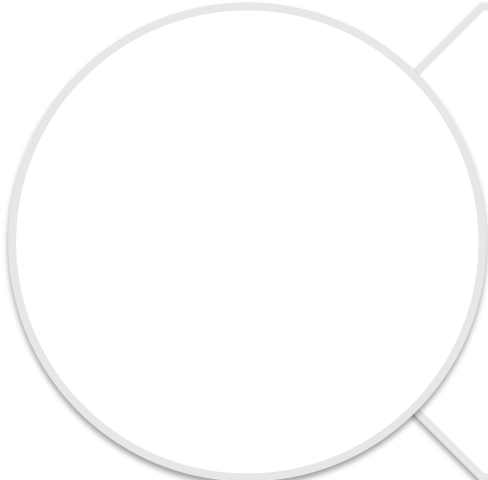
Se recomienda el uso de Cloud Computing debido al precio y al alcance global, que, de otra manera sería más costoso, además de ser amigable y tener documentación de cómo hacer los despliegues en la propia plataforma, facilitando así el poner en producción las aplicaciones Web.

Se recomienda seguir buenas prácticas, código limpio (en inglés, *Clean Code*), y utilizar un gestor de versiones para que el código sea reutilizable cualquier persona pueda entenderlo, modificarlo y añadir nuevas funciones, además de siempre asegurar la aplicación a nivel de protocolo con un certificado SSL y a nivel de autenticación con JWT o algún método de autenticación moderno que garantice la veracidad de los datos.





Para trabajos futuros se espera el uso de Docker y Kubernetes para montar sobre un contenedor el servicio de aplicación, aplicar un balanceador de carga, utilizar una base de datos para caché y orquestar la aplicación para soportar mayor cantidad de usuarios y optimizar recursos.



Se espera la obtención de la geolocalización del usuario para poder notificar solo a personas dentro del área de afectación del desastre natural, además de la integración con otros proyectos de grado para obtener rutas seguras de evacuación y una guía de qué hacer cuando sucede un determinado fenómeno.



MUCHAS GRACIAS



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA