



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
AERONÁUTICA Y DEL ESPACIO
MÁSTER UNIVERSITARIO EN INGENIERÍA AERONÁUTICA

TRABAJO FIN DE MÁSTER

**Introducción de la resolución óptima de conflictos en el
simulador de posición ATC**

AUTOR: Ariadna GÓMEZ ROZALÉN

INTENSIFICACIÓN: Aeronaves

TUTOR PROFESIONAL: V. Fernando GÓMEZ COMENDADOR

TUTOR DEL TRABAJO: Rosa María ARNALDO VALDÉS

Junio de 2021

Índice

Índice de figuras	VI
Índice de tablas	IX
Nomenclatura	XI
Prólogo	XIII
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	4
1.3. Alcance y limitaciones	5
1.4. Planificación	5
1.5. Estructura del documento	6
2. Estado del arte	9
2.1. La evolución del sistema ATM en la historia	9
2.2. La gestión de conflictos en el tráfico aéreo	12
2.3. El sistema actual de control de tráfico aéreo	13
2.4. Modernización global del sistema ATM	16
2.5. SESAR y el ATM Masterplan 2020	17
2.6. El tráfico aéreo en España: iniciativas de modernización nacionales	19
2.7. Trabajos previos	22
3. Simulador y equipos	31
3.1. Introducción	31
3.2. Servidor	32
3.3. Manager	33
3.4. ROSE	33
3.5. SkySim	34

3.6.	Equipos de medida	35
3.6.1.	Emotiv INSIGHT y EmotivPRO	35
3.6.2.	Eye tracker Gazepoint GP3 HD y Gazepoint Control and Gazepoint Analysis	37
4.	Diseño de nuevas funcionalidades	39
4.1.	Requerimientos de diseño	39
4.2.	Definición de propuestas	40
4.3.	Diseño de la solución final	43
4.3.1.	Selección del lenguaje de programación	43
4.3.2.	Programación de la aplicación	44
4.3.3.	Descripción de la aplicación	46
4.3.4.	Implementación de la nueva funcionalidad	49
4.4.	Programación de los mensajes en una simulación	52
5.	Procedimiento de validación	55
5.1.	Justificación del estudio	55
5.2.	Objetivos e hipótesis	56
5.3.	Protocolo experimental	58
5.3.1.	Participantes	58
5.3.2.	Entorno de evaluación	58
5.3.3.	Instrumentos de medida	59
5.4.	Limitaciones del estudio	61
6.	Diseño del ejercicio de validación	63
6.1.	Elementos característicos de los ejercicios	63
6.2.	Procedimiento de estimación de carga de trabajo	65
6.3.	Programación del ejercicio	66
6.4.	Programación de los mensajes en la aplicación	68
6.5.	Prueba de simulación	69
7.	Análisis de resultados	73

7.1. Medidas psicofisiológicas	73
7.2. Medidas de rendimiento	77
8. Conclusiones y líneas futuras	79
Referencias	80
A. Código de Python de la nueva funcionalidad	84
B. Parámetros del Eye Tracker	91

Índice de figuras

1.1. Crecimiento del tráfico aéreo en el mundo de 1945 a 2021. Fuente: OACI [22]	1
1.2. Predicción la recuperación del tráfico aéreo europeo a corto plazo tras la pandemia. Fuente: EUROCONTROL [9]	2
1.3. Objetivos	3
1.4. Planificación del trabajo	6
1.5. Estructura del documento	7
2.1. Controladores en 1939 separando aeronaves en ruta utilizando fichas de progresión de vuelo. Fuente: Nolan, 1990 [33]	10
2.2. Evolución del sistema ATM	12
2.3. Niveles que intervienen en la resolución de conflictos [49]	13
2.4. Estructura del sistema ATM	14
2.5. Fases de vuelo y sus centros de control. Fuente: ENAIRE [6]	15
2.6. Objetivos de SES de alto nivel. Fuente: <i>ATM Masterplan</i> [43]	17
2.7. Objetivos de SESAR 2020. Fuente: <i>ATM Masterplan</i> [43]	18
2.8. Ciclo de los programas SESAR. Fuente: [41]	19
2.9. Estado de los diferentes países europeos en la implementación del Implementing Objective ATC12.1. Fuente: SESAR JU [40]	20
2.10. Distribución del tráfico aéreo en España. Fuente: ENAIRE	21
2.11. Funcionalidades relacionadas con ER ATC 157. Fuente: [42]	28
2.12. Funcionalidades relacionadas con CM-0211. Fuente: [42]	29
3.1. Sectores de la ACC de Madrid vistos en SkySim	31
3.2. Disposición de equipos del simulador	32
3.3. Ventana del manager de monitorización de todos los equipos	33
3.4. Interfaz de ROSE con el espacio aéreo del ACC de Madrid	34
3.5. Interfaz de <i>Sky Visual</i> con los sectores BLU y BLL del ACC de Madrid asignados	35
3.6. Hardware Emotiv INSIGHT	36

3.7. Software Emotiv	37
3.8. Eye Tracker	38
4.1. Mensaje programado con ROSE	40
4.2. Herramienta de edición de SkyVisual	41
4.3. Mensajes en tiempo real entre puestos ATC	41
4.4. Ventana emergente programada en Python	42
4.5. Diseño final de la aplicación	43
4.6. Ejemplo de la nueva funcionalidad en el simulador	46
4.7. Interfaz de la nueva funcionalidad	46
4.8. Esquema de las posibles situaciones durante la simulación	48
4.9. Ventana emergente para salir del ejercicio de simulación	48
4.10. Exportación de los datos de la simulación recogidos por los botones	49
4.11. Página web para descargar el compilador de Python [34]	50
4.12. Comando de instalación de la librería <i>reportLab</i> en la ventana de comandos .	50
4.13. Cambiar la ubicación de los archivos	51
4.14. Cómo ejecutar el programa con el compilador de Python	52
4.15. Función para programar mensajes	53
5.1. Tres escenarios de costes diferentes en la implementación de una nueva funcionalidad en función de la estrategia de integración de los factores humanos en el proceso. Fuente: EUROCONTROL [12]	55
5.2. Etapas del procesamiento clásico de la información	56
5.3. Escenarios posibles de respuesta. En gris se representa la carga de trabajo de diseño, en azul claro la respuesta del sujeto sin funcionalidades nuevas y en azul oscuro la respuesta del sujeto con funcionalidades nuevas.	57
5.4. Ley de Yerkes-Dodson. Fuente: [29]	59
5.5. NASA TLX	60
6.1. Iconos del callsign de las aeronaves en función de la responsabilidad de los controladores.	63

6.2.	Resolución de un conflicto por re-routing.	64
6.3.	Resolución de un conflicto por directo a un punto.	64
6.4.	Carga de trabajo asociada a la monitorización de aeronaves	66
6.5.	Programación de ejercicios en ROSE y cálculo de trayectorias	68
6.6.	Caption	69
6.7.	Interfaz de la simulación sin la resolución óptima de conflictos (21 de mayo)	70
6.8.	Interfaz de la simulación con la resolución óptima de conflictos (23 de junio)	71
6.9.	Desarrollo de la simulación	71
7.1.	Datos exportados del Eye Tracker	73
7.2.	Fijaciones con alta y baja carga de trabajo durante la simulación	75
7.3.	Número de fijaciones por minuto	76
7.4.	Tiempo de duración de las fijaciones	76
7.5.	Frecuencia de parpadeo	77
7.6.	Resultado de los reports de los botones	78

Índice de tablas

1.1. Tasa de crecimiento anual de pasajeros-kilómetro transportado (pax-km) de 2006 a 2015 en todo el mundo. Fuente: OACI [19]	2
2.1. Puntuaciones de las funcionalidades. En azul el nivel máximo, en gris nivel aceptable, en naranja nivel descartable y en blanco nivel inaceptable.	27
3.1. Configuración numérica de los sectores del ACC de Madrid	32
4.1. Función de las librerías utilizadas	44
6.1. Puntuación de los eventos de los ejercicios	65
6.2. Tabla de eventos del ejercicio de validación	67
7.1. Variación de los parámetros al aumentar la carga de trabajo	74
B.1. Parámetros exportados del <i>Eye Tracker</i> (I)	91
B.2. Parámetros exportados del <i>Eye Tracker</i> (II)	92

En primer lugar, quiero dar las gracias a mis profesores del MUIA por su predisposición y profesionalidad a pesar de la pandemia, y en especial a Fernando porque gracias a su flexibilidad el camino ha sido mucho más fácil.

A mis compañeros de SATAA, que siempre me han ayudado en todo lo que he necesitado aunque tuvieran que adelantar el despertador.

A mis amigos, por animarme siempre a seguir hacia delante.

A mis padres y a mi hermano por su apoyo incondicional y su eterna paciencia durante toda esta etapa.

En definitiva, gracias a todos los que habéis hecho posible que haya llegado hasta este momento.

Prólogo

El control aéreo ha evolucionado mucho a lo largo de la historia respondiendo a las necesidades del tráfico aéreo. En el siglo XXI el principal problema que deben enfrentar los controladores es la congestión del tráfico y por consiguiente el elevado número de aeronaves que deben tener bajo su responsabilidad. Esto les ocasiona una alta carga de trabajo que hay que reducir para dar servicio al tráfico que se espera en el futuro. Una de las maneras de reducir esta carga de trabajo es automatizando sus funciones y mediante la introducción de herramientas que ayuden reducir su carga de trabajo y optimizar su rendimiento.

La introducción de nuevas funcionalidades en las posiciones ATC tiene un impacto directo en el desempeño de los controladores, y es por ello que se debe estudiar el efecto de nuevas funcionalidades, pues estas herramientas pueden ser óptimas para el funcionamiento del sistema ATM pero tener un efecto negativo en la carga de trabajo del controlador.

En este contexto, han surgido numerosas iniciativas de modernización del sistema ATM, entre ellas la más importante es el programa SESAR, cuyas soluciones son el punto de partida de este trabajo. Entre las soluciones de SESAR este trabajo aborda la implementación de una nueva funcionalidad de resolución óptima de conflictos entre aeronaves.

Como el simulador del Laboratorio UPM no contaba con ninguna herramienta que pudiera adaptarse a este fin, se ha programado una aplicación en Python. Esta aplicación se ejecuta en el mismo ordenador donde se realiza la simulación y aparece en forma de ventana emergente en la pantalla.

Los ejercicios están programados y por lo tanto se pueden anticipar las soluciones de los conflictos para mostrarlas por pantalla y de esta manera mejorar su desempeño. Para comprobar cómo afecta esta nueva funcionalidad en la carga de trabajo de los controladores, se ha propuesto un plan de validación para dar recomendaciones acerca del desarrollo futuro de esta funcionalidad.

A lo largo de este documento, se explican los pasos seguidos para introducir esta nueva funcionalidad y sentar las bases de este estudio de validación.



UNIVERSIDAD
POLITÉCNICA
DE MADRID



1. Introducción

El objetivo de este capítulo es introducir el Trabajo de Fin de Máster: Introducción de la resolución óptima de conflictos en el simulador de posición ATC. En él se contextualiza el proyecto y se definen los principales objetivos, así como el alcance y las limitaciones del mismo. Por último se incluye la planificación que se ha llevado a cabo a lo largo del trabajo y se expone la estructura del documento.

1.1. Motivación

Desde los inicios del transporte aéreo a principios del siglo XX hasta la actualidad el tráfico aéreo ha evolucionado en el tiempo convirtiéndose en un elemento clave para la economía mundial. La evolución de la industria aeronáutica se debe en parte a la gran cantidad de operadores existentes que están continuamente buscando la eficiencia operacional, lo que da lugar a un mercado muy competitivo.

En la figura 1.1 se muestra la evolución del tráfico de pasajeros desde 1945 hasta 2021. En la figura se evidencia la fuerte dependencia del crecimiento del tráfico aéreo con la situación económica como la guerra del Golfo en los años 90, la crisis del 11-S en 2001 (ataques terroristas en Estados Unidos), la crisis financiera de 2008 (recesión económica) y la actual crisis sanitaria en 2020 y 2021.

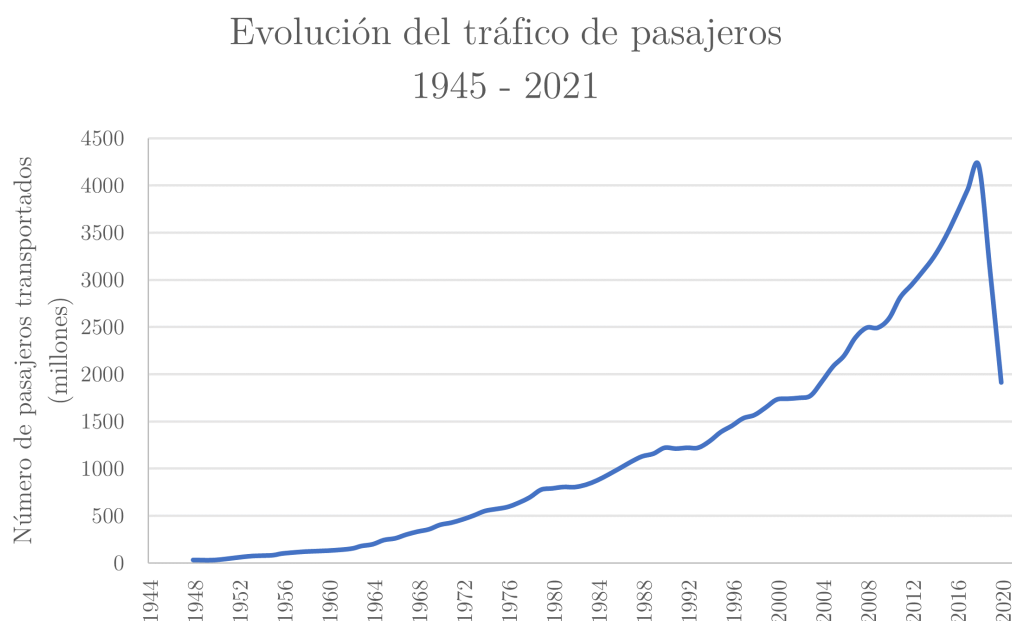


Figura 1.1: Crecimiento del tráfico aéreo en el mundo de 1945 a 2021. Fuente: OACI [22]

Para cuantificar este crecimiento, en la tabla 1.1 se presentan las tasas de crecimiento anual de pasajeros-kilómetros transportado (pax-km) de 2006 a 2015. La consecuencia de este gran crecimiento es que el número de operaciones que el sistema ATM debe gestionar aumenta significativamente cada año por lo que es necesario adaptar la capacidad del sistema a las necesidades del tráfico aéreo.

Año	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Variación anual de pax-km (%)	6.2	8.2	2.0	-1.1	8.0	6.6	5.3	5.5	6.0	7.1

Tabla 1.1: Tasa de crecimiento anual de pasajeros-kilómetro transportado (pax-km) de 2006 a 2015 en todo el mundo. Fuente: OACI [19]

La crisis sanitaria y económica actual originada por la pandemia mundial de la COVID-19 ha supuesto un cambio sin precedentes en el sector aéreo, si embargo, no hay duda que el tráfico aéreo se recuperará y continuará creciendo con el paso de los años, por lo que el sistema ATM debe estar preparado para dar servicio a toda la demanda de tráfico aéreo en el futuro. Según EUROCONTROL [10], entre 1997 y 2017 el tráfico aéreo europeo aumentó un 31.7%, llegando a alcanzar un total de 10.6 millones de vuelos en 2017, cifras que comprometen la capacidad el sistema ATM actual. En la figura 1.2 se puede comprobar que de acuerdo a las predicciones de EUROCONTROL a corto plazo el sistema se recuperará en unos pocos años, lo que significa que volverá a tener unas altas tasas de crecimiento en el medio y largo plazo.

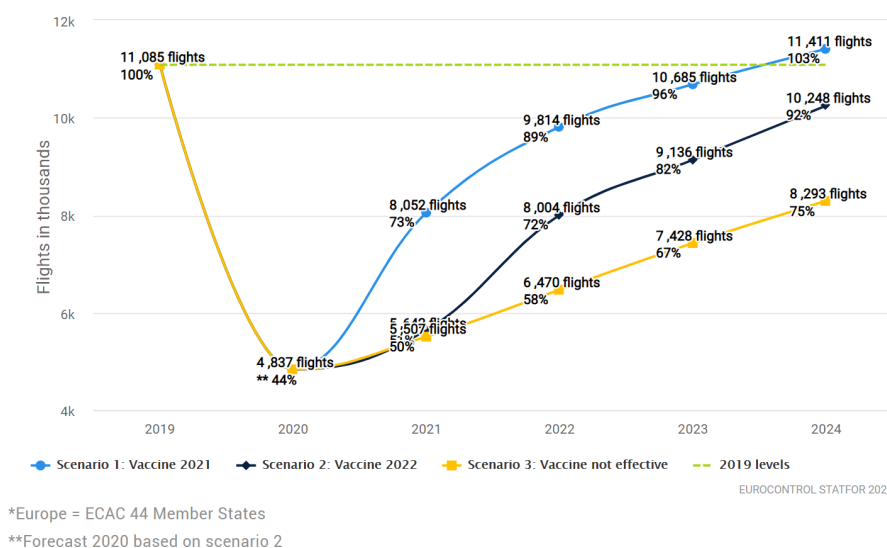


Figura 1.2: Predicción la recuperación del tráfico aéreo europeo a corto plazo tras la pandemia. Fuente: EUROCONTROL [9]

El objetivo del sistema ATM es abastecer la demanda de tráfico aéreo gestionando sus recursos de manera óptima. Actualmente en las zonas de mayor densidad de tráfico aéreo como Europa el gran volumen de operaciones que tienen lugar ha dado lugar a retrasos y otras ineficiencias relacionadas con la falta de capacidad del sistema. En el futuro estas ineficiencias se agravaran debido al crecimiento de la demanda de tráfico aéreo, lo que causará la saturación del sistema ATM y el desabastecimiento de la demanda.

Para evitar esta situación y aumentar la capacidad han aparecido iniciativas globales para la modernización del sistema ATM. El *Single European Sky* es el encargado de llevar a cabo esta modernización en Europa a través del programa *Single European Sky ATM Research* (SESAR).

El punto de partida de este trabajo es el *ATM Masterplan 2020* de SESAR. En particular, este trabajo evalúa una de las principales líneas de modernización que es aumentar la capacidad declarada asegurando niveles adecuados de efectividad y seguridad. La estimación de esta capacidad se limita, entre otros factores, por la alta carga de trabajo de los controladores aéreos (*Air Traffic Control Operator, ATCO*). Con el crecimiento del tráfico aéreo los controladores tienen que asumir un volumen mayor de aeronaves. En este contexto, el objetivo de este trabajo es la disminución de esta carga de trabajo, para que los controladores sean capaces de controlar de forma segura el tráfico aéreo que se espera en el futuro.

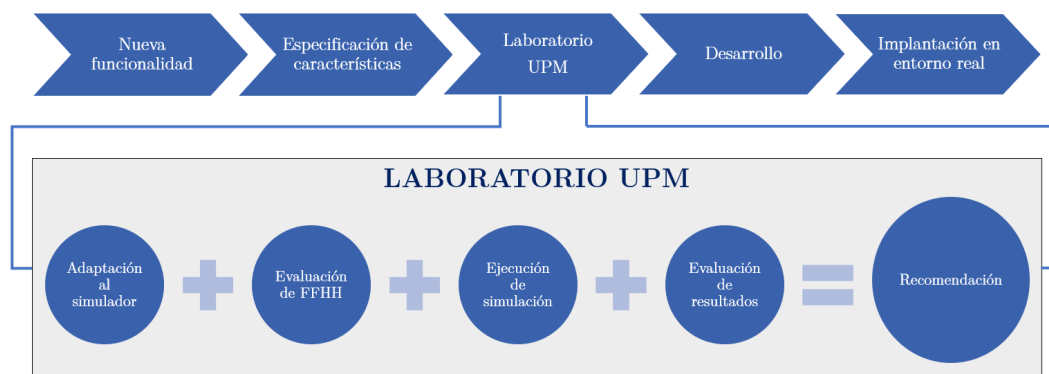


Figura 1.3: Objetivos

La figura 1.3 esquematiza el proceso de introducción de nuevas funcionalidades en el entorno real del tráfico aéreo. Hasta llegar ahí, cada mejora debe pasar por una serie de pasos encaminados a asegurar su viabilidad y a comprobar cómo afecta a la carga de trabajo de los controladores.

Una de las contribuciones dentro de este proceso es el Laboratorio UPM. Desde la Universidad Politécnica de Madrid (UPM) se pretende valorar las posibles nuevas funcionalidades a añadir al sistema de control aéreo que se proponen desde organismos internacionales como

las que provienen del programa SESAR, y de emitir recomendaciones en consecuencia para su posterior desarrollo. La misión del Laboratorio UPM es, a partir de una nueva mejora propuesta, adaptarla al simulador, definir los puntos de control y la metodología de cómo medirlos, ejecutarla en el simulador con controladores aéreos potenciales y analizar los resultados de las mejoras introducidas, especialmente desde el punto de vista de los factores humanos.

A partir del análisis de esos resultados se emiten las recomendaciones oportunas para determinar si la mejora es viable y se puede implementar en el entorno real. Hay que tener en cuenta que los cambios dentro del control aéreo deben ser estudiados minuciosamente, debido a la trascendencia de su función. Existe un alto riesgo en cualquier modificación no evaluada hasta el último detalle. Dentro de estas mejoras y de las pruebas que se diseñan en la UPM hay que destacar el análisis de cómo afectan los FFHH a cada una de las posibles modificaciones. Los controladores aéreos están sometidos a una gran presión y estrés por la naturaleza de su trabajo. Como hemos visto en los párrafos anteriores, se espera un gran crecimiento tanto en el número de aeronaves como en el tráfico de pasajeros. Por ello, para aumentar la capacidad declarada se debe aliviar la carga de trabajo de los controladores, pues en el corto plazo va a crecer el número de aeronaves que tienen que controlar. Por todo esto, es primordial ver cómo afecta cada mejora a la carga de trabajo del controlador aéreo y como los FFHH que hay a su alrededor pueden afectar o no a dichas mejoras. Aunque sea mínimamente, cualquier modificación introducida en los sistemas de gestión del tráfico les puede afectar negativamente. Hay mucho dinero y muchas vidas humanas que pueden estar en riesgo si cualquier medida no tiene los efectos esperados o si no se evalúa adecuadamente cómo pueden afectar estas medidas a los propios controladores aéreos.

De esta manera, una vez que el Laboratorio UPM ha ejecutado la simulación, interpretado los resultados y emitido las recomendaciones oportunas para el desarrollo de nuevas funcionalidades son los proveedores de control de tráfico aéreo (ANSPs) los que deben decidir si desarrollar las funcionalidades valoradas. No obstante, parten de un estudio de viabilidad desde el punto de vista de los FFHH, que puede ahorrar esfuerzo y dinero a la hora de desarrollar las nuevas funcionalidades.

1.2. Objetivos

Como se ha anticipado en el apartado anterior, el fin último de este trabajo es proporcionar una recomendación acerca de la implementación de una nueva herramienta para disminuir la carga de trabajo de los controladores aéreos. Para que esto sea posible, en este trabajo se han definido los siguientes objetivos:

- A partir del Trabajo de Fin de Grado de Juan Antonio Murcia [32], seleccionar las

funcionalidades del *ATM Masterplan* que se van a añadir en el simulador de posición ATC de *SkySoft-ATM* que se encuentra en la Universidad Politécnica de Madrid.

- Diseño e implementación de la nueva funcionalidad de resolución óptima de conflictos en el simulador.
- Elaboración de un plan de actuación para la validación de las nuevas funcionalidades desde el punto de vista de los factores humanos.
- Propuesta de metodología de evaluación de resultados.
- Prueba de validación con un sujeto

1.3. Alcance y limitaciones

El alcance de este proyecto comprende la definición, diseño y programación de la ventana emergente con mensajes programados temporalmente, las pruebas de validación para verificar que funciona correctamente cuando se integra dentro del simulador de puesto del operador de ATC y la elaboración de una metodología para evaluar los resultados cuando se integre su uso. Como únicamente se ha realizado una prueba con un sujeto, los resultados no se van a poder extrapolar y generalizar, por lo tanto, queda fuera del alcance de este trabajo sacar conclusiones respecto a la carga de trabajo analizando los resultados extraídos de las pruebas de validación, aunque sí se han procesado los datos para dar una guía de tratamiento para líneas futuras.

1.4. Planificación

La planificación del proyecto se presenta en la figura 1.4 donde se puede observar que se ha realizado en un periodo de 21 semanas. El proyecto se ha dividido en cuatro fases diferentes:

- Fase 0: Estado del arte
- Fase 1: Definición de la nueva funcionalidad
- Fase 2: Diseño del procedimiento de validación
- Fase 3: Resultados

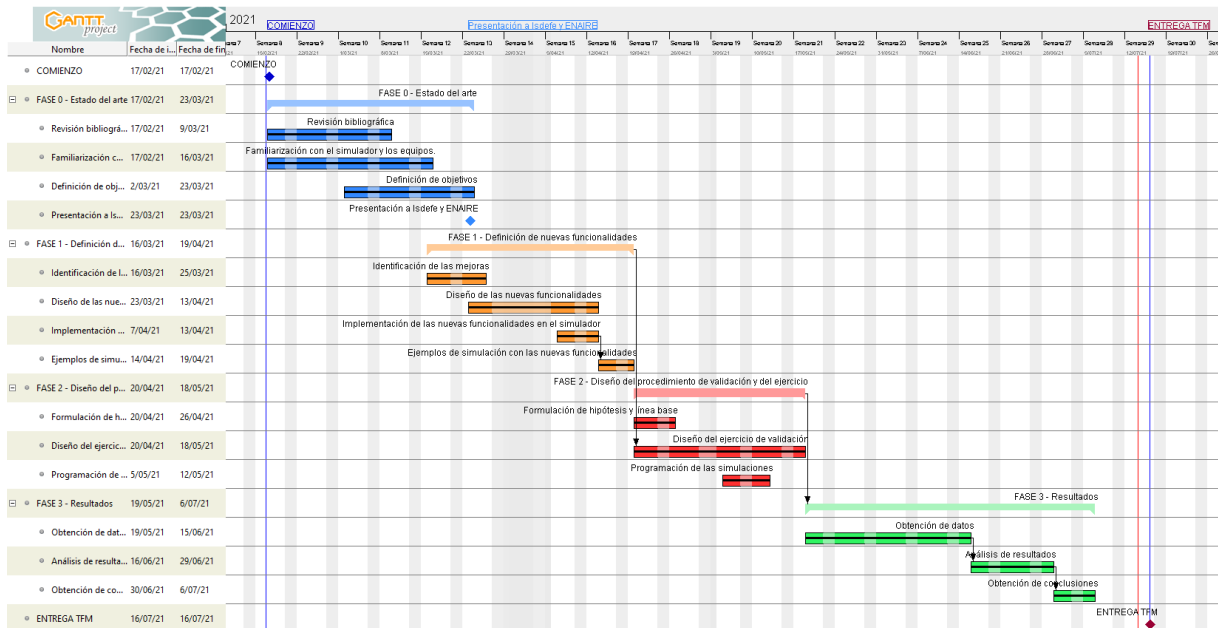


Figura 1.4: Planificación del trabajo

1.5. Estructura del documento

Este documento está estructurado en 8 capítulos ordenados como se muestra en la figura 1.5. El contenido de los capítulos es el siguiente:

1. Introducción: contiene la motivación del proyecto, los objetivos, el alcance y limitaciones y la estructura del documento.
2. Estado del arte: constituye el estado del arte del trabajo. En él se describe la evolución del sistema ATM en la historia, cómo se gestionan los conflictos y cómo se estructura el sistema ATM y hacia donde se espera que evolucione. Además, se incluye una visión general del tráfico aéreo en España y los progresos de modernización nacionales. Por último se incluye un análisis del TFG de Juan Antonio Murcia [32] que es el antecedente de este trabajo.
3. Simulador y equipos: en este capítulo se describen los diferentes elementos del simulador, así como los equipos que se utilizan para la toma de datos.
4. Diseño de nuevas funcionalidades: este capítulo incluye los requerimientos de diseño de la nueva funcionalidad, las diferentes propuestas valoradas y finalmente el diseño de la solución final.

5. Procedimiento de validación: se trata de una propuesta de estudio para la validación de la nueva funcionalidad. Siguiendo la estructura de los estudios con sujetos, se divide en justificación del estudio, objetivos e hipótesis a demostrar, protocolo experimental y limitaciones del estudio.
6. Diseño del ejercicio de validación: pasos seguidos para la programación del ejercicio de simulación, incluyendo la estimación de la carga de trabajo y la realización de una prueba de validación.
7. Análisis de resultados: se analizan los principales resultados de la prueba de validación.
8. Conclusiones y líneas futuras

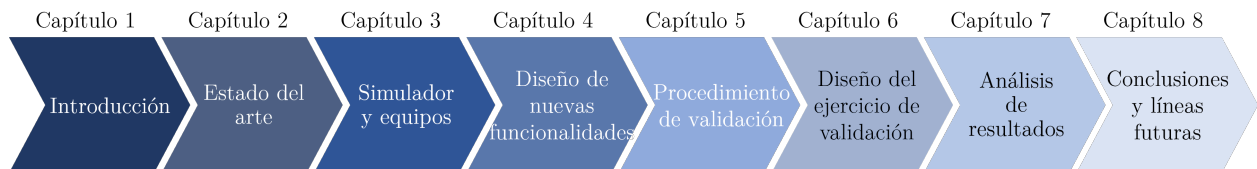


Figura 1.5: Estructura del documento



UNIVERSIDAD
POLITÉCNICA
DE MADRID



2. Estado del arte

2.1. La evolución del sistema ATM en la historia

Históricamente, el sistema ATM ha evolucionado de acuerdo a las necesidades de la industria aeronáutica, es decir, el sistema se ha ido adaptando a la demanda de tráfico aéreo. A pesar de esta evolución, su principal objetivo permanece invariante con el paso del tiempo: garantizar la seguridad y la eficiencia del transporte aéreo. La OACI distingue cinco periodos en la evolución del sistema ATM en la historia: periodo pionero, entreguerras, globalización, liberalización y siglo XXI.

■ Periodo pionero (1903 - 1919)

El tráfico aéreo comenzó el 17 de diciembre de 1903 con el primer vuelo de los hermanos Wright. En los inicios de la aviación, la densidad de tráfico aéreo era suficientemente baja como para que las aeronaves pudieran separarse confiando en la identificación visual del tráfico realizada por el piloto. Se adoptó la regla de "mantenerse a la derecha", originaria del transporte marítimo y terrestre, y esto era suficiente para mantener las aeronaves separadas.

La separación en tierra apareció alrededor de 1920 con el aumento del tráfico aéreo. Los controladores aéreos, desde el aeropuerto, usaban ayudas visuales (banderas y luces) para comunicarse con los pilotos, informándoles de las condiciones de la pista, la dirección del viento y de la presencia de otras aeronaves. El primer aeródromo en implantar este servicio fue el aeródromo de Croydon en 1916, donde se autorizaba o denegaba a las aeronaves a aterrizar y despegar con luces verdes o rojas.

■ Periodo entreguerras (1920 - 1943)

Después de la Primera Guerra Mundial, numerosos bombarderos se convirtieron en aviones de transporte de mensajería, lo que aumentó la densidad del tráfico aéreo, y aunque la separación visual seguía siendo suficientemente segura, en 1919 la Comisión Internacional de la Navegación Aérea (CINA), perteneciente a las Naciones Unidas, estandarizó las primeras normas de tráfico aéreo que fueron suscritas inicialmente por 19 estados.

En las décadas de 1920 y 1930, la introducción de la radio supuso una mejora significativa, pues proveía a los pilotos y controladores de un canal de comunicación bidireccional. Como la posición de las aeronaves no se podía determinar de manera exacta, esta separación se hacía de acuerdo a procedimientos pre-establecidos donde la posición de la aeronave se estimaba basándose en la última posición reportada por el piloto, lo que se conoce como control

convencional. Durante este periodo, aparece el concepto de aerovía (*airway*), se desarrollaron también los primeros procedimientos de salida de un aeródromo y desde la torre de control, el controlador de ruta situaba a los aviones en un mapa en base a las señales de radio que los aviones enviaban y de esta manera se seguía la progresión del vuelo.

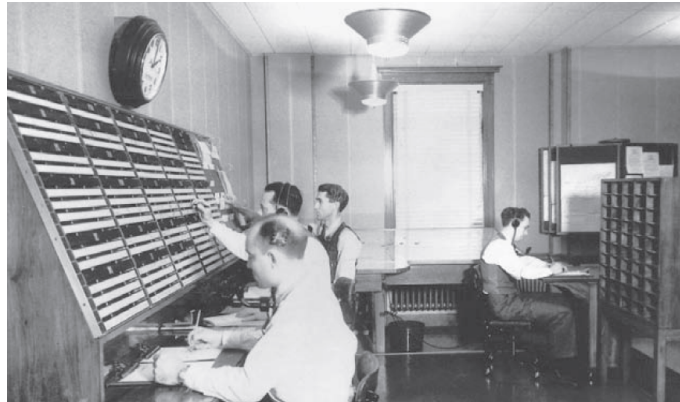


Figura 2.1: Controladores en 1939 separando aeronaves en ruta utilizando fichas de progresión de vuelo. Fuente: Nolan, 1990 [33]

■ Periodo de globalización (1944 - 1977)

Tras la Segunda Guerra Mundial, se define el sistema ATM en Convenio de Chicago y se estandarizan las normas y procedimientos de vuelo en todo el mundo. Muchas aeronaves militares se convirtieron en civiles, lo que supuso un gran aumento de la densidad del tráfico aéreo comparado con el tráfico antes de la guerra. Además, las mejoras tecnológicas en las aeronaves, especialmente en velocidad y alcance, impidieron la operación segura de las aeronaves en áreas de alta ocupación como las TMAs, por lo que era necesario mejorar el sistema ATM. Otro hito asociado a la Segunda Guerra Mundial fue la introducción del radar. Los radares se utilizaban para detectar aeronaves y otros vehículos a través de las ondas reflejadas por el vehículo y capturadas por un receptor en tierra. Esta tecnología habilitaba operaciones en todo tipo de condiciones meteorológicas. La implementación del radar produjo un cambio en la manera de controlar el tráfico aéreo desde el control convencional al control por radar.

Con la introducción del radar, la situación dentro de su cobertura mejoró la estimación de posición. En la aproximación con control convencional, la posición se estimaba con la última posición reportada por el piloto, mientras que con el radar se podía conocer la posición exacta de las aeronaves. Este cambio supuso una gran mejora de la capacidad del espacio aéreo. El tráfico aéreo se separaba por autorizaciones ATCO basadas en la posición de la aeronave y la estimación de su futura posición (*clearance-based separation*).

Durante la década de 1950, se introdujeron en la aviación comercial los aviones a reacción, lo que supuso un gran cambio en términos de actuaciones, que llevó a la segmentación del espacio aéreo en diferentes sectores y niveles de vuelo para separar los aviones a reacción rápidos de las aeronaves más lentas. Además, la vigilancia del tráfico aéreo y la navegación se mejoró significativamente con el uso de balizas en tierra como el VOR (*Very high frequency Omnidirectional Range*). El uso del VOR, habilitaba el posicionamiento de las aeronaves relativo a la baliza en tierra, lo que mejoró significativamente la precisión de la navegación del tráfico aéreo y contribuyó a la creación de una red de rutas.

En los años 60, el radar primario evolucionó al radar secundario (*Secondary Surveillance Radar, SSR*), que además de dar al controlador la posición y dirección de la aeronave, también proporciona la identificación de la aeronave y su altitud.

■ **Periodo de liberalización (1978 - 1999)**

Entre los años 70 y 80, el transporte aéreo continuó evolucionando y se introdujeron las aeronaves de alta capacidad, las aeronaves supersónicas y los turbofanos de largo alcance. Todo esto, junto a la liberalización del tráfico aéreo, fomentó el aumento transporte de pasajeros por todo el mundo lo que desembocó en la necesidad de mejorar la infraestructura en tierra para abastecer la demanda del tráfico aéreo. Un ejemplo de estas mejoras es la introducción del ILS (*Instrument Landing System*), que permite operaciones en condiciones de visibilidad reducida en aproximación y aterrizaje en pista. En los años 90, el principal problema era cómo abordar la congestión de la red con el desarrollo de nuevas herramientas de predicción y comunicación entre controladores y pilotos, apareció la necesidad de gestionar flujos de tráfico de aeronaves, *Air Traffic Flow Management (ATFM)*.

■ **Siglo XXI (2000 - 2021)**

Después de la introducción del SSR, el sistema ATM no ha sufrido ningún otro avance tecnológico revolucionario. El radar primario y secundario (SSR) siguen siendo actualmente fundamentales para la vigilancia ATC a pesar de la introducción del ADS-B y esta falta de nuevos grandes desarrollos en el sistema han contribuido a un gran número de ineficiencias que necesitan ser corregidas. Como resultado, a principio del siglo XXI aparecieron muchos programas de modernización basados en el *ICAO's Global ATM Operational Concept* que están en proceso de actualizar los sistemas y procedimientos ATM con el principal objetivo de cambiar del control radar a operaciones basadas en la trayectoria (*Trajectory-Based Operations, TBOs*) conocido como PBN (*Performance Based Navigation*). Las TBOs están basadas en separar aeronaves por *trajectory clearances* en lugar de la separación actual basadas en *radar clearances*.

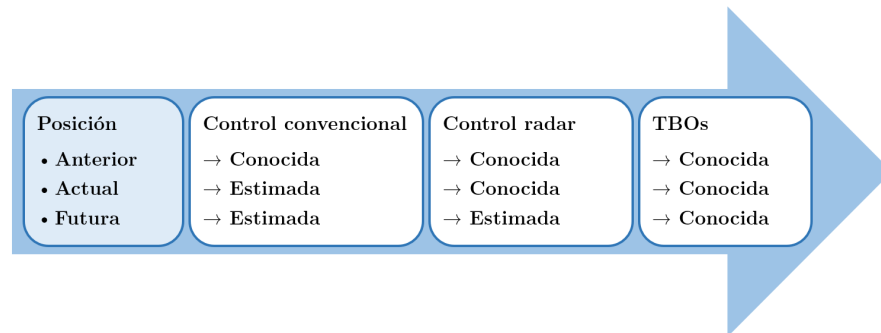


Figura 2.2: Evolución del sistema ATM

La introducción de TBOs supondrá un tráfico aéreo más eficiente, pues la separación de aeronaves se realizará con mayor anticipación gracias a la mejora en las herramientas de predicción y a que el seguimiento de trayectorias será más preciso. El conocimiento del estado futuro de la aeronaves dará un rol más importante a las funciones de gestión estratégica de la capacidad de tráfico aéreo (ASM y ATFCM).

2.2. La gestión de conflictos en el tráfico aéreo

Los conflictos son situaciones que involucran a una o varias aeronaves que se enfrentan a peligros en los que se compromete el nivel de seguridad mínimo para la operación del vuelo. Este nivel de seguridad se garantiza a través de la separación de aeronaves de manera que siempre se mantenga una separación mínima entre ellas. Cuando se infringe esta separación mínima tiene lugar un incidente que debe ser reportado.

Ante un potencial conflicto, hay que garantizar que el riesgo de colisión se reduzca a un nivel aceptable de seguridad. La resolución de conflictos es una parte fundamental en el trabajo de un controlador aéreo. Gran parte de las mejoras introducidas en el control de tráfico aéreo sirven para optimizar y fortalecer la resolución de estos conflictos, de cara a que la carga de trabajo que tiene un controlador no se vea afectada seriamente ante un aumento de tráfico, y en consecuencia, de los conflictos.

En la figura siguiente se presentan los diferentes niveles que intervienen cuando se presenta un conflicto. Los ATC intervienen tanto en el nivel táctico de separación basada en la trayectoria como en la capa subsiguiente de normalización de las trayectorias después de los movimientos automáticos de los sistemas anti-colisión (Safety Nets) para evitar el peligro. El presente trabajo se centra en la capa táctica, en cómo se puede disminuir la carga de trabajo de un controlador haciendo que sea más fácil la resolución de conflictos. Si se adelanta la resolución óptima del conflicto y se muestran las indicaciones al controlador para resolver

conflictos sin perjudicar al tráfico, su carga de trabajo probablemente disminuirá y por tanto podrá atender más situaciones cuando el nivel de tráfico crezca.

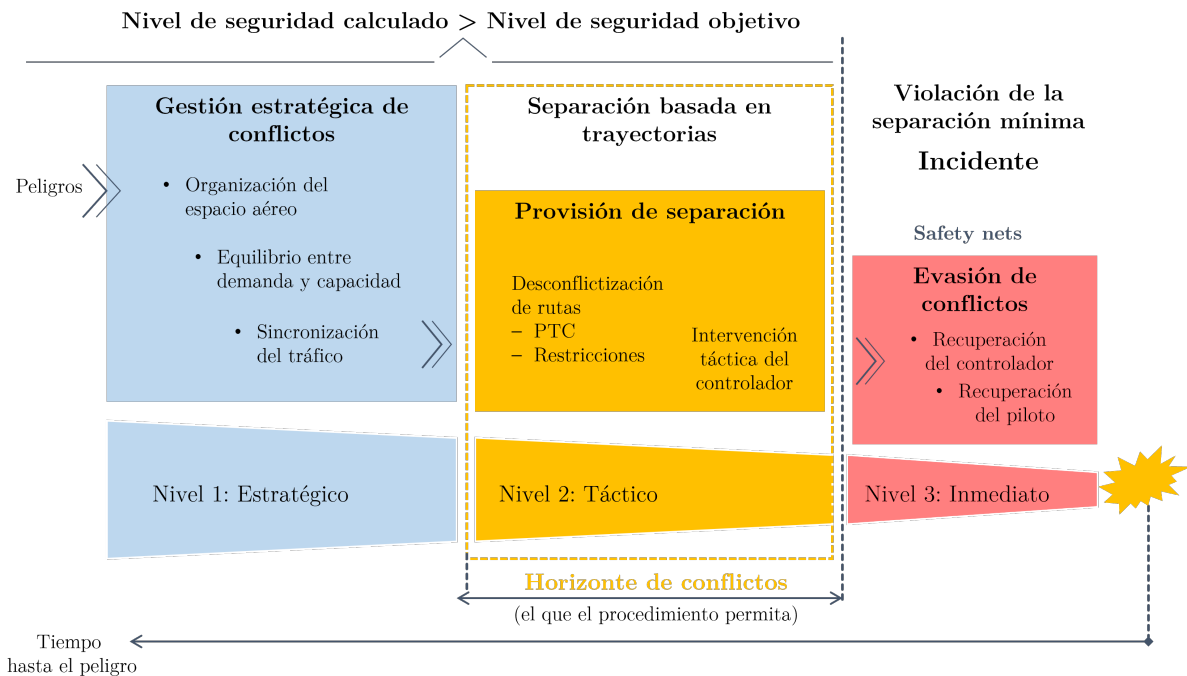


Figura 2.3: Niveles que intervienen en la resolución de conflictos [49]

2.3. El sistema actual de control de tráfico aéreo

Los servicios de tránsito aéreo (*Air Traffic Services, ATS*) actuales están recogidos en el Anexo 11 del Convenio de Chicago [21]. Los ATS están definidos como un concepto que engloba servicios de información, alerta y asesoramiento de vuelo y los servicios de control de tráfico aéreo. Su objetivo es prevenir y evitar colisiones entre aeronaves y entre aeronaves y tierra, mantener el orden en el tránsito aéreo, informar y asesorar a las aeronaves para que sus vuelos sean eficaces y seguros y notificar a los organismos pertinentes respecto aeronaves que necesitan ayuda de búsqueda y salvamento y auxiliar a estos organismos en sus tareas.

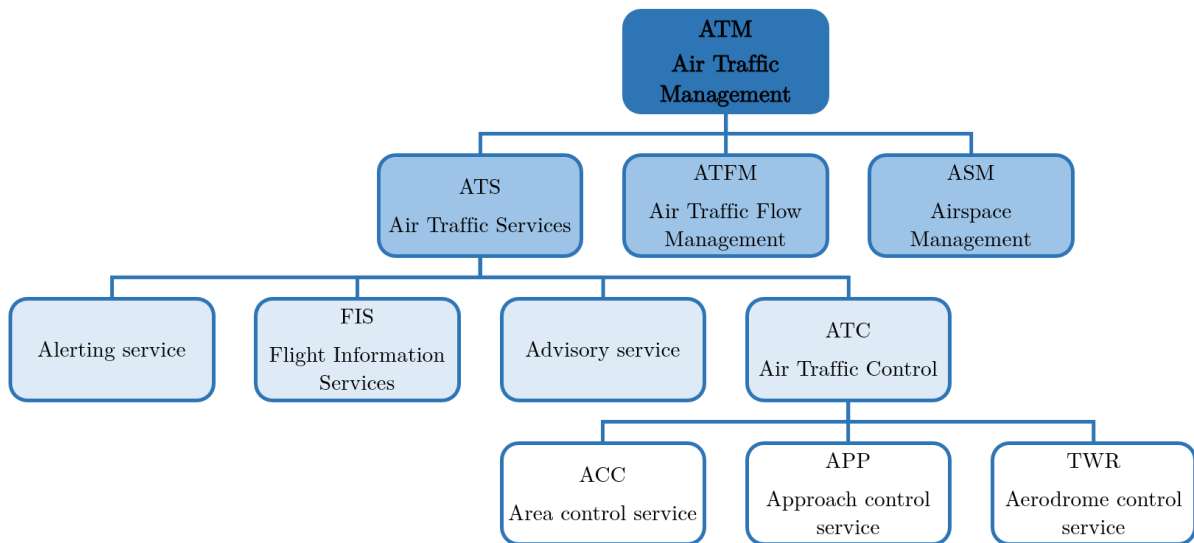


Figura 2.4: Estructura del sistema ATM

En la figura 2.4 se ha representado la estructura del sistema ATM. En ella se puede ver que los ATS se dividen en cuatro categorías: servicios de alerta, servicios de información de vuelo (*Flight Information Services*, FIS), servicios de asesoramiento y servicios de control del tráfico aéreo (*Air Traffic Control*, ATC). Estos últimos se subdividen en servicios de control de área (ACC), servicios de control de aproximación (APP) y servicios de tránsito aéreo en el aeródromo (TWR).

Los controladores aéreos son los encargados de proporcionar estos servicios de ATC. Su principal función es garantizar la seguridad del tráfico aéreo manteniendo la separación vertical y horizontal entre aeronaves para evitar que haya conflictos entre ellas. Además, una aeronave en un vuelo controlado debe estar bajo el control de un único controlador a la vez, por lo que los controladores aéreos deben coordinar la transferencia y asunción de aeronaves que circulan por sus sectores asignados.

Existen tres tipos de controladores atendiendo a la naturaleza del tráfico aéreo:

- Controlador de aeródromo: gestiona el tráfico en sus fases de rodadura, despegue y aterrizaje.
- Controlador de aproximación: gestiona el tráfico en el área terminal (TMA) en la fase anterior al aterrizaje y la posterior al despegue.
- Controlador de ruta: gestiona el tráfico durante la fase de crucero entre aeropuertos.

Los controladores de ruta (ACC ATCO) están divididos en dos equipos: el controlador ejecutivo o táctico (EC) y el controlador coordinador (PC). El EC es el responsable de la

separación del tráfico en su sector operativo y el PC ayuda al controlador ejecutivo en sus tareas y asegura la coordinación entre sectores. Este trabajo se centra en el controlador ejecutivo.

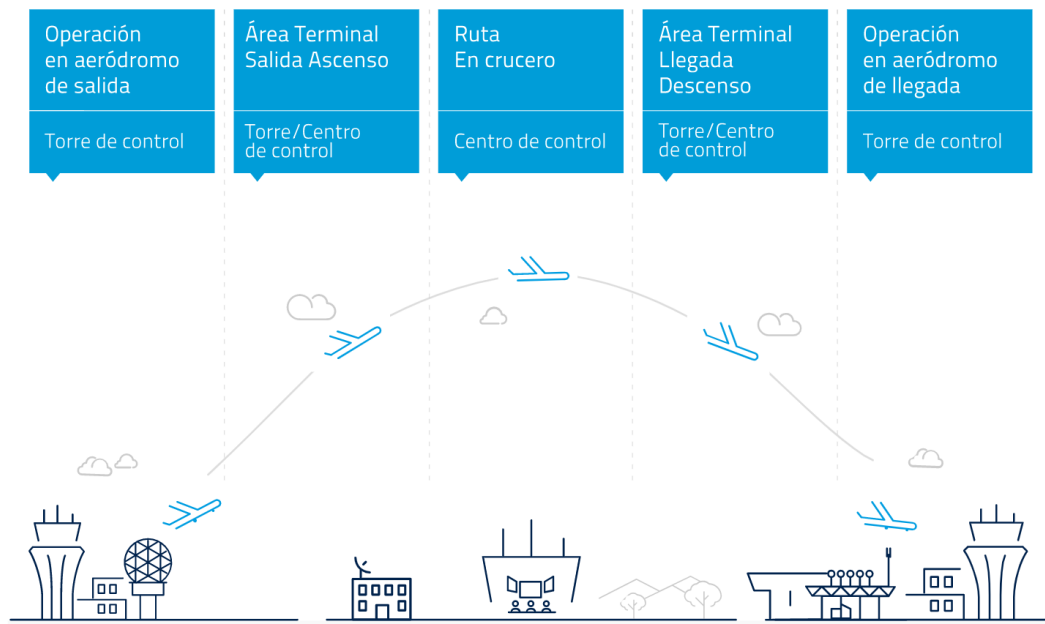


Figura 2.5: Fases de vuelo y sus centros de control. Fuente: ENAIRE [6]

Los controladores aéreos son los responsables de la seguridad de los vuelos. Como el número de vuelos que tienen lugar es muy elevado, se divide el espacio aéreo en sectores de manera que cada controlador sea responsable de su sector asignado. En áreas de alta demanda, es necesario imponer restricciones para asegurar que el número de vuelos de un sector no exceda un límite especificado. Este límite es la capacidad del sector y se define como el número de vuelos que pueden asumir los controladores aéreos en un área, en un tiempo específico sin perjudicar la seguridad y sin causar una carga de trabajo excesiva para el controlador (en la práctica, el número aproximado de aeronaves para que un sector esté saturado va de 10 a 15 aeronaves). La sobrecarga del controlador aéreo es una limitación fundamental para aumentar la capacidad del sistema.

Existen diferentes definiciones de capacidad de un sector:

- La capacidad horaria de un sector es el número máximo de vuelos que entran en él en una hora que pueden ser asignados de forma segura a los controladores. Lo que significa que el controlador de ese sector pueda procesar todos los vuelos sin exceder un umbral de sobrecarga de trabajo (*overload threshold*).

- La capacidad declarada de un sector es el número de vuelos que entran por hora evaluada por ANSP (*Air Navigation Service Provider*) y declarada a una CFMU (*Central Flow Management Unit*), *Network Manager* o cualquier organismo que realice funciones de ATFM de manera que el sector pueda estar protegido frente a sobrecargas.

Por todo esto, antes de introducir cambios en el sistema ATC, como un nuevo concepto de espacio aéreo, nuevos procedimientos de control o nuevas tecnologías, es necesario identificar el efecto que tienen estos cambios en la carga de trabajo de los controladores. Los métodos para determinar la capacidad de un sector atendiendo a la carga de trabajo de un controlador se dividen en dos grupos: estimaciones percibidas o medidas. EUROCONTROL ha diseñado un modelo matemático denominado CAPAN para medir la carga de trabajo del controlador y con él se define la capacidad de un sector. El principal inconveniente de CAPAN es que es un procedimiento largo y complejo.

2.4. Modernización global del sistema ATM

Para poder abastecer a toda la demanda de tráfico aéreo numerosas instituciones a nivel mundial, europeo y nacional han iniciado en los últimos años planes de innovación para la adaptación del sistema actual a las demandas del futuro tráfico aéreo.

A nivel mundial, en colaboración con otras instituciones globales como IATA, OACI ha publicado el *Global Air Navigation Plan (GANP)*, [23]. Este documento es de carácter estratégico y contiene el plan mundial de navegación aérea. Su principal objetivo es impulsar la evolución del sistema actual en una misma dirección para todos los miembros de OACI, apoyando la planificación e implementación de medidas a nivel mundial, regional y nacional.

En Europa aparece la iniciativa *Single European Sky (SES)* para implementar las trayectorias basadas en operaciones (TBOs) o PBNs mencionadas anteriormente. Para que esto sea posible, en el año 2004 se creó SESAR JU (*Single European Sky ATM Research Joint Undertaking*) cuyo objetivo principal es garantizar el desarrollo sostenible del transporte aéreo en Europa de forma eficiente y segura.

En Estados Unidos aparece en 2007 el programa *NextGen* liderada por las FAA para la modernización del sistema ATM americano, así como mejorar las predicciones y avanzar hacia las operaciones basadas en trayectorias. Actualmente se encuentra en la mitad de su plan de implementación que tiene su fecha fin en 2030.

2.5. SESAR y el ATM Masterplan 2020

En Europa, en 2004 se creó la iniciativa *Single European Sky* con el objetivo de lograr una aviación sostenible y eficaz, destacando la importancia del tráfico aéreo como motor de la economía europea. Los objetivos de alto nivel propuestos en 2005 para el tráfico aéreo en 2035 se presentan en la figura 2.6: triplicar la capacidad del sistema ATM, reducir retrasos y costes, reducir las emisiones de CO₂ y aumentar la seguridad

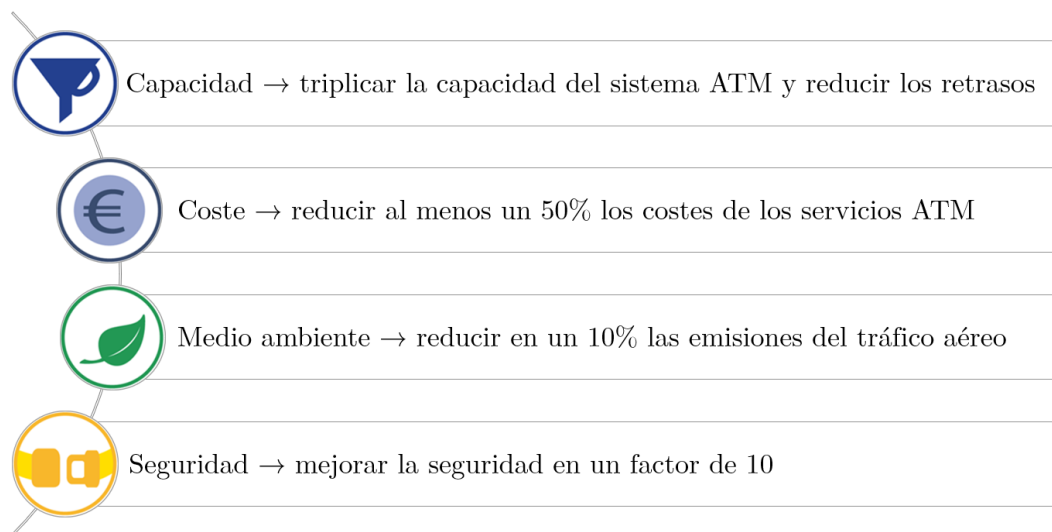
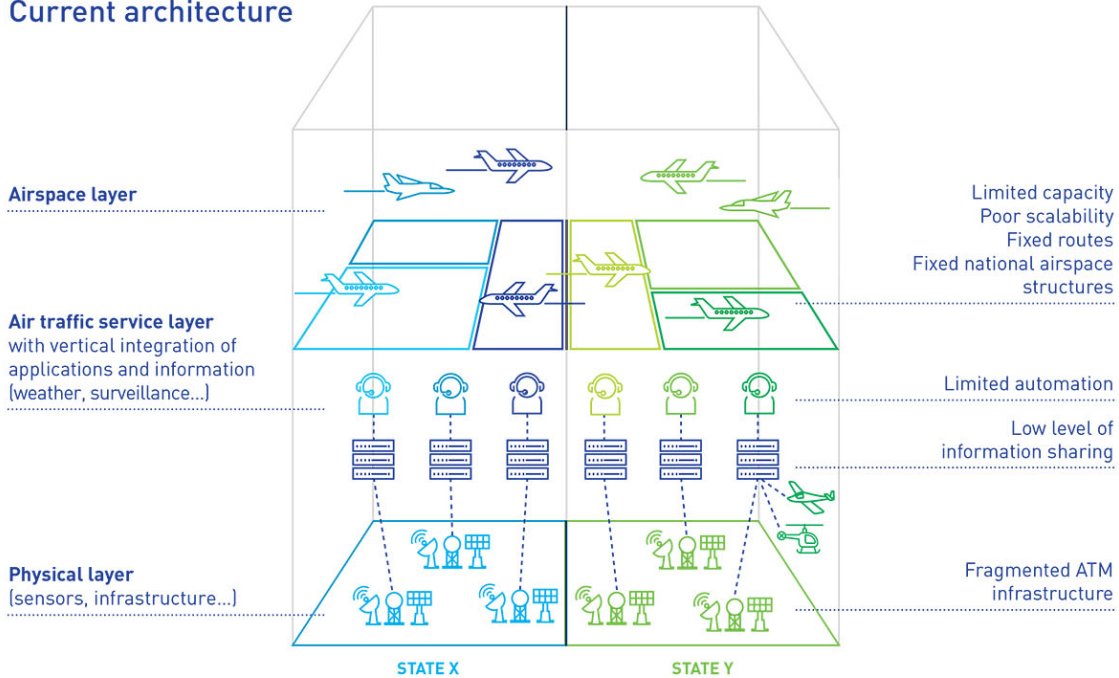


Figura 2.6: Objetivos de SES de alto nivel. Fuente: *ATM Masterplan* [43]

Para poder cumplir estos objetivos se creó el programa SESAR (*Single European ATM Research*) que se encuentra actualmente en su segunda versión: SESAR 2020. La primera versión, SESAR 1, obtuvo como resultado un aumento del 11 % de capacidad en el aeropuerto, un 33 % en el TMA y un 34 % en ruta, además de reducir los costes de los servicios de navegación en un 4 %, las emisiones por vuelo en un 2.3 % y el tiempo de retraso de los vuelos un 30 %. Todos estos resultados se lograron entre 2008 y 2016 sin aumentar el número de accidentes a pesar del incremento del tráfico aéreo.

Actualmente se encuentra en progreso el programa SESAR 2020 cuyos objetivos se muestran en la figura 2.7. Entre estos objetivos se incluyen la automatización de las comunicaciones, la integración de los diferentes tipos de aeronaves en el espacio aéreo, la unificación de los servicios de información y la integración de los diferentes sistemas.

Current architecture



Future architecture

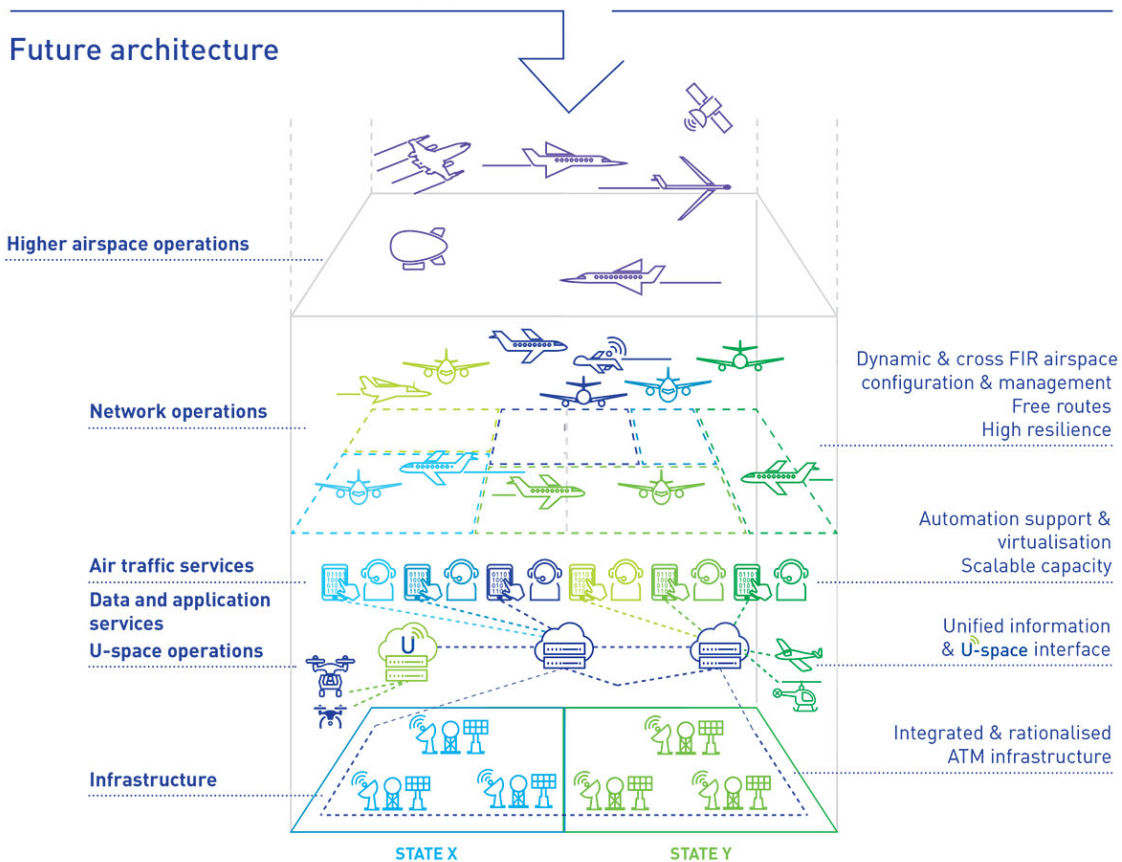


Figura 2.7: Objetivos de SESAR 2020. Fuente: *ATM Masterplan* [43]

El programa SESAR 2020 está dividido en 3 fases: definición, desarrollo y despliegue, las cuales se alimentan unas a otras dando a lugar un proceso cíclico como se muestra en la figura 2.8.

1. Fase de definición: en ella se ha entregado el *ATM Masterplan 2020* [43] que se define como la herramienta principal de planificación para definir las prioridades para la modernización del sistema ATM y garantizar que el concepto de futuro se convierta en una realidad.
2. Fase de desarrollo: en esta fase se producen los nuevos sistemas tecnológicos, infraestructura y procedimientos operacionales según se definen en el *ATM Masterplan*. Se elaboran las soluciones *SESAR Solutions* [45], que agrupan las soluciones en 4 áreas: operaciones de alto rendimiento en el aeropuerto, servicios de tráfico aéreo avanzados, red de servicios ATM optimizada y habilitar infraestructura aeronáutica necesaria.
3. Fase de despliegue: se implementa a gran escala la nueva infraestructura de gestión del tráfico aéreo.



Figura 2.8: Ciclo de los programas SESAR. Fuente: [41]

2.6. El tráfico aéreo en España: iniciativas de modernización nacionales

En el contexto nacional, tienen lugar las *Local Single Sky Implementation (LSSIP)* [14], que se tratan de implementaciones de los proyectos definidos en SESAR que deben ser implementados en el ámbito nacional.

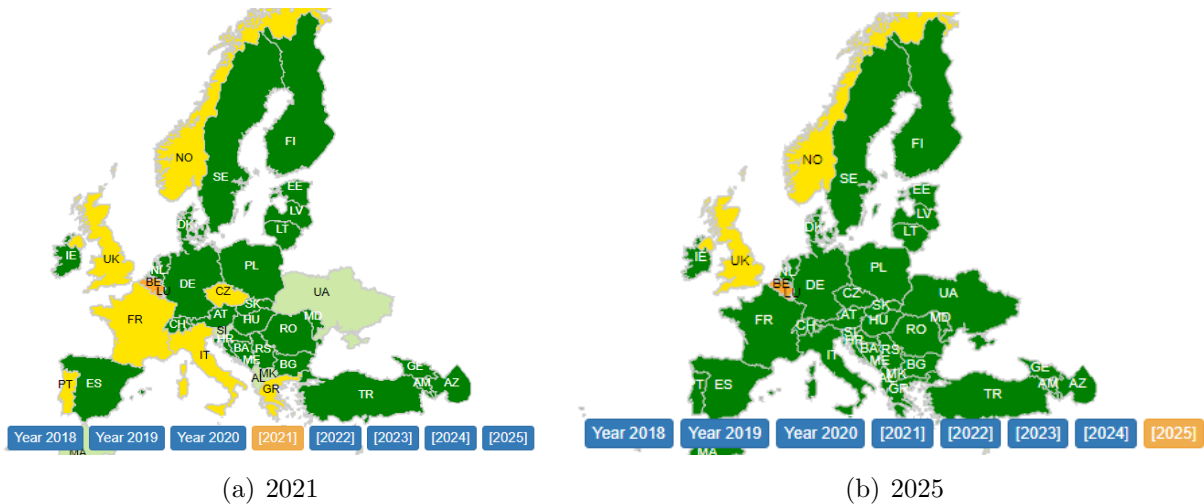


Figura 2.9: Estado de los diferentes países europeos en la implementación del Implementing Objective ATC12.1. Fuente: SESAR JU [40]

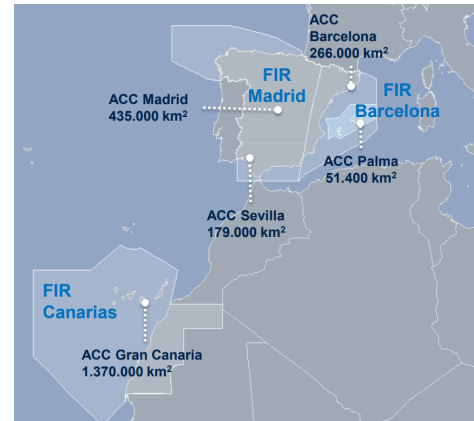
En la figura 2.9 se muestra un ejemplo de uno de los planes los objetivos de implementación de un objetivo del programa SESAR por países. En él se indica en verde oscuro los países que ya han completado su objetivo anual, en verde claro los que están siguiendo la programación para implementarlo, en amarillo los que se han retrasado respecto a su plan y en naranja los que aún no han planeado ese objetivo.

El objetivo de implementación más relacionado con este trabajo es el ATC12.1: *Automated Support for Conflict Detection, Resolution Support Information and Conformance Monitorig*, que según el eATM Portal en junio de 2021 va por el 68 % de su implementación en Europa.

En la figura 2.10(a) se puede observar que en España el principal ANSP es ENAIRE, aunque existen otros proveedores privados y militares. ENAIRE como principal proveedor de servicios de navegación aérea gestiona un espacio aéreo de 2.19 millones de kilómetros cuadrados a través de las 5 ACCs que se muestran en la figura 2.10(b): Madrid, Sevilla, Barcelona, Palma y Canarias. Como es el principal gestor, ENAIRE establece el Plan de Vuelo 2020 [7] para la modernización e innovación tecnológica del sistema de navegación aérea donde se definen sus planes nacionales alineados con los objetivos del *Single European Sky*.



(a) Mapa de servicios ATS en España



(b) FIRs y ACCs de España

Figura 2.10: Distribución del tráfico aéreo en España. Fuente: ENAIRE

ENAIRE cuenta con SACTA, el Sistema Automatizado de Control de Tránsito Aéreo, que es un sistema que se encarga de la gestión del control de tráfico aéreo. Su misión es facilitar la prestación de los servicios ATS (de tránsito aéreo) de los que ENAIRE es responsable. Integra todos los centros de control de ruta, aproximación y aeródromo españoles, de forma que la gestión se realiza sobre datos coherentes y de una manera coordinada. De esta forma facilita una operación sin interrupciones entre todas las dependencias de control.

En el ámbito internacional, SACTA permite la comunicación automática entre centros de control domésticos e internacionales. Para ello utiliza estándares internacionales de intercambio de datos, reduce las actuaciones manuales al mínimo, detecta automáticamente posibles conflictos y aporta flexibilidad para la reconfiguración del espacio aéreo operacional. También permite minimizar los efectos de los denominados picos de tráfico. El sistema SACTA evoluciona continuamente y se actualiza mediante versiones, para que los cambios se puedan implementar de forma progresiva.

Estas actualizaciones se desarrollan en parte en el programa SACTA 4.0 que se trata de un proyecto dentro de ENAIRE cuyo objetivo principal es investigar y llevar a cabo el estudio de nuevas funcionalidades para implementarlas en el simulador del laboratorio del SACTA para posteriormente implementarlas en los sistemas de navegación aérea de ENAIRE.

Las funciones principales de SACTA son:

- Procesar toda la información relativa a los planes de vuelo así como gestionar las herramientas de ayuda a la planificación del tráfico.
- Procesar la información radar para realizar funciones de seguimiento e identificación de las aeronaves y mantener la separación del tráfico aéreo.

- Coordinar la asociación entre la información radar y los planes de vuelo, de forma que el controlador pueda identificar las aeronaves de forma automática e inequívoca.
- Presentar información meteorológica para ayudar a la navegación en condiciones adversas.
- Dar alertas al controlador de las desviaciones de las aeronaves con respecto a la planificación, o con respecto a las autorizaciones recibidas en tiempo real.
- Garantizar la comunicación de datos entre las aeronaves y el controlador, permitiendo el intercambio de mensajes con las aeronaves y, de esa forma, evitar comunicaciones orales.
- Presentación de la información aeronáutica necesaria para el control de tráfico aéreo.
- Supervisar, configurar, grabar y explotar todo el sistema de forma distribuida.

2.7. Trabajos previos

A continuación, se va a realizar una revisión de las nuevas funcionalidades identificadas por Juan Antonio en su Trabajo de Fin de Grado [32].

Las nuevas funcionalidades se han identificado a partir de un exhaustivo estudio del ATM Masterplan [43], en el cual se definen nuevas funcionalidades agrupadas en diferentes categorías: *SESAR Solutions* (SOL), *Operational Improvement Steps* (OI), *Enablers* (EN), *Pilot Common Project Elements* (PCP), *Implementation Objectives* (OBJ) y *SESAR Projects* (PJ). Profundizando en todas estas categorías se han seleccionado un total de 177 nuevas funcionalidades que son de interés para el objetivo del proyecto y se han agrupado dentro de 26 conceptos o áreas técnicas que se enumeran a continuación.

1. Eliminación de voz

El objetivo de la eliminación de voz es establecer diálogo electrónico y eliminar la comunicación oral entre el controlador y la tripulación. Para ello se debe definir un protocolo de transmisión de información electrónica en la tripulación. Esto se puede aplicar también al intercambio de información civil-militar y a la asistencia automatizada durante la coordinación y transferencia entre sectores. Además, uno de los objetivos de la eliminación de voz es colaborar en la implementación del proyecto PJ.24 (*Network Collaborative Management*).

2. Secuenciar salidas y llegadas involucrando la fase de ruta

Secuenciar la fase de ruta según las salidas más tempranas con ayuda de DMAN (Departures Management) y secuenciar los descensos ya en la fase de ruta con ayuda de AMAN (Arrival Management).

3. Vuelos ATC

Consiste en la realización de vuelos céntricos dentro del ATC (*Air Traffic Control Zone*) sin necesidad de tener un controlador específico asignado al control de la aeronave cuando el sector está sobrecargado. Esta funcionalidad está relacionada con la iniciativa de suprimir la sectorización del espacio aéreo.

4. Conflictos en la *taxiway*

Detección y predicción de conflictos en la *taxiway* en función de la posición, velocidad y dirección del vehículo. Además, se incluye dentro de esta funcionalidad el monitoreo de la pista y la alerta de conflicto, asistencia de orientación a través de iluminación en pista, alerta en caso de incursión en pista o área restringida, optimización de la ruta de rodaje y más funcionalidades relacionadas.

5. Conflictos en la *airway*:

El objetivo es la prevención y detección de conflictos en ruta y su resolución mediante asistencia automatizada con la maniobra óptima. Esta funcionalidad involucra el monitoreo de conformidad de autorizaciones ATC en ruta, procedimientos ATC para helicópteros, alertas de intrusión de RPAS y de conflictos a corto plazo (STCA), predicción de ocupación de aerovías no correspondientes según el plan de vuelo y otras muchas funcionalidades.

6. Densidad de torre

El objetivo es la estimación de la densidad de tráfico en torre para evitar colapsos futuros, lo que requiere un estudio capacidad-demanda con registro de datos en el aeropuerto.

7. Avisos meteorológicos

Dar información meteorológica en tiempo real, monitorizar posibles turbulencias usando LIDAR/radar e incorporar sistemas de alertas críticas en el tiempo meteorológico.

8. Asistencia automática

Esta funcionalidad tiene como objetivo dar asistencia a la resolución de conflictos en el TMA/ATC y alertas de conformidad (CMAC) para controladores.

9. Separación mínima

El objetivo es reducir la separación mínima o RVR en crucero y aumentar los niveles de vuelo disponibles.

10. CTA/CTOT (Calculated Time of Arrival/ Calculated Take-Off Time)

El objetivo es la secuenciación óptima de salida de aeronaves.

11. Estimación de posición

Uso de los datos del ADS-B para la estimación de posición de otras aeronaves.

12. Trayectoria 4D

El objetivo es la integración de trayectorias 4D, lo que requiere un correcto rediseño del espacio aéreo.

13. FL disponible

El objetivo de esta funcionalidad es indicar los niveles de vuelo disponibles y ocupados para prevenir y detectar conflictos en las trayectorias de aeronaves tanto de entrada y salida de un sector como en el interior del sector.

14. Detección temprana

Asistencia al controlador en la identificación de conflictos y las tareas de planificación. Se proporciona una detección temprana automatizada de posibles conflictos, de manera que se facilite la identificación de rutas flexibles y trayectorias libres de conflictos, así como las aeronaves que impiden la resolución del conflicto o que ocupan un nivel de vuelo solicitado por otra aeronave.

15. Free route

Evaluación de rutas libres en entornos de complejidad medio/baja con directrices y waypoints para evaluar su correcto desarrollo.

16. Separación óptima

El objetivo de esta funcionalidad es la separación óptima de salida entre aeronaves para evitar conflictos en ruta. Para ello, el tiempo de ocupación en pista y altura prevista durante el despegue deben ser evaluados.

17. Sectorización dinámica del ATC

El objetivo es el control dinámico del espacio aéreo basado en complejidad, para ello, debe existir soporte automatizado para evaluar la complejidad del espacio aéreo en tiempo real y la capacidad de cambiar la sectorización en función ésta.

18. CCO y CDO (*Continuous Climb and Descent Operations*)

Operaciones de ascenso y descenso continuas en vez de escalonadas.

19. Planificador único

El objetivo de esta funcionalidad es tener varios controladores tácticos bajo la responsabilidad de un único controlador planificador.

20. Operaciones segmentadas

Esta funcionalidad consiste en dar soporte a las operaciones segmentadas en el TMA. Este soporte está relacionado con la separación, secuenciación y monitorización de cada segmento, por ejemplo, detección y resolución de conflictos, perfil de descenso de aproximación intermedia, senda de planeo, etc.

21. HMI/Autoseparación

Esta funcionalidad consiste en mejorar la interfaz hombre-máquina (HMI) para que sea capaz de identificar e indicar los vuelos autorizados para mantener la separación de forma automática.

22. Servicios de torre remota

El objetivo de esta funcionalidad es dar servicio ATC a varios aeródromos desde una ubicación remota y no desde la torre de control de alguno de los aeropuertos que da servicio. De esta funcionalidad hay muchos proyectos en fases avanzadas que se pueden ver en [26].

23. Medición de flujos

El objetivo es la gestión de flujos y del espacio aéreo. Esta funcionalidad está dirigida a la medición de flujos de llegada para anticiparse a rutas que se cruzan o intersectan en la TMA y a la medición de perfiles verticales de aeródromo. También está relacionada con el uso del plan de vuelo extendido.

24. Low visibility

Mejora de las operaciones en condiciones de baja visibilidad en despegue, rodaje y aterrizaje.

25. Redes de seguridad

El objetivo de esta funcionalidad es tener redes de seguridad para pilotos y controladores en el aeródromo. Esto supone tener requerimientos operacionales y especificaciones técnicas para el sistema de detección de conflictos para que exista un control de conformidad ante las autorizaciones ATC de resolución de conflictos.

26. SACTA 4.0

SACTA 4.0 es el programa de ENAIRE para la automatización del control del tráfico aéreo. Funcionalidad DAPS: Downlink Aircraft Parameters Modo S, que proporciona

al controlador de ruta y TMA de cualquier centro de control disponer de datos reales actualizados de las aeronaves: GS, TAS, FL rumbo, derrota, alabeo entre otras.

Para poder identificar las funcionalidades de mayor interés se han clasificado de acuerdo a los siguientes criterios con un sistema de puntuación numérica:

- Nivel de Adaptación (NA): mayor o menor dificultad de introducir la funcionalidad en el simulador. La puntuación es 12 puntos para funcionalidades integradas, 8 puntos compensadas, 4 puntos comprometidas y 0 puntos excluidas.
- Funcionalidades Relacionadas (FR): establece la puntuación según el número de funcionalidades relacionadas en el ATM Masterplan. La puntuación es 8 puntos si hay más de 9 funcionalidades relacionadas, 6 puntos si las funcionalidades relacionadas son entre 9 y 7, 4 puntos si las funcionalidades relacionadas están entre 6 y 4 y 2 puntos si las funcionalidades relacionadas son menos de 4.
- Fase de Vuelo (FV): puntuación de acuerdo a la fase de vuelo en la que se aplica la funcionalidad debido a que el simulador es únicamente de ruta. La puntuación es 8 puntos si se aplica en ruta, 6 puntos en ascenso o descenso, 4 puntos en aproximación y 2 puntos en rodaje.
- Fecha de Finalización (FF): de acuerdo a la fecha prevista de finalización, se da 4 puntos si la funcionalidad se prevé implementar antes de 2025 y 0 puntos si es después.
- Información adicional (IA): este criterio da 2 puntos a las funcionalidades que se implementan aportando información adicional en la pantalla.

	Funcionalidad	NA	FR	FV	FF	IA	Total
1	Eliminación voz	12	4	8	4	2	28
2	Secuenciar salidas	8	6	2	4	2	20
3	Vuelos ATC	4	2	4	0	0	10
4	Conflicto taxiway	8	8	2	4	2	22
5	Conflicto airway	12	8	8	4	2	32
6	Densidad torre	4	4	2	4	2	14
7	Aviso meteorológico	12	4	8	4	2	28
8	Asistencia Auto.	8	4	8	0	2	20
9	Separación mínima	8	2	8	4	2	22
10	CTA/CTOT	4	4	6	0	2	14
11	Estimación posición	4	2	8	4	2	18
12	Trayectoria 4D	0	6	8	0	0	14
13	FL disponible	12	6	8	4	2	30
14	Detección temprana	12	2	8	4	2	26
15	Free route	0	6	8	0	0	14
16	Separación óptima	4	2	8	4	2	18
17	Sec. dinámica ATC	0	4	8	0	2	12
18	CDO/CCO	0	2	4	4	0	10
19	Planificador único	0	6	8	0	0	14
20	Op. segmentadas	4	6	2	4	0	16
21	Autoseparación	4	4	8	4	2	20
22	Torre remota	4	8	8	4	0	24
23	Medición flujos	4	6	6	4	0	20
24	Low visibility	0	2	8	4	2	14
25	Redes seguridad	4	2	8	4	2	18
26	SACTA 4.0	8	2	8	4	2	22

Tabla 2.1: Puntuaciones de las funcionalidades. En azul el nivel máximo, en gris nivel aceptable, en naranja nivel descartable y en blanco nivel inaceptable.

En la tabla 2.1 se visualizan las puntuaciones de las funcionalidades. Para seleccionar las funcionalidades de interés se han impuesto dos criterios. EL primero, que tengan una puntuación máxima de más de 22 puntos, y el segundo que tengan la puntuación máxima en 4 o más de los criterios.

Atendiendo a estos dos criterios, los grupos donde se engloban las funcionalidades de interés son los cinco siguientes: eliminación de voz, conflicto en la *airway*, aviso meteorológico, FL disponible y detección temprana.

Para implementar en el simulador se han elegido, de entre todas las posibles, dos funcionalidades:

- **ER ATC 157:** sistema ATC mejorado para dar soporte al controlador táctico en la detección y resolución de conflictos en ruta.

La herramienta de gestión de conflictos se actualizará para integrar los conflictos causados por los diferentes tipos de trayectorias con el fin de detectar conflictos: de entrada al sector, dentro del sector y de salida del sector, así como los conflictos de aeronaves con autorizaciones abiertas (por ejemplo, militares) o que se desvían de su trayectoria planificada por autorización de ATCO. La resolución es asistida indicando los niveles disponibles y ocupados y permitiendo al controlador sondear conflictos en una trayectoria hipotética. El objetivo de esta funcionalidad es actualizar la HMI del controlador para proporcionar la información de conflicto para el controlador táctico.

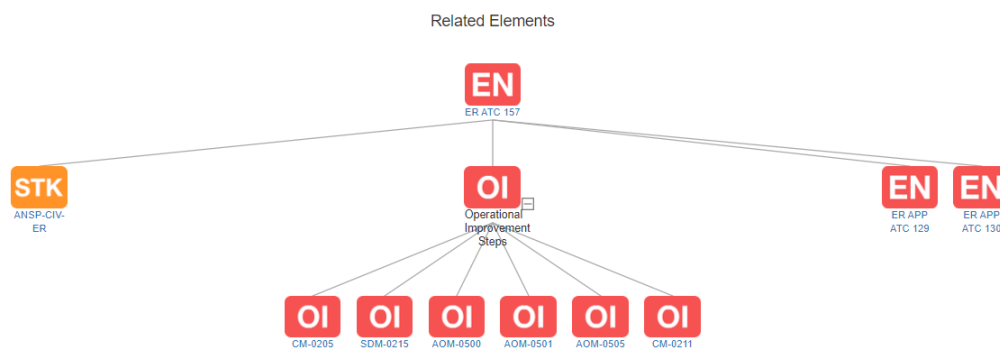


Figura 2.11: Funcionalidades relacionadas con ER ATC 157. Fuente: [42]

- **CM-0211:** sistema avanzado de detección y resolución de conflictos para la planificación de ATC en ruta.

El sistema proporcionará asistencia en tiempo real al planificador de ruta ATC para la detección y resolución de conflictos. Está basado en nuevos enfoques que mejorarán y refinarán las herramientas existentes dando lugar a servicios más eficientes y útiles.

Para proveer de herramientas más eficientes y automatizadas de detección y resolución de conflictos comparadas con la preexistentes, esta propuesta necesita de un enfoque innovativo para mejorar las herramientas actuales o añadir nuevas funcionalidades. Las nuevas o mejoradas funcionalidades deben abarcar:

- Mejorar la predicción de qué conflictos ocurren cuando una autorización es aplicada.

- Proponer cambios de trayectoria o de nivel de vuelo a los ATCO para la resolución de conflictos.
- Evaluación automática de ciertos tipos de tráfico en su entrada al sector gracias a la coordinación e integración entre sectores.
- Pantalla que muestra continuamente los riesgos de conflicto incluyendo: los riesgos de contexto de coordinación (cuando se predice que el conflicto estará dentro del área de responsabilidad) y los riesgos de contexto de interés (cuando los conflictos tendrán lugar dentro del área de interés).

