

PROPUESTA IDEA TFG

Alexandru Cristian Stinga Micu – 2Dam

Mi propuesta se basa en la creación de un sistema de rastreo de aviones mediante una API que ofrece un conglomerado de datos sobre los aviones que se identifiquen en un en el espacio aéreo de una dimensión que nosotros definamos. Podemos ser capaces de filtrar por modelo de avión, compañía que lo opera incluso si es de uso militar.

Gracias al conglomerado de datos vamos a ser capaces de ver, posición (coordenadas), velocidad, altitud, origen y destino, código identificador y Callsign, si está o no volando y el tipo de avión (Si es pesado o no y cuanto pesa, si es pequeño (Avioneta XS), mediano (Jets S), grande (Avion comercial, cazas M), muy grande (Aviones comerciales gigantes, cargueros etc. L) y Gigantes (mega cargueros o aviones apaga fuegos XL).

1. ¿Qué problema resuelve?:

1. Logística Aeroportuaria

- **El problema:** En un aeropuerto pequeño o en la zona logística de los aeropuertos grandes, los operarios suelen estar esperando de brazos cruzados esperando una confirmación de que en 10 o 15 o el tiempo que sea va a aterrizar un avión carguero de no sé dónde. Cuando les avisan deben acordarse o apuntarlo en su lista para empezar a actuar.
- **Solución:** Con mi solución “Chivato automático”, metes el código del avión que estas esperando y cuando cruza una línea imaginaria (El principio de nuestro espacio aéreo) la app pita y les avisa cuando se detecta al avión que están esperando (tamaño, peso, cuanto le queda para aterrizar) y esos datos se les guardan así pueden tomar decisiones de si coger 2 o 3 camiones de carga de paquetes, evaluar cuando salir a atender el avión etc. así hasta que les avisan pueden estar dedicando su tiempo a otro trabajo.

la app pita. Así los carretilleros, los del combustible y los de la furgo se ponen en marcha justo cuando hace falta. Ni antes, ni después.

2. El "mono" de los aficionados (Spotters)

- **El problema:** Hay muchos aficionados a la fotografía aeronáutica “Plane spotting”. El que quiere hacerle una foto a un avión militar o a un modelo raro (como un Beluga o un caza) no puede estar pegado a la valla del aeropuerto 24 horas.

- **Solución:** Un sistema de alertas personalizadas. El usuario dice: "Avisame si aparece un F-18, o tal modelo de avión o lo que sea". Mi app escanea escanea el cielo y, si lo encuentra, lanza un aviso al móvil o al PC. Es como un Tinder de aviones: solo te avisa cuando hay un "match".

3. El problema de la seguridad (Análisis de rutas)

- **El problema:** No sabemos si un avión está haciendo algo raro o si siempre pasa por el mismo sitio. Las webs gratis no nos dejan ver qué hizo un avión hace tres días a menos que pagues.
- **Tu solución:** Como guardamos en mi propia BBDD (PostgreSQL), puedo ver el historial. Y generar estadísticas y decir: "Oye, este avión militar ha pasado por encima de esta base cinco veces esta semana a la misma hora". Eso sirve para pillar patrones o ver si alguien está "husmeando" donde no debe.

2.TECNOLOGIAS:

Lenguajes de programación:

JAVA & KOTLIN: Estructura y lógica moderna y limpia.

PYTHON : Ingesta y Procesamiento de Datos

KOTLIN

Sistemas de gestión de BBDD:

Motor: PostgreSQL

Extensión: PostGIS. Imprescindible para que la base de datos entienda de mapas, coordenadas y radios de alerta (geofencing).

Gestor: pgAdmin 4. Para diseñar las tablas y ver los datos visualmente.

Interfaz de Usuario y Lógica (Front-end)

Framework de Ventanas: JavaFX. Para crear una interfaz de escritorio con aspecto de "radar militar".

APIS:

- 1- <https://openskynetwork.github.io/opensky-api/rest.html>
- 2- <https://www.adsbexchange.com/>