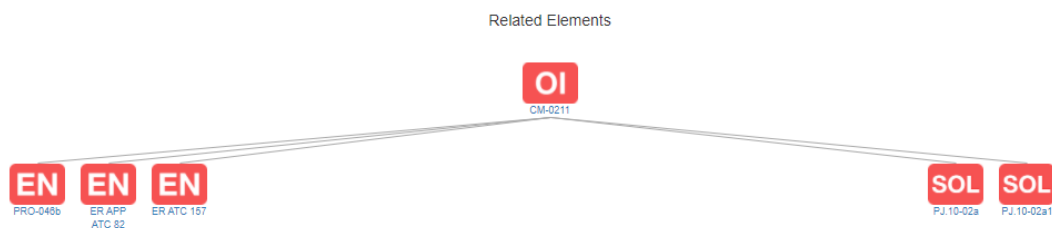


Figura 2.12: Funcionalidades relacionadas con CM-0211. Fuente: [42]

Por lo tanto, el objetivo de este trabajo es añadir al simulador del laboratorio ATC una nueva funcionalidad que proporcione la resolución óptima de conflictos.



UNIVERSIDAD
POLITÉCNICA
DE MADRID



3. Simulador y equipos

El objetivo de este apartado es realizar una breve descripción del simulador y de los equipos del laboratorio de simulación de ATC.

3.1. Introducción

Se trata de un simulador de posiciones ATC de *Skysoft ATM*, que es el proveedor de software del ANSP suizo *Skyguide*. El simulador está configurado para simular posiciones de controladores de ruta y se encuentra cargado el espacio aéreo de la ACC de Madrid que como se puede ver en la figura 3.1 está dividido en espacio aéreo superior e inferior y en algunos sectores medio. No obstante, es posible cargar otros espacios aéreos o programarlo en modo TMA en lugar de ACC, de forma que se pueden simular posiciones de controladores de aproximación además de ruta.



(a) Lower Level

(b) Upper level

Figura 3.1: Sectores de la ACC de Madrid vistos en SkySim

El simulador está formado por los 5 equipos que se presentan en la figura 3.2. De izquierda a derecha estos equipos son:

- Servidor (*Skyserver*): en este equipo se aloja virtualmente la simulación y se ajusta la sectorización del espacio aéreo.
- Manager (*SkySimMonitoring*): es el encargado del funcionamiento coordinado de todos los equipos

- ROSE: en este equipo se realiza el diseño y construcción de las simulaciones ATC.
- SkySim1 (*SkyVisual*): es el equipo que permite simular el rol de controlador ATC en las simulaciones diseñadas en ROSE:
- SkySim2 (*SkyVisual*): equipo con las mismas funciones que SkySim1. Ambos equipos pueden acceder a la misma simulación de forma simultánea e interactuar entre ellos.

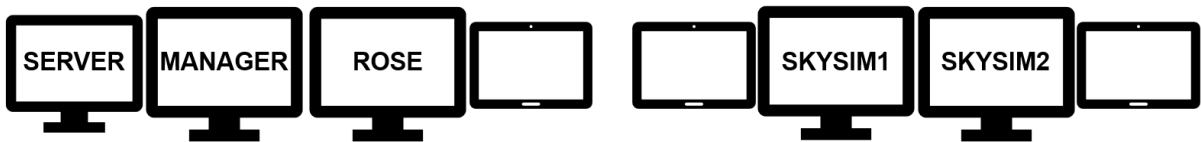


Figura 3.2: Disposición de equipos del simulador

3.2. Servidor

El equipo servidor tiene dos funciones fundamentales: alojar virtualmente la simulación y ajustar la sectorización ATC del ejercicio de simulación.

El espacio aéreo que se utiliza en el laboratorio es el espacio aéreo del ACC de Madrid. Cada sector del espacio aéreo utilizado en ROSE y en *SkyVisual* tiene un determinado número asociado como se muestra en la tabla 3.1. Para asignar a cada posición ATC su sector correspondiente, se debe hacer coincidir el número asignado a la posición ATC con el número asignado al sector en la interfaz de *SkyServer* del equipo servidor.

ASL	ASU	BLL	BLU	CJL	CJU	DGL	DGU	PAL	PAU	SAL	SAU	TER	TLA	TLL	TLU	ZGZ	ZML	ZMM	ZMU
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Tabla 3.1: Configuración numérica de los sectores del ACC de Madrid

3.3. Manager

En este equipo se encuentra la aplicación *SkySimMonitoring*. Esta aplicación permite activar todo el software necesario para diseñar y ejecutar las simulaciones. Una vez iniciadas todas las aplicaciones los equipos quedan representados en verde como se puede observar en la figura 3.3. Si en algún momento algún equipo se apaga o se desconecta esta ventana avisará del problema.

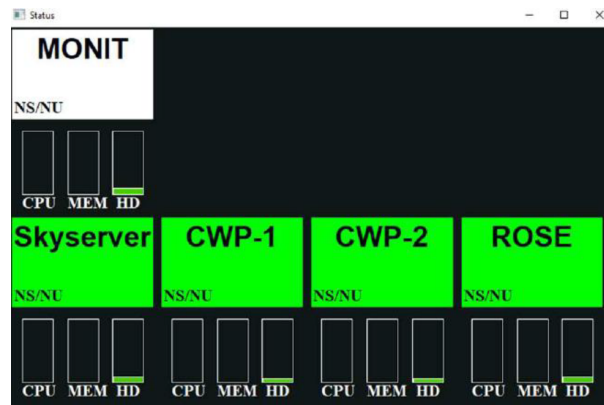


Figura 3.3: Ventana del manager de monitorización de todos los equipos

3.4. ROSE

ROSE son las siglas de *Radar Operation Simulator & Editor* y es una interfaz desarrollada por la empresa ROSE para el diseño de ejercicios de simulación para el entrenamiento teórico de los controladores ATC. El programa permite cargar un espacio aéreo y programar el tráfico aéreo que se encuentra en él de manera que se pueda simular muchos tipos de situaciones de tráfico. Además, cuenta con herramientas de diseño que permiten el cálculo de conflictos, lo que resulta muy útil en el diseño de los ejercicios.

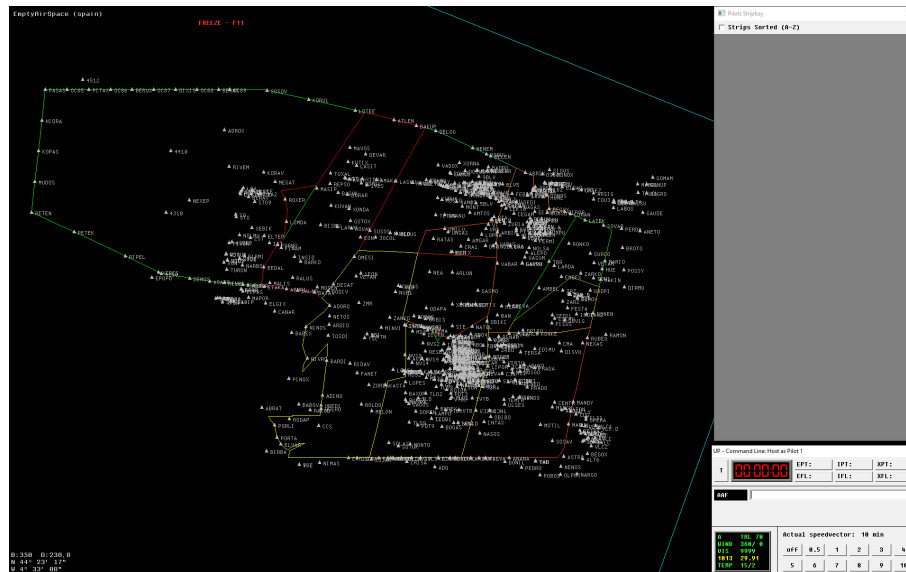


Figura 3.4: Interfaz de ROSE con el espacio aéreo del ACC de Madrid

La figura 3.4 muestra la pantalla principal de ROSE donde se puede ver a la izquierda el espacio aéreo que se ha cargado, arriba a la derecha la bahía de fichas de progresión de vuelo, donde se mostrarán las fichas de progresión de vuelo de todas las aeronaves involucradas en la simulación y abajo a la derecha la ventana de control maestro.

Es esta interfaz la que se ha utilizado para programar el ejercicio de validación como se explica en el capítulo 6 en más detalle.

3.5. SkySim

Los equipos SkySim 1 y 2 son los equipos de las posiciones ATCO 1 y 2 respectivamente. En estos equipos se tiene la interfaz *SkyVisual* que es la aplicación que permite realizar las simulaciones. En la figura 3.5 se puede ver la interfaz de *SkyVisual* de una posición ATC. En la parte superior de la figura se encuentra el panel de herramientas e información sobre el ejercicio de simulación.

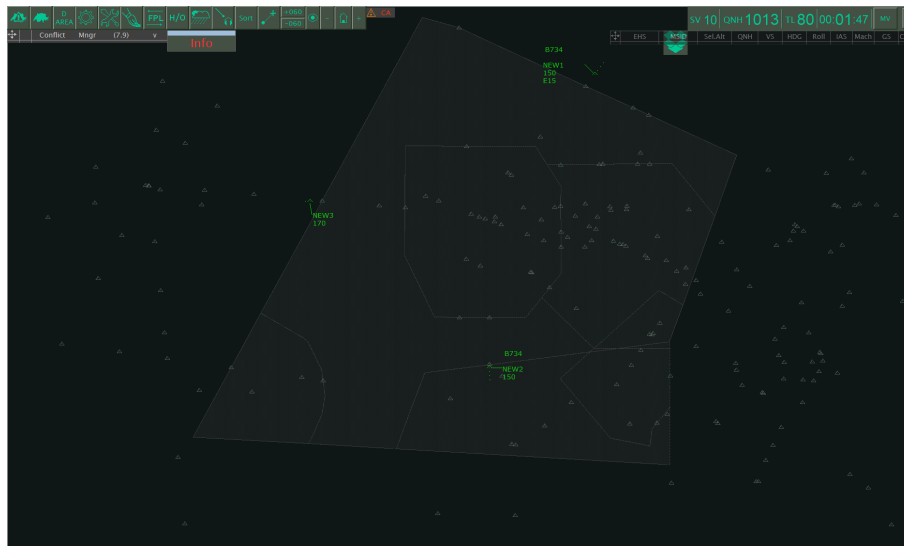


Figura 3.5: Interfaz de *SkyVisual* con los sectores BLU y BLL del ACC de Madrid asignados

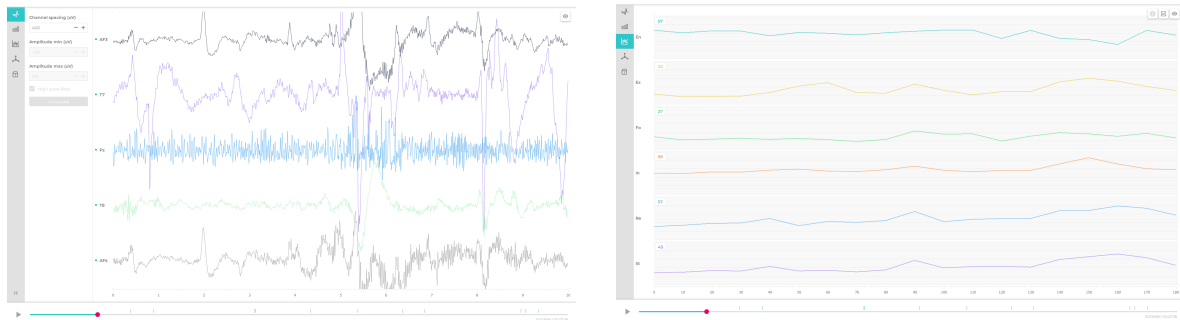
3.6. Equipos de medida

Los equipos de medida que se utilizan en las simulaciones son sistemas que miden factores ergonómicos de manera que a través de ellos se puede calcular la carga de trabajo asociada a los ejercicios de las simulaciones.

3.6.1. Emotiv INSIGHT y EmotivPRO

Diadema de medición de actividad cerebral compuesta por 5 electrodos de lectura de ondas cerebrales. Esta diadema ofrece la lectura directa de los datos en crudo correspondientes a las diferentes frecuencias según la onda cerebral medida y la región de interés [5].

En la figura 3.6 se puede ver a la izquierda la diadema utilizada y a la derecha el posicionamiento de los sensores en la cabeza que según el sistema 10-10. El posicionamiento de los sensores es el siguiente: AF3, AF4, T7, T8 y Pz. De acuerdo a este sistema Fp se refiere al pre-frontal, F al lóbulo frontal, P al lóbulo parietal, O al lóbulo occipital y T al lóbulo temporal y el resto de posiciones son posiciones intermedias entre estos lóbulos.



(a) Datos brutos de los sensores

(b) Parámetros medidos

Figura 3.7: Software Emotiv

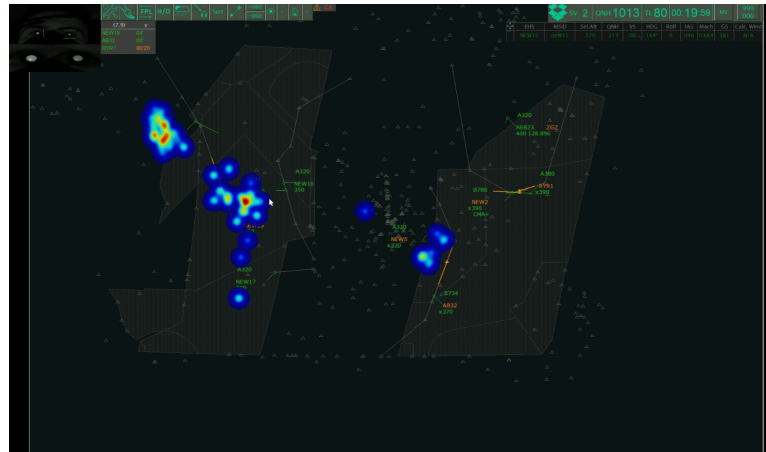
3.6.2. Eye tracker Gazepoint GP3 HD y Gazepoint Control and Gazepoint Analysis

Dispositivo de seguimiento visual encargado de la medición de parámetros oculomotrices relacionados con la atención. El seguimiento visual proporcionará los parámetros:

- Diámetro pupilar: relacionado con la concentración requerida para determinadas tareas.
- Número de fijaciones: entendido por fijación el acto de dirigir la mirada a un lugar particular.
- Duración de las fijaciones: relacionado con el parámetro anterior, refleja también el nivel de cambio de tareas.
- Frecuencia de parpadeo: depende también del grado de atención requerido por la tarea.
- Longitud sacádica: consiste en una medida que determina la amplitud del campo visual, a menores valores, la concentración del usuario es mayor y por ello se reduce su campo visual.



(a) Gazepoint GP3 HD



(b) Software Gazeplot Analysis

Figura 3.8: Eye Tracker

4. Diseño de nuevas funcionalidades

En esta sección se explica el proceso de diseño de las nuevas funcionalidades que se van a introducir en el simulador. Para ello, se han tenido en cuenta las especificaciones de las funcionalidades ER ATC 157 y CM-0211 del *ATM Masterplan* presentadas en el apartado 2.7 de este documento.

4.1. Requerimientos de diseño

El objetivo de la nueva funcionalidad es dar al controlador ATC la resolución óptima de los conflictos que aparecen durante el ejercicio de simulación. Para que esto sea posible, los requisitos de diseño de la nueva funcionalidad son los siguientes:

- Mostrar información adicional en pantalla en tiempo real. Esta información debe estar presentada de forma clara y concisa para evitar una sobrecarga de datos.
- Aparición de mensajes con la resolución óptima de conflictos en el momento que el sistema alerta del conflicto sin necesidad de pulsar un botón para mostrarlo.
- Permitir que el usuario tome la última decisión en la resolución de conflictos.

También se deben tener en cuenta los tres principios para el diseño centrado en el usuario (*user-centered*) descritos en [8]. De acuerdo a estos principios, la nueva tecnología debe:

1. Estar orientada a los objetivos, tareas y habilidades del usuario. Los nuevos sistemas no deben exceder las capacidades mentales, de percepción y físicas de los operadores. Además, los sistemas deben adaptarse a los objetivos dinámicos de los operadores si éstos trabajan en un entorno cambiante.
2. Organizarse en torno al camino que siguen los operadores para procesar la información y toman decisiones. En entornos complejos del mundo real, se dedica una cantidad considerable de tiempo a evaluar una situación y compararlo con información que se tiene en la memoria a largo plazo antes de elegir una acción. Por lo tanto, el nuevo sistema debe ayudar al usuario a mantenerse consciente de la situación.
3. Mantener al usuario consciente del estado del sistema. El sistema no puede poner al operador humano fuera del proceso de toma de decisiones (*out-of-the-loop*), pues de esta manera se perdería la conciencia de situación.

4.2. Definición de propuestas

A continuación se explican las posibles alternativas que se han estudiado para introducir la nueva funcionalidad. Como se quiere mostrar información adicional en pantalla, se ha valorado la posibilidad de utilizar las herramientas ya integradas dentro del software de *SkyVisual* y de ROSE o utilizar software externo al simulador. Finalmente, se definen cuatro propuestas diferentes:

1. Mensaje programado con ROSE

ROSE permite programar mensajes en el diseño de las simulaciones de manera que se muestren en pantalla en determinados momentos de la simulación. La herramienta que utiliza es el *Test Information Publisher* (TIP) de manera que aparezca un mensaje en la pantalla del controlador. En la figura 4.1 se muestra la ventana emergente donde aparecen los mensajes. Esta herramienta se ha descartado porque no aparecen los mensajes programados en la interfaz de *SkyVisual* durante el desempeño del ejercicio. Al no conocer la manera de configurar estos mensajes para el fin del proyecto no se puede usar esta herramienta en el simulador del laboratorio.



Figura 4.1: Mensaje programado con ROSE

2. Herramienta de edición de SkyVisual

Dentro del propio *SkyVisual* existe una herramienta de edición donde se puede mostrar texto, formas e iconos en la pantalla donde se realiza el ejercicio de simulación. La desventaja de esta herramienta es que no es posible programar los mensajes en el tiempo, es decir, si se quisiera usar esta herramienta la resolución de todos los conflictos del ejercicio aparecerían en pantalla desde el inicio de la simulación lo que supone una sobrecarga de datos en pantalla. En la figura 4.2 se muestran los controles de esta herramienta.



Figura 4.2: Herramienta de edición de SkyVisual

3. Chat entre puestos ATC

Esta herramienta permite que se escriban mensajes entre las dos posiciones ATC del laboratorio, de manera que si se conocen previamente las resoluciones óptimas de los conflictos, un operador externo puede proporcionar la información al controlador que esté realizando la simulación por medio del chat. La desventaja de este chat es la necesidad de una persona externa escribiendo los mensajes en el chat, pues estos mensajes no se pueden pre-programar, se trata de un chat en tiempo real. En la figura 4.3 se muestra el mensaje tal y como aparecería en la pantalla, primero aparece la ventana de Info que se ve en la parte superior izquierda de la figura, y al hacer click sobre esta ventana se despliega el mensaje. La necesidad de hacer click para mostrar el mensaje se ha identificado como otra desventaja de esta funcionalidad porque es posible que el controlador no vea la información hasta después de resolver el conflicto.

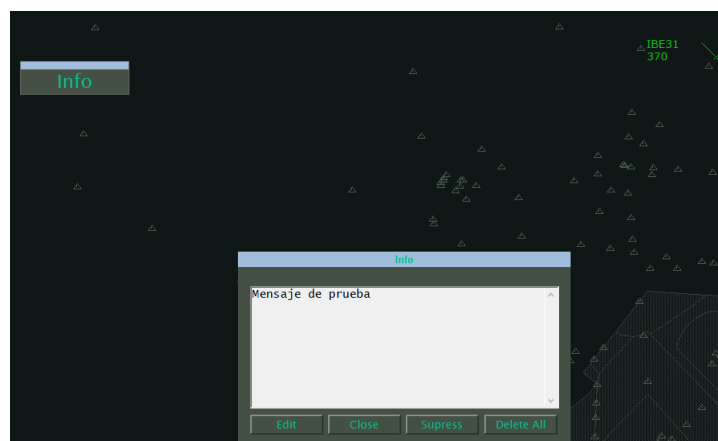


Figura 4.3: Mensajes en tiempo real entre puestos ATC

4. Aplicación externa en Python

Las tres propuestas anteriores utilizaban herramientas propias del software del simulador, y por lo tanto su diseño no está orientado a la nueva funcionalidad que se quiere introducir si no que la nueva funcionalidad se adapta a las herramientas pre-existentes. Debido a que son herramientas ya diseñadas para otras funcionalidades y a que se han encontrado diferentes limitaciones en cada una de ellas se ha propuesto el diseño de una nueva aplicación orientada a esta funcionalidad.

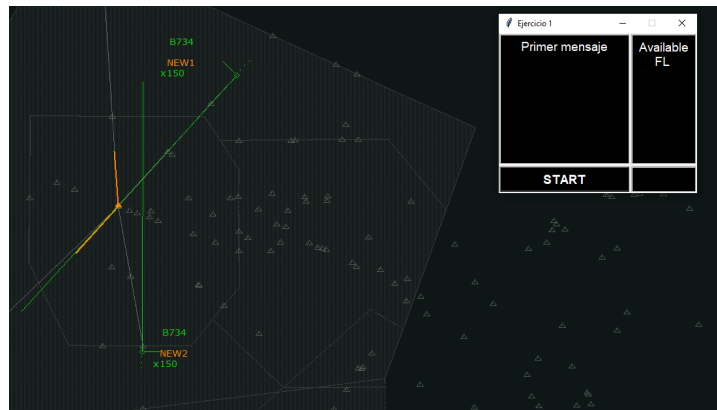


Figura 4.4: Ventana emergente programada en Python

Esta aplicación consiste en una ventana emergente en la pantalla del simulador donde se muestran mensajes programados como se muestra en la figura 4.4. De esta manera no se interfiere en la operación normal en el simulador, pero se tiene una ventana adicional fija en la pantalla donde se muestra información. Como se puede ver, esta pantalla se puede dividir en diferentes secciones y mostrar diferentes tipos de información como por ejemplo, los niveles de vuelo disponibles.

Por los motivos expuestos anteriormente en la definición de cada una de las propuestas, se ha elegido diseñar una nueva aplicación externa al software del simulador. La principal ventaja de esta propuesta es que permite un diseño flexible y adaptado a las necesidades del laboratorio de simulación, pues todas las demás herramientas integradas en el simulador no se pueden modificar desde el laboratorio. Esto significa que la nueva aplicación puede ser diseñada a medida, evitando las desventajas identificadas en el resto de propuestas. Además gracias a que es una aplicación externa al simulador se puede implementar en otros simuladores de posición ATC, es decir, se trata de una solución interoperable, que es otra gran ventaja respecto al resto de propuestas. En lo que sigue se va detallar el proceso de diseño de la aplicación.

4.3. Diseño de la solución final

La figura 4.5 presenta el diseño final de la aplicación. A continuación se explica el proceso que se ha seguido para su creación, que incluye la selección del lenguaje de programación y la programación en sí misma, una descripción del funcionamiento de la aplicación y los pasos a seguir para su implementación en un equipo y finalmente una guía de las líneas de código a modificar para introducir los mensajes de resolución de conflicto.



Figura 4.5: Diseño final de la aplicación

4.3.1. Selección del lenguaje de programación

El punto de partida del diseño de la aplicación es la selección del lenguaje de programación que se va a utilizar. Se ha elegido la programación en Python por varias razones:

1. Se trata de un software libre que no requiere licencias de pago.
2. Es un lenguaje orientado a objetos, lo que permite mayor facilidad en el diseño de aplicaciones.
3. Este lenguaje simplifica mucho la programación de manera que cualquier modificación es rápida de llevar a cabo.
4. Es flexible, ordenado y limpio.

Por contra, la principal desventaja de Python con otros lenguajes de programación como Matlab y c#.net es que no existe una interfaz de diseño de aplicaciones predefinida, sin embargo al ser necesaria una licencia para su uso se ha descartado la utilización de estos lenguajes en este proyecto. Esto significa que la adición de esta nueva funcionalidad no tiene un coste adicional al simulador.

4.3.2. Programación de la aplicación

Una vez escogido el lenguaje, se procede a la programación de la aplicación. Para ello, se ha utilizado la interfaz de Visual Studio Code debido a que incluye una extensión de Python que permite la visualización de errores y sigue una lógica de colores que facilita la programación. El código completo se encuentra en el Anexo A.

En el programa se distinguen los siguientes elementos:

- **Librerías:** son un conjunto de subprogramas utilizados para desarrollar software. Las librerías contienen código y datos que pueden utilizar programas independientes, es decir, pasan a formar parte de estos.
- **Variables globales:** son variables de alcance global, por lo que se puede acceder a ellas desde cualquier nivel dentro del programa.
- **Clases:** es una plantilla para la creación de objetos de datos según un modelo predefinido, es decir, da una estructura para manejar los objetos en su interior.
- **Funciones:** también llamadas métodos, permiten dividir el trabajo que hace un programa en tareas más pequeñas separadas de la parte principal.

Librerías

Las librerías que se han utilizado se presentan en la siguiente tabla:

Librería	Funcionalidad
tkinter	Utilización de interfaces gráficas
time	Manejar variables en el tiempo
threading	Gestión de operaciones concurrentes (varios hilos de ejecución)
queue	Implementación de colas a lo largo del programa
datetime	Capturar la fecha y la hora
canvas	Creación de reportes en pdf
os	Creación de archivos .txt

Tabla 4.1: Función de las librerías utilizadas

Variables globales

Se han usado dos variables globales: **cont** y **time**. La primera de ellas es una variable numérica que actúa como contador de clicks de los botones. La variable **time** es un vector que almacena la información de los botones.

Clases

El funcionamiento del programa incluye dos ventanas emergentes: la aplicación principal y un diálogo de salida en forma de formulario. Cada una de estas ventanas constituye una clase, el nombre de estas clases dentro del programa es **Ventana_principal** y **My_dialog** respectivamente.

Funciones

El programa utiliza las funciones que se presentan a continuación. Dentro del programa estas funciones se introducen con la palabra **def** por lo que es sencillo localizarlas.

- **__init__**: inicia las ventanas y la estructura de las mismas.
- **mensajes**: gestiona la cola de mensajes de texto y activa y desactiva los botones.
- **CONFLICT**, **NOCONFLICT**, **YES** y **NO**: escribe en el vector **time** cuando se pulsa un botón almacenando como texto el nombre del botón y la hora en formato hh:mm:ss.
- **update_status**: actualiza la cola de mensajes.
- **create_widgets**: crea los elementos de la ventana, que son los paneles y los botones.
- **on_closing**: llama a la clase **My_dialog** que abre la ventana de salida de la aplicación.
- **salir**: cierra la aplicación, escribe el vector **time** en formato .pdf y .text y guarda los archivos.

4.3.3. Descripción de la aplicación

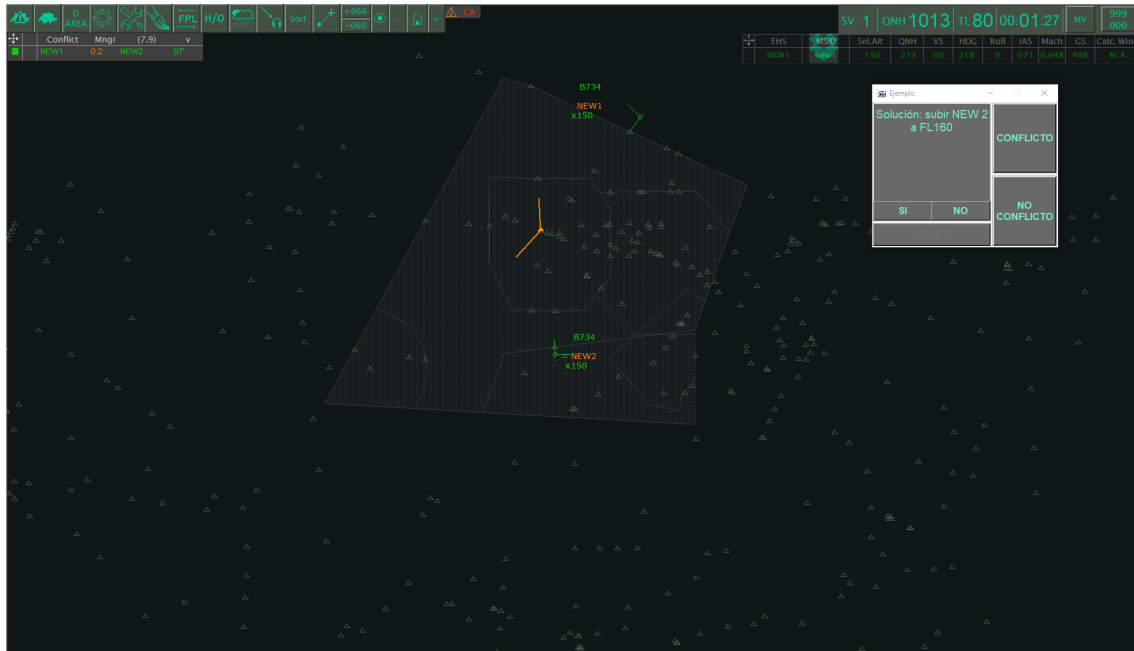


Figura 4.6: Ejemplo de la nueva funcionalidad en el simulador

En la figura 4.6 se puede observar el diseño final de la aplicación. Esta pantalla tiene un tamaño fijo de 310x240 printer points, lo que equivale a 11.5x8.5 cm aproximadamente. Esto supone menos de un 4% del área de la pantalla del simulador, pero permite que el tamaño de la fuente sea de 13pt para que se puedan leer los mensajes con facilidad.

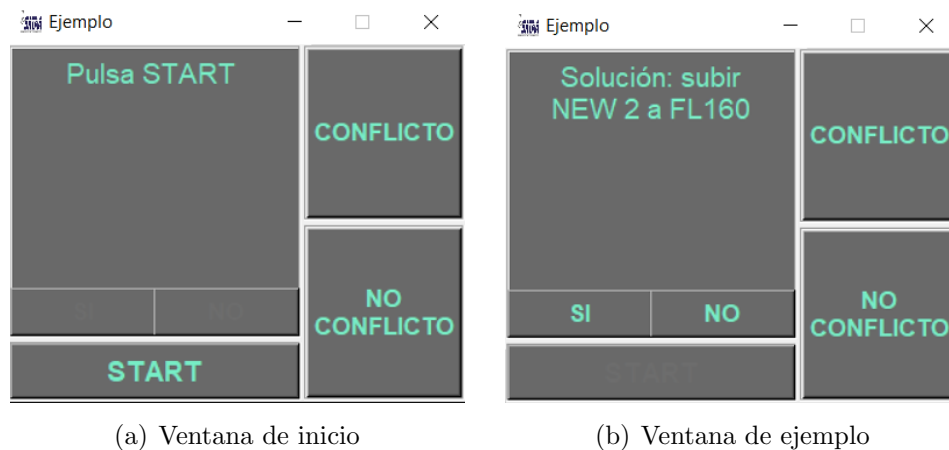


Figura 4.7: Interfaz de la nueva funcionalidad

La ventana emergente está formada por un espacio que contiene el texto de los mensajes programados en la parte superior izquierda y cinco botones diferentes. Cada vez que se hace click sobre un botón queda registrado el botón y la hora a la que se ha presionado para poder analizar los resultados posteriormente. En la figura 4.7 se muestra la interfaz de la ventana antes de iniciar la simulación y una vez iniciada la simulación dando la resolución de un conflicto. Los botones de la nueva funcionalidad son los siguientes:

- **START:** sirve para iniciar la simulación en la ventana para que empiece a contar el tiempo en el reloj de los mensajes programados. Este botón hay que presionarlo a la vez que el botón de iniciar la simulación en ROSE.
- **SI/NO:** estos botones se deben utilizar tras la resolución de un conflicto. Si se ha realizado la resolución propuesta en el mensaje programado se pulsa SI y si se ha solucionado de otra manera se pulsa NO.
- **CONFLICTO:** este botón se debe pulsar si aparece en la simulación un conflicto sin solución.
- **NO CONFLICTO:** este botón se debe pulsar cuando aparece la solución óptima de un conflicto pero el conflicto no se produce.

La necesidad de los botones CONFLICTO y NO CONFLICTO se justifica debido a que una vez iniciada la simulación el controlador realiza cambios en las trayectorias de las aeronaves que no se pueden anticipar a priori, de manera que se producirán situaciones en las que se eviten conflictos y situaciones en las que se generen conflictos no previstos y no se haya programado una solución óptima para ellos. Este problema se resuelve con herramientas de resolución de conflictos automatizadas (por ejemplo inteligencia artificial y machine learning) que quedan fuera del alcance de este proyecto.

Atendiendo a los cuatro botones que el controlador debe pulsar durante el ejercicio de simulación pueden encontrarse las cuatro situaciones resumidas en la figura 4.8.

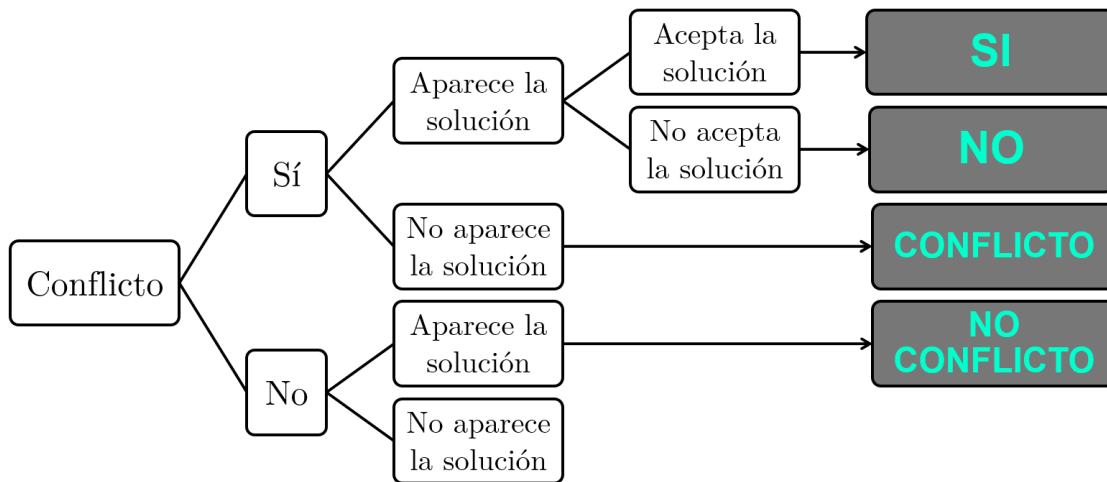




Figura 4.8: Esquema de las posibles situaciones durante la simulación

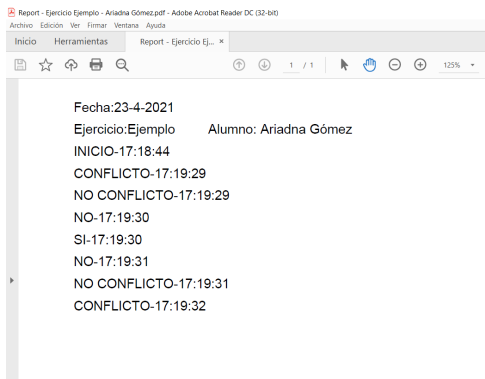
Para el posterior análisis de los datos, al cerrar la ventana emergente aparece un diálogo como el que se muestra en la figura 4.9, en la que se pide al sujeto introducir el número de ejercicio dentro del programa y su nombre y apellido. Al hacer click en "Guardar y salir" de forma automática se guarda un informe en formato .pdf y otro en formato .txt para la posterior explotación de los datos. Estos informes se guardan en una carpeta definida en el programa y su formato es el que se muestra en la figura 4.10.

Figura 4.9: Ventana emergente para salir del ejercicio de simulación

Gracias a la información de estos informes se puede recalificar la carga de trabajo de los ejercicios de simulación para tener una mayor comprensión de los datos obtenidos de los sensores de EGG y del Eye Tracker.

<input type="checkbox"/> Nombre	Fecha de modificación	Tipo	Tamaño
 Report - Ejercicio Ejemplo - Ariadna Gómez.pdf	23/04/2021 17:20	Adobe Acrobat D...	2 KB
 Report - Ejercicio Ejemplo - Ariadna Gómez.txt	23/04/2021 17:20	Documento de tex...	1 KB

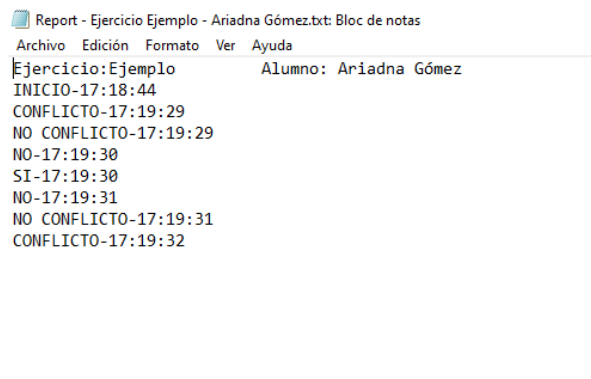
(a) Informes guardados en la carpeta



Report - Ejercicio Ejemplo - Ariadna Gómez.pdf - Adobe Acrobat Reader DC (32-bit)

Inicio Herramientas Report - Ejercicio Ej...

Fecha:23-4-2021
Ejercicio:Ejemplo Alumno: Ariadna Gómez
INICIO-17:18:44
CONFLICTO-17:19:29
NO CONFLICTO-17:19:29
NO-17:19:30
SI-17:19:30
NO-17:19:31
NO CONFLICTO-17:19:31
CONFLICTO-17:19:32



Report - Ejercicio Ejemplo - Ariadna Gómez.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Ejercicio:Ejemplo Alumno: Ariadna Gómez
INICIO-17:18:44
CONFLICTO-17:19:29
NO CONFLICTO-17:19:29
NO-17:19:30
SI-17:19:30
NO-17:19:31
NO CONFLICTO-17:19:31
CONFLICTO-17:19:32

(b) Informe en formato .pdf

(c) Informe en formato .txt

Figura 4.10: Exportación de los datos de la simulación recogidos por los botones

4.3.4. Implementación de la nueva funcionalidad

Para la implementación de esta aplicación únicamente se deben seguir 4 pasos que se indican a continuación. Cabe señalar que la implementación de esta aplicación solamente se ha realizado en Windows, pudiendo haber diferencias en otros sistemas operativos.

■ PASO 1: Instalar el compilador de Python



Primero se requiere instalar el compilador de Python disponible de forma gratuita en [34]. Como el ordenador del simulador tiene Windows, tan solo hay que descargar la versión para Windows e instalarla. En la figura 4.11 se puede ver el botón de descarga en la página web.



Figura 4.11: Página web para descargar el compilador de Python [34]

■ PASO 2: Añadir la librería *reportLab*

La aplicación de la nueva funcionalidad necesita la librería *reportLab* para la generación del informe en formato pdf. Para añadir esta librería se deben seguir estos pasos:

1. Pulsar  +  para abrir el cuadro de diálogo de ejecutar de Windows.
2. Escribir «*cmd*» para abrir la ventana de comandos.
3. Una vez en la ventana de comandos escribir en ella «*pip install reportLab*» y pulsar Enter. Así se iniciará la instalación de esta librería de forma automática. En caso de que no se inicie de forma automática se debe descargar primero las librerías de <https://pypi.org/project/reportlab/#files> y a continuación escribir el comando.

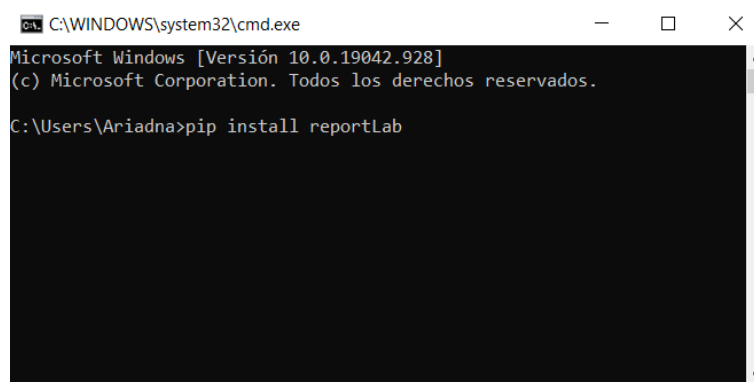
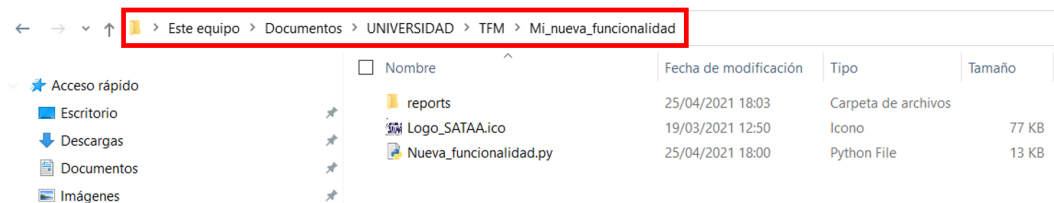


Figura 4.12: Comando de instalación de la librería *reportLab* en la ventana de comandos

Este paso se puede saltar si no se desea el informe en formato .pdf, tan solo hay que comentar las líneas del código que generan el informe.

- **PASO 3: Descomprimir la carpeta donde se tiene el código y cambiar las ubicaciones de guardado**

Primero se descomprime la carpeta donde se encuentra el código y a continuación se deben cambiar las ubicaciones de los archivos en el código. Para ello se abre el código con cualquier editor de texto (por ejemplo el bloc de notas) y se busca: «iconbitmap», «open» y «c=» con la herramienta de búsqueda. Con estas búsquedas aparecen 4 resultados: 2 del icono de la aplicación y 2 de los informes de la simulación. Se debe cambiar la ubicación que aparece escrita en esas líneas por la ubicación donde se encuentre la carpeta en el ordenador donde se vaya a utilizar. En la figura 4.13(a) se muestra dónde encontrar la dirección y en la 4.13(b) está marcada la ubicación que se debe modificar. Al copiar la ubicación desde el explorador de archivos se copia con una separación de un único que se debe duplicar para que funcione.



(a) Ubicación de la carpeta

```
261 #Declara que la función superior es la ventana principal
262 if __name__ == "__main__":
263     root = tk.Tk()
264     root.resizable(width=False, height=False) #Fija el tamaño de la ventana
265     root.wm_attributes("-topmost", True) # Esta línea es importante: fija la ventana y se puede trabajar por debajo
266     root.iconbitmap('c:\Users\Ariadna\Documents\UNIVERSIDAD\TFM\Mi_nueva_funcionalidad\Logo_SATAA.ico')
267     app = Ventana_principal(master=root)
268     app.mainloop()
```

(b) Línea del código que debe modificarse

Figura 4.13: Cambiar la ubicación de los archivos

- **PASO 4: Ejecutar el programa con el compilador de Python**

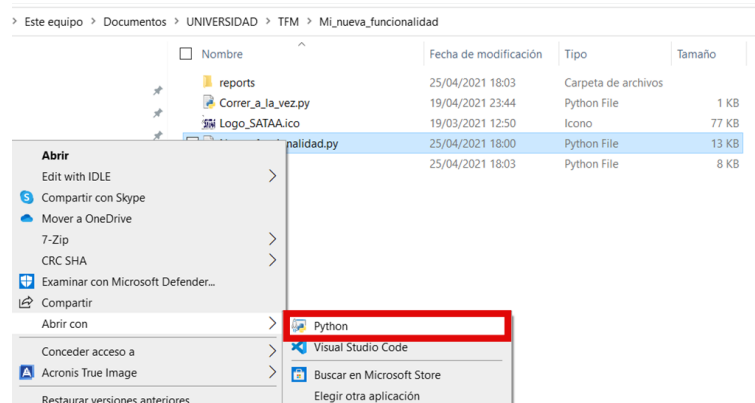


Figura 4.14: Cómo ejecutar el programa con el compilador de Python

4.4. Programación de los mensajes en una simulación

Aunque el código completo comentado de la nueva funcionalidad se encuentra en el Anexo A y ya se ha explicado la estructura del código, en el presente apartado se explica cómo programar los mensajes para que aparezcan a la vez que aparece un conflicto. Antes de la programación de estos mensajes el ejercicio de simulación debe estar programado, pues los mensajes se programarán a partir de la tabla de eventos del ejercicio.

Los mensajes se programan en la función «mensajes» que comienza en la línea 41 del código. En la figura 4.15 se muestra esta función en la que distinguimos diferentes comandos:

- `cola.put(" ")`: en el interior de las comillas se escribe el texto que se desea mostrar.
- `time.sleep()`: en el interior del paréntesis se introduce el intervalo de tiempo entre mensajes. En el ejemplo de la figura 4.15 aparece el primer mensaje 4 segundos después de pulsar START, se mantiene 10 segundos y después de otros 4 segundos aparece el siguiente mensaje y se mantiene en el tiempo.
- `self.YES_button['state'] = tk.NORMAL`: estas funciones sirven para habilitar o deshabilitar los botones de SI y NO. En el ejemplo, estos botones se activan cuando hay mensaje en la pantalla y desactivan cuando no.

```
40     @staticmethod
41     def mensajes(cola, self): #Esta función me permite programar los mensajes en el tiempo
42         cola.put(" ")
43         time.sleep(4)
44         cola.put("Solución: subir NEW 2 a FL160")
45         self.YES_button['state'] = tk.NORMAL
46         self.NO_button['state'] = tk.NORMAL
47         time.sleep(10)
48         cola.put(" ")
49         self.YES_button['state'] = tk.DISABLED
50         self.NO_button['state'] = tk.DISABLED
51         time.sleep(4)
52         cola.put("Solución: desviar NEW 3 por ...")
53         self.YES_button['state'] = tk.NORMAL
54         self.NO_button['state'] = tk.NORMAL
55
```

Figura 4.15: Función para programar mensajes

Una vez se han programado los mensajes siguiendo la tabla de eventos del ejercicio ya se puede ejecutar el programa.



UNIVERSIDAD
POLITÉCNICA
DE MADRID



5. Procedimiento de validación

En este capítulo se propone un procedimiento de validación de la nueva funcionalidad introducida en el simulador a través de la medida de factores ergonómicos.

5.1. Justificación del estudio

El objetivo del estudio es comprobar el efecto en la carga de trabajo de los controladores aéreos de las nuevas funcionalidades propuestas para el simulador del laboratorio. Como ya se ha mencionado anteriormente, la introducción de nuevas funcionalidades o la mejora de los sistemas ATC tienen un impacto directo en la carga de trabajo de los controladores. Por ello, el objetivo de este estudio es la implementación de nuevas funcionalidades en el simulador para obtener datos de los sensores y medidas subjetivas de los sujetos estudiados para poder emitir una conclusión final que pueda llevar a la implementación de funcionalidades parecidas en entorno real.

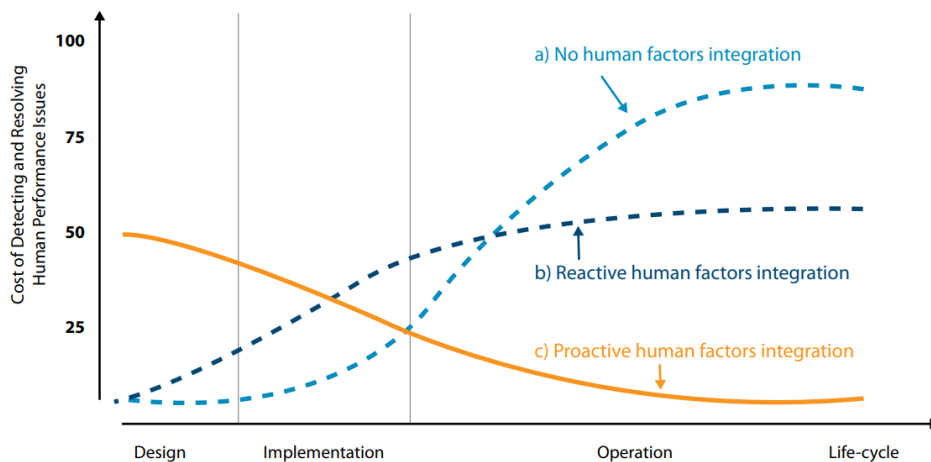


Figura 5.1: Tres escenarios de costes diferentes en la implementación de una nueva funcionalidad en función de la estrategia de integración de los factores humanos en el proceso. Fuente: EURO-CONTROL [12]

En la figura 5.1 se representan los costes de detección y resolución de incidencias relacionados con los factores humanos en las diferentes fases que se siguen para el introducir nuevas funcionalidades (diseño, implementación y operación). Se puede ver que si los factores humanos se tienen en cuenta desde la fase de diseño, el coste de detección y resolución de problemas es mucho menor que si no se tuvieron en cuenta y además estos problemas se detectarían en fases iniciales y no cuando el proyecto está más avanzado, por lo que no sólo

se disminuye el coste de detección y resolución de problemas relacionados con los factores humanos, también se disminuyen los costes de diseño, pues si estos problemas se detectan en fases más avanzadas se debe volver atrás en el proyecto, dicho con otras palabras, gracias a la integración de los factores humanos en el proceso de diseño (*user-in-the-loop*), se pueden anticipar dificultades antes de la implementación de funcionalidades, lo que se traduce en una disminución de los costes.

Además de tener en cuenta los costes representados en la figura, se debe tener en cuenta que en el caso particular de introducir una nueva funcionalidad en el sistema ATC el efecto en la carga de trabajo del controlador repercute de forma inmediata en la seguridad de las operaciones y por lo tanto se debe garantizar que la nueva funcionalidad está adaptada a las necesidades del usuario.

5.2. Objetivos e hipótesis

Antes de enunciar las hipótesis del estudio es necesario definir el concepto carga de trabajo. La carga de trabajo se puede definir como la medida de la demanda de procesos mentales que tiene una persona durante el desempeño de una tarea [4]. Es decir, son los recursos mentales o capacidades que son necesarios para completar una tarea satisfactoriamente [29]. Los procesos mentales o capacidades se refieren al procesamiento de la información. Los estímulos deben ser percibidos y traducidos a una respuesta. La memoria cognitiva y la atención cognitiva afectan a las diferentes etapas del procesamiento de la información. Estas etapas son las que se muestran en la figura 5.2.

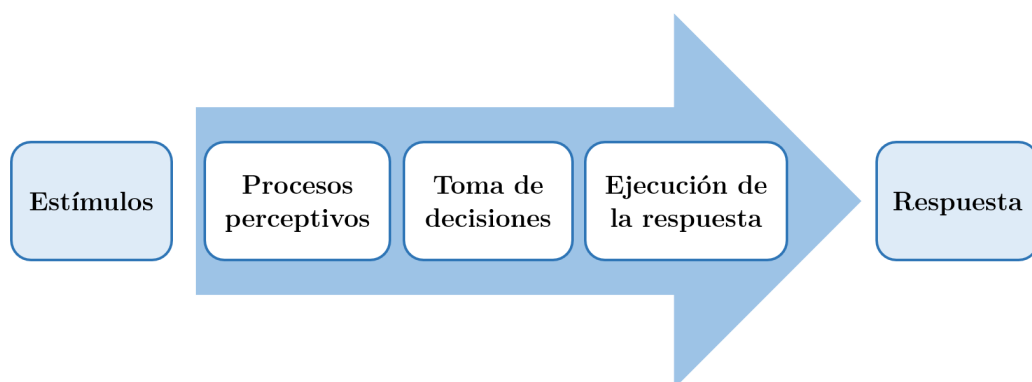


Figura 5.2: Etapas del procesamiento clásico de la información

La razón por la que se mide la carga de trabajo es para cuantificar el coste mental que tiene la ejecución de tareas para predecir el rendimiento del operador y del sistema. De esta forma, la medida de la carga de trabajo proporciona una predicción de qué demanda de

tareas conduce a un rendimiento inaceptable.

Atendiendo a estas definiciones, el objetivo general de este estudio es demostrar que introduciendo la nueva funcionalidad de dar resolución óptima de conflictos se disminuye la carga de trabajo del controlador.

Los resultados esperados con la introducción de la nueva funcionalidad son los correspondientes al escenario normal de la figura 5.3, es decir, tener una disminución de la carga de trabajo del controlador con la introducción de la nueva funcionalidad. Además, se pueden dar otros dos escenarios, uno favorable donde la disminución de la carga de trabajo es muy notable y otro desfavorable donde la carga de trabajo aumente en vez de disminuir.

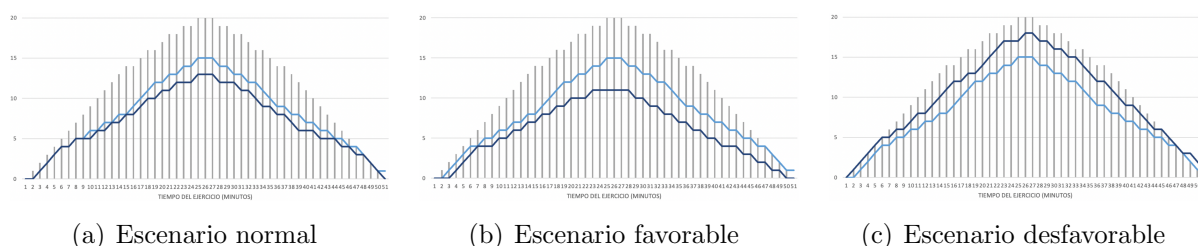


Figura 5.3: Escenarios posibles de respuesta. En gris se representa la carga de trabajo de diseño, en azul claro la respuesta del sujeto sin funcionalidades nuevas y en azul oscuro la respuesta del sujeto con funcionalidades nuevas.

Las hipótesis que se quieren demostrar con el estudio son:

- **Hipótesis 1:** Cuando se cambia la estrategia a la hora de desempeñar una tarea hay un efecto en la carga de trabajo. La introducción de la nueva funcionalidad supone un cambio en la toma de decisiones del controlador, pues se proporciona una solución a conflictos que se debe valorar y aplicar o descartar.
- **Hipótesis 2:** Cuando se proporciona la solución óptima de un conflicto la carga de trabajo asociada a la tarea de resolver el conflicto disminuye.
- **Hipótesis 3:** Cuando disminuye la carga de trabajo, el controlador mejora su rendimiento y realiza sus tareas con mayor antelación, es decir, no sólo resuelve los conflictos de forma óptima si no que mejora el desempeño de otras tareas como la asunción y transferencia de aeronaves.

5.3. Protocolo experimental

Este estudio se ha diseñado siguiendo los protocolos experimentales analizados en [24] debido a que en esta fuente se revisan experimentos relacionados con la medida de la carga de trabajo en aviación.

5.3.1. Participantes

Los participantes de este estudio serán un conjunto de 10 estudiantes de tercer y cuarto curso del Grado en Gestión y Operaciones del Transporte Aéreo (GyOTA) de la Universidad Politécnica de Madrid. La selección de los participantes se ha realizado de acuerdo a su nivel de experiencia con el simulador. Estos estudiantes al finalizar su formación obtienen la licencia de estudiante controlador *Student ATCO License*.

El criterio de selección es que hayan completado la fase básica del programa de ejercicios del simulador del Laboratorio UPM. Se trata por tanto de una muestra incidental, es decir, los participantes han sido seleccionados previamente a la realización del estudio y será una muestra heterogénea formada idealmente por 5 hombres y 5 mujeres. El rango de edad de todos los participantes está entre los 18 y los 22 años y fuera de este rango quedan excluidos.

5.3.2. Entorno de evaluación

El entorno de evaluación del estudio consiste en la realización de un ejercicio de simulación de 60 minutos de duración. Para comprobar el impacto de la nueva funcionalidad el ejercicio éste se realizará dos veces, una de ellas sin la nueva funcionalidad y otra con la nueva funcionalidad. Requerimientos del plan de ejercicios:

1. La carga de trabajo estimada del ejercicio debe ser elevada para obtener niveles de arousal (niveles de excitación cortical, alerta y estrés) correspondientes a un rendimiento alto según la ley de Yerkes-Dodson. En la figura 5.4 se representa esta ley, que relaciona el nivel de arousal con el rendimiento cuando se realiza una tarea. El caso de controlar el tráfico aéreo corresponde a una tarea difícil por lo que exceder el nivel de arousal causa una disminución del rendimiento. En el siguiente capítulo se explica cómo se estima la carga de trabajo del ejercicio.

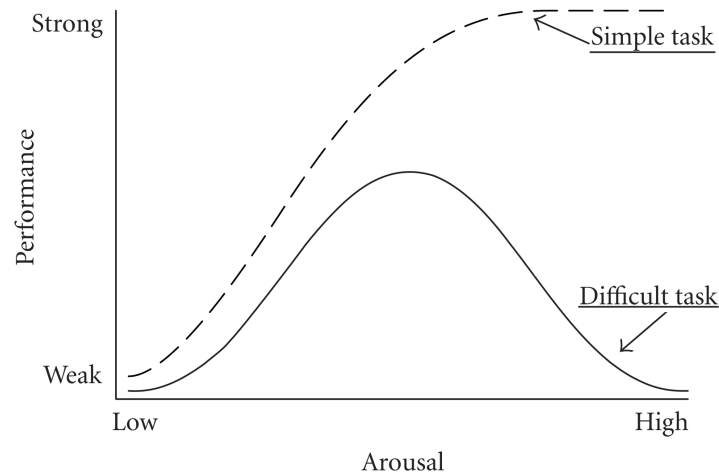


Figura 5.4: Ley de Yerkes-Dodson. Fuente: [29]

2. La realización de un mismo ejercicio con y sin la asistencia en la resolución de conflictos deben estar suficientemente separadas temporalmente para que los participantes no se acuerden del tráfico aéreo del ejercicio. Por esta misma razón, la duración del ejercicio debe ser prolongada en el tiempo por eso se ha elegido una duración de 60 minutos.
3. Durante la realización de los ejercicios los participantes no deben ser distraídos ni deben realizar otras tareas ajenas a la simulación para que las medidas no se vean afectadas.

5.3.3. Instrumentos de medida

La medición de la carga de trabajo sirve para cuantificar el coste mental de realizar una tarea como se anticipó en el apartado 5.2. Las medidas de carga de trabajo se clasifican en tres grupos: psicofisiológicas, subjetivas y de rendimiento [30].

- Medidas psicofisiológicas: son medidas objetivas de las reacciones físicas del cuerpo frente a una determinada carga de trabajo mental. Esta medida se puede considerar la más exacta porque no requiere una respuesta inmediata de los sujetos como en las medidas subjetivas, sin embargo, el cuerpo también responde a otras cosas además de a la carga mental y los resultados pueden verse afectados. Son variables psicofisiológicas el latido del corazón, la frecuencia de parpadeo y los EEG entre muchas otras.
- Medidas subjetivas: las medidas subjetivas reflejan la percepción de carga de trabajo de los sujetos. Se tratan de preguntas tipo *scaling-rate* donde los sujetos deben dar una puntuación dentro de una escala como respuesta a las preguntas. Estas medidas

son muy prácticas y se ha demostrado que están directamente relacionadas con las variables psicofisiológicas, sin embargo, la respuesta de los diferentes sujetos depende de su actitud y de sus habilidades. Cuestionarios para medir la carga de trabajo de esta manera son por ejemplo: el NASA TLX (figura 5.5, el SWAT (Subjective Workload Assessment Technique) y el Instantaneous Self-Assessment (ISA).

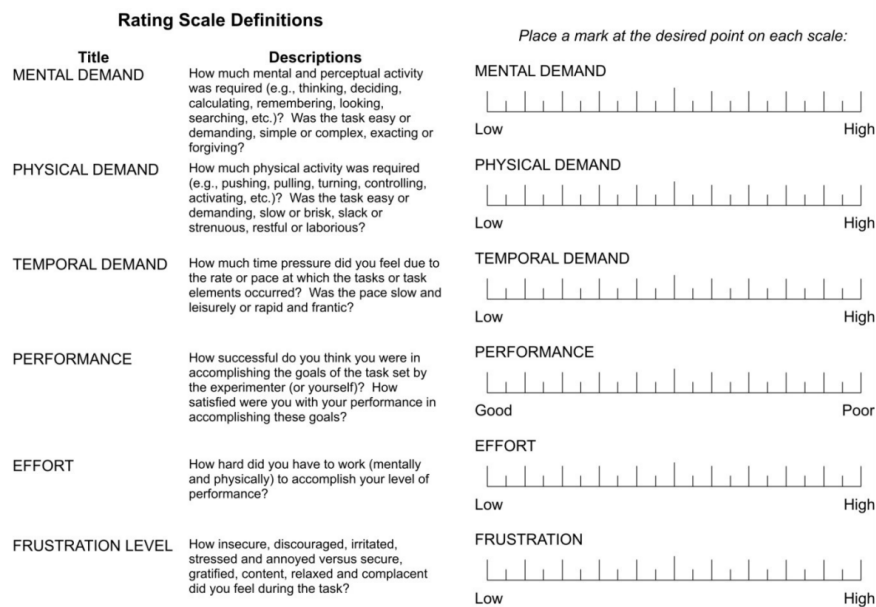


Figura 5.5: NASA TLX

- Medidas de rendimiento: estas medidas están relacionadas con el rendimiento o la efectividad para resolver tareas. Existen dos maneras de medir la carga de trabajo, a través de tareas primarias y secundarias. Estas medidas de la carga de trabajo son las más imprecisas pero son adecuadas para determinar si un sujeto tiene la capacidad de realizar determinadas tareas o no. En este caso esta medida se va a tomar con el registro temporal de conflictos y la efectividad de su resolución.

La literatura demuestra que, existen numerosos métodos para medir la carga de trabajo y no está consensuado una metodología específica para definirla. Para la realización de los ejercicios de simulación se van a monitorizar:

1. Variables psicofisiológicas con la diadema de electroencefalografía EMOTIV Insight y con el dispositivo de seguimiento visual Gazepoint GP3 HD presentados en el apartado 3.6.
2. Variables subjetivas a través del cuestionario NASA TLX.

3. Variables del rendimiento del ejercicio de simulación con el registro de conflictos proporcionado por la aplicación.

5.4. Limitaciones del estudio

1. Los sujetos son alumnos y no controladores aéreos. Esto quiere decir que los alumnos no han completado su formación y algunas variables pueden verse afectadas por el aprendizaje.
2. Número limitado de participantes disponibles para realizar el estudio. Los resultados de este estudio no son generalizables para todos los controladores aéreos de España porque la muestra no es representativa, y tampoco es representativa de todos los estudiantes controladores porque los participantes son de un mismo centro y programa de formación.
3. Predictibilidad e incertidumbre: no se puede predecir qué es lo que pasa si el sujeto modifica las trayectorias, de manera que la nueva funcionalidad no proporciona la solución óptima del conflicto, por lo tanto no todos los ejercicios realizados serán idénticos para todos los sujetos. Para ajustar la carga de trabajo del ejercicio se dispone de la información de la nueva funcionalidad pero esto supone pérdida de generalidad al compararlo con el ejercicio diseñado.



UNIVERSIDAD
POLITÉCNICA
DE MADRID



6. Diseño del ejercicio de validación

6.1. Elementos característicos de los ejercicios

En este apartado se van a explicar los elementos característicos de los ejercicios de simulación. Se quiere señalar que tan sólo se mencionan aquellos elementos que son de interés para los ejercicios diseñados, ya que ROSE tiene muchas posibilidades de diseño que no se van a utilizar. Por lo tanto, los elementos o acciones características de estos ejercicios son:

- Asunción de un vuelo: consiste en asumir la responsabilidad de una aeronave de otro controlador. Antes de asumir un vuelo, el controlador debe comprobar que su trayectoria entre en su sector asignado. Para facilitar esta tarea, SkyVisual está configurado para que el icono del *callsign* de las aeronaves sea diferente en función de si la aeronave es de responsabilidad del controlador en esa posición o no, aunque sí que permite que se asuman aeronaves fuera de responsabilidad. La figura 6.1 muestra los diferentes iconos.

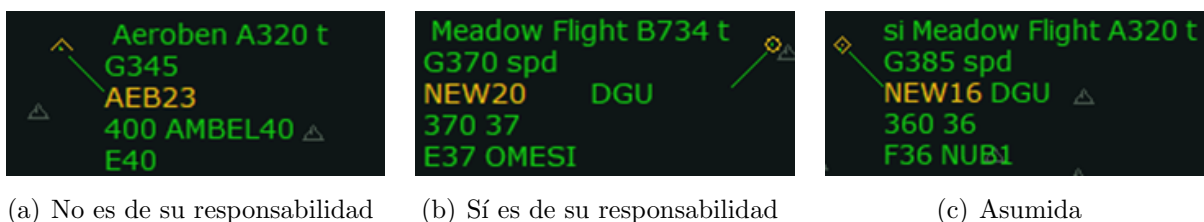


Figura 6.1: Iconos del *callsign* de las aeronaves en función de la responsabilidad de los controladores.

- Monitorización de un vuelo: consiste en el seguimiento del vuelo dentro del sector de responsabilidad. Dentro de este elemento se incluye comprobar que las aeronaves están siguiendo el plan de vuelo establecido.
- Transferencia de un vuelo: consiste en transferir la responsabilidad de una aeronave a otro controlador. La transferencia a otro sector se realiza cuando la aeronave ha abandonado o está abandonando el sector de responsabilidad y ya no va a entrar en conflicto con otra aeronave dentro del sector de responsabilidad.
- Resolución de conflictos: un conflicto en ruta ocurre cuando las aeronaves infringen la separación mínima. Para evitar que el conflicto ocurra existen cuatro maneras de solucionarlo en SkyVisual:

- *Re-routing*: consiste en ordenar el cambio de la trayectoria de una de las dos aeronaves implicadas en el conflicto para que, posteriormente a la evasión del conflicto, se reincorporen a la trayectoria planeada de la que se disponía inicialmente.

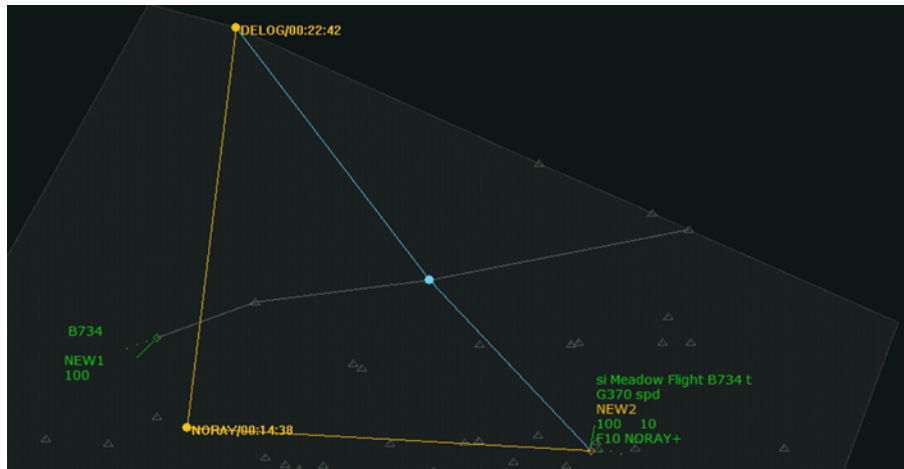


Figura 6.2: Resolución de un conflicto por re-routing.

- Cambio de nivel de vuelo: desde el punto de vista técnico se trata de la acción más simple y más fácil de calcular.
- Cambio de velocidad: consiste en modificar la velocidad de una o de las dos aeronaves para evitar el conflicto. Esta alternativa no supone un cambio en el posicionamiento de las aeronaves.
- Directo a un punto: consiste en dar a una aeronave la instrucción de omitir uno o varios puntos intermedios de su trayectoria y volar directamente a un punto.

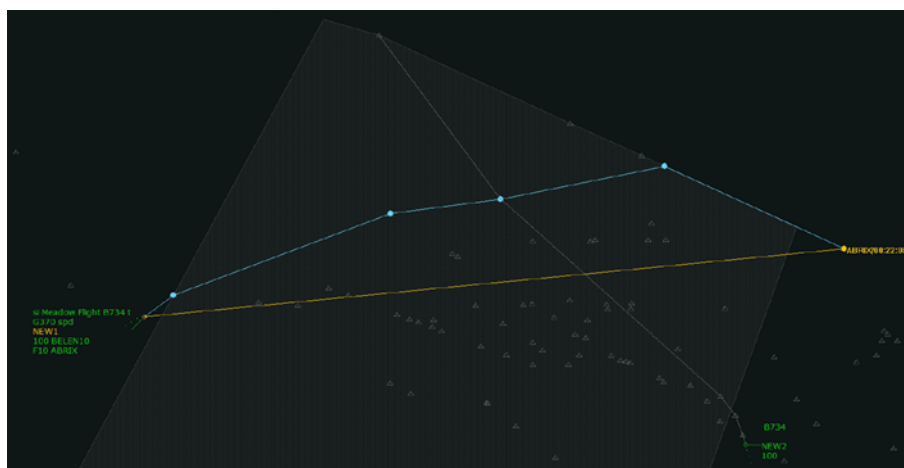


Figura 6.3: Resolución de un conflicto por directo a un punto.

6.2. Procedimiento de estimación de carga de trabajo

Para el diseño de los ejercicios es necesario asignar valores cuantitativos a las diferentes acciones que se llevan a cabo a lo largo de la simulación. Gracias a estos valores cuantitativos se puede conocer si una situación de tráfico conlleva más o menos carga de trabajo. Existen muchos elementos que añaden carga de trabajo a los controladores, como el aumento de complejidad, la existencia de tráfico aéreo segregado o una alta densidad de aeronaves en su sector de responsabilidad. Entre todos estos elementos, los eventos que se han utilizado para estimar la carga de trabajo de los ejercicios se resumen en la siguiente tabla:

Código	Puntuación	Descripción
LSSR	3	Identificación de responsabilidad de una aeronave
COO_A	4	Asunción de una aeronave
CC	7	Conflicto entre dos aeronaves en fase de crucero
VEC	10	Vectorización de una ruta
MON	2	Monitorización

Tabla 6.1: Puntuación de los eventos de los ejercicios

Las puntuaciones asignadas provienen de trabajos previos realizados en el laboratorio de simulación de la UPM y siguiendo este criterio de puntuación se ha calificado este ejercicio, igual que el resto de ejercicios del programa de ejercicios del simulador.

Los eventos que se han considerado son los que se muestran en la tabla 6.1:

- **LSSR**: se suman 3 puntos cuando se identifica una aeronave que se debe asumir dentro del tráfico adyacente al sector.
- **COO_A**: se suman 4 puntos al asumir una aeronave bajo responsabilidad. De esta manera, al asumir una nueva aeronave se suman un total de 3+4 puntos.
- **CC**: se suman 7 puntos cuando se produce un conflicto entre dos aeronaves en su fase de crucero. Como el simulador está configurado para la fase de ruta, todos los conflictos de este ejercicio se dan entre aeronaves en fase de crucero.
- **VEC**: consiste en la vectorización de la ruta, es decir, se suma cuando se hacen *re-routings* o directos. Para el cálculo de la carga de trabajo se suman 10 puntos siempre que haya un conflicto. Sin embargo, se podría recalificar el ejercicio en caso de que el conflicto se resuelva a través de cambio de velocidad o de nivel de vuelo.
- **MON**: se suman 2 puntos repartidos entre todo el tiempo que la aeronave está bajo la responsabilidad del controlador.

6.3. Programación del ejercicio

Como se ha mencionado en el capítulo anterior, la duración del ejercicio es de 60 minutos. Esta duración corresponde según [46] a la duración de los ejercicios de la fase básica, que es en la fase del programa de ejercicios donde deben encontrarse los participantes del estudio. El programa de ejercicios que se sigue está formado por una fase cero, en la que los alumnos se familiarizan con el simulador, y luego las fases: básica, intermedia, avanzada y de consolidación

Dentro de la ACC de Madrid existen 21 sectores repartidos en 3 grupos de niveles de vuelo: *lower*, *upper* y *medium*. Para la programación de este ejercicio se ha decidido utilizar el sector ZMU (Zamora *Upper*). Para que el ejercicio tenga una carga de trabajo elevada y favorecer el proceso de obtención de resultados el ejercicio se ha programado en dos ciclos: primero la carga de trabajo aumenta hasta alcanzar un máximo, luego llega a un valle y a continuación vuelve a subir hasta un segundo máximo y finalmente baja de nuevo.

En la figura 6.4 se muestran estos dos ciclos. Como se puede observar alrededor de los minutos 15 y 45 de simulación se producen los máximos de carga de trabajo (mayor número de aeronaves en el sector) mientras que en el minuto 30 hay un mínimo.

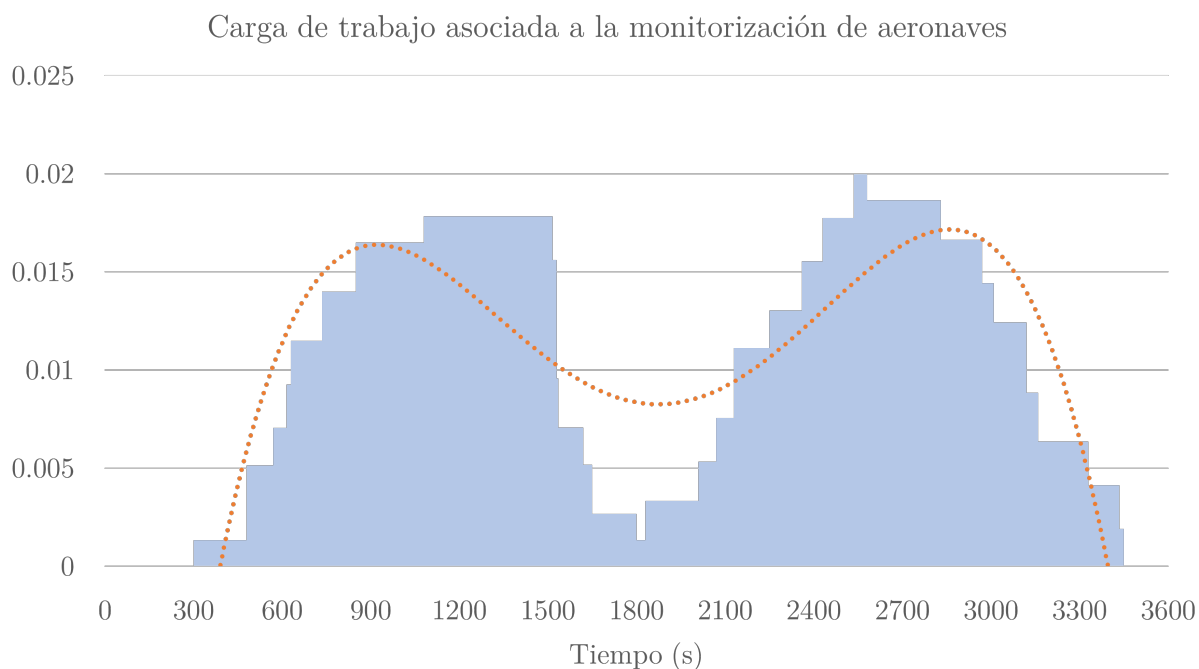


Figura 6.4: Carga de trabajo asociada a la monitorización de aeronaves

Para que la carga de trabajo sea elevada, pero esté dentro de la operación normal de los

controladores se han programado 18 aeronaves y 7 conflictos, de los cuales 5 son conflictos entre trayectorias y 2 son de alcance. De esta forma la puntuación total de carga de trabajo es: 281 puntos. La tabla de eventos del ejercicio se presenta a continuación:

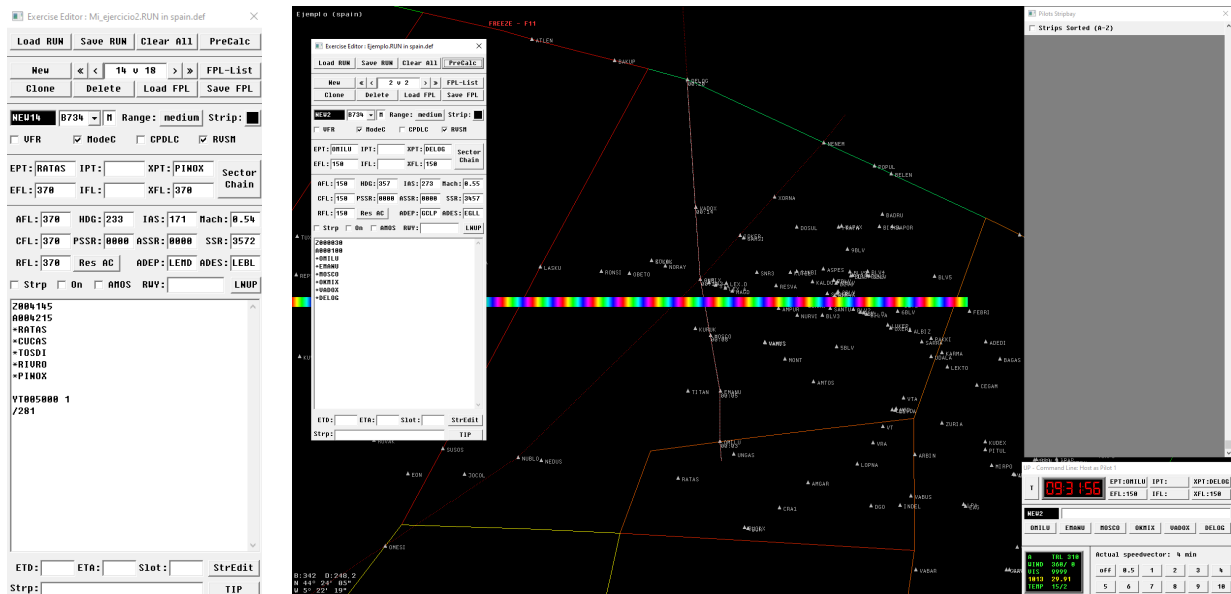
Evento	Identificador	Tiempo
Aparición aeronave: N9134	I_SSR + COO_A	05:00
Aparición aeronave: KLM1696	I_SSR + COO_A	08:00
Aparición aeronave: IB3830	I_SSR + COO_A	08:00
Aparición aeronave: TK1452	I_SSR + COO_A	09:30
Aparición aeronave: IB6586	I_SSR + COO_A	10:15
Aparición aeronave: IBE6756	I_SSR + COO_A	10:30
Aparición aeronave: IBE7632	I_SSR + COO_A	12:15
Conflicto	CC + VEC	13:05
Conflicto	CC + VEC	13:10
Aparición aeronave: NEW17	I_SSR + COO_A	14:10
Conflicto	CC + VEC	14:50
Aparición aeronave: NEW8	I_SSR + COO_A	18:00
Aparición aeronave: NEW9	I_SSR + COO_A	30:30
Aparición aeronave: NEW15	I_SSR + COO_A	33:30
Aparición aeronave: NEW16	I_SSR + COO_A	34:30
Aparición aeronave: NEW10	I_SSR + COO_A	35:30
Aparición aeronave: NEW11	I_SSR + COO_A	35:30
Conflicto	CC + VEC	35:45
Aparición aeronave: NEW12	I_SSR + COO_A	37:30
Conflicto	CC + VEC	39:10
Aparición aeronave: NEW18	I_SSR + COO_A	39:20
Conflicto	CC + VEC	39:50
Aparición aeronave: NEW13	I_SSR + COO_A	40:30
Aparición aeronave: NEW14	I_SSR + COO_A	42:15
Conflicto	CC + VEC	50:09

Tabla 6.2: Tabla de eventos del ejercicio de validación

Para la programación de los conflictos resulta útil la herramienta de ROSE *PreCalc* que permite conocer el tiempo de paso por cada punto de las aeronaves. La figura 6.5(a) representa los planes de vuelo de las aeronaves que hay que rellenar para que aparezca en el simulador.

La programación de aeronaves se realiza rellenando el plan de vuelo, introduciendo el

tiempo de aparición en la consola de ROSE (Zhhmss) y en el simulador (Ahhmss) y los puntos de la ruta. La información en detalle de cómo programar aeronaves se encuentra en [50]. Para la programación de los conflictos trayectorias la herramienta *Pre Calc* resulta muy útil, pues sólo hay que ajustar la velocidad de una de las dos aeronaves y comprobar que se va a infringir la distancia de seguridad mínima. Por otro lado, para la programación de los conflictos de velocidad se añade un *event catcher* de forma que una aeronave aumente o disminuya la velocidad para alcanzar o ser alcanzada por otra con la misma ruta. Esto se realiza con las dos últimas líneas que se ven en el plan de vuelo: YT005000 1, que significa que se produce un cambio de velocidad en el minuto 50 de simulación y que la velocidad se aumenta a 281 knots. De esta manera alcanza a otra aeronave que sigue la misma ruta en el mismo nivel de vuelo.



(a) Flight Plan

(b) Pre Calc

Figura 6.5: Programación de ejercicios en ROSE y cálculo de trayectorias

6.4. Programación de los mensajes en la aplicación

Los mensajes programados en la aplicación para la resolución óptima de los conflictos se muestran en la siguiente figura, donde el comando *time.sleep* es el tiempo entre mensajes en segundos. Además en la figura se pueden ver la resolución propuesta para los conflictos.

```

40     @staticmethod
41     def mensajes(cola, self): #Esta función me permite programar los mensajes en el tiempo
42         cola.put(" ")
43         time.sleep(785)
44         cola.put("IBE7632 rerouting por ODAPA y XERMA")
45         self.YES_button['state'] = tk.NORMAL
46         self.NO_button['state'] = tk.NORMAL
47         time.sleep(5)
48         cola.put("IBE7632 rerouting por ODAPA y XERMA          KLM1696 cambiar nivel de vuelo a FL39")
49         self.YES_button['state'] = tk.NORMAL
50         self.NO_button['state'] = tk.NORMAL
51         time.sleep(100)
52         cola.put("TK1452 rerouting por KASTA y RESBI")
53         self.YES_button['state'] = tk.NORMAL
54         self.NO_button['state'] = tk.NORMAL
55         time.sleep(255)
56         cola.put(" ")
57         self.YES_button['state'] = tk.DISABLED
58         self.NO_button['state'] = tk.DISABLED
59         time.sleep(1000)
60         cola.put("NEW16 rerouting por ADORO y MOSEN")
61         self.YES_button['state'] = tk.NORMAL
62         self.NO_button['state'] = tk.NORMAL
63         time.sleep(205)
64         cola.put("NEW11 directo a TLD9")
65         self.YES_button['state'] = tk.NORMAL
66         self.NO_button['state'] = tk.NORMAL
67         time.sleep(40)
68         cola.put("NEW11 directo a TLD9          NEW9 rerouting por RAKOD y BIRBA")
69         self.YES_button['state'] = tk.NORMAL
70         self.NO_button['state'] = tk.NORMAL
71         time.sleep(100)
72         cola.put(" ")
73         self.YES_button['state'] = tk.DISABLED
74         self.NO_button['state'] = tk.DISABLED
75         time.sleep(519)
76         cola.put("NEW13 cambiar nivel de vuelo a FL360")
77         self.YES_button['state'] = tk.NORMAL
78         self.NO_button['state'] = tk.NORMAL
79

```

Figura 6.6: Caption

6.5. Prueba de simulación

Se ha realizado una prueba de simulación con dos objetivos:

- que el ejercicio está programado de forma correcta y se ejecuta de forma correcta, sin errores ni fallos de ejecución.
- que la aplicación que introduce la nueva funcionalidad no interfiere en el desarrollo normal de la simulación.

Esta prueba se ha realizado solamente con un sujeto, que ha ejecutado el procedimiento

de validación descrito en el capítulo 5. Siguiendo este procedimiento, se han realizado dos simulaciones:

- La primera se ha realizado en condiciones normales, sin activar la nueva funcionalidad de resolución óptima de conflictos. En la figura 6.7 se puede observar la interfaz de la posición ATC durante la primera simulación. Se ha añadido un botón de conflicto a esta pantalla para registrar los conflictos que resuelve sin necesidad de revisar la grabación del ejercicio completo.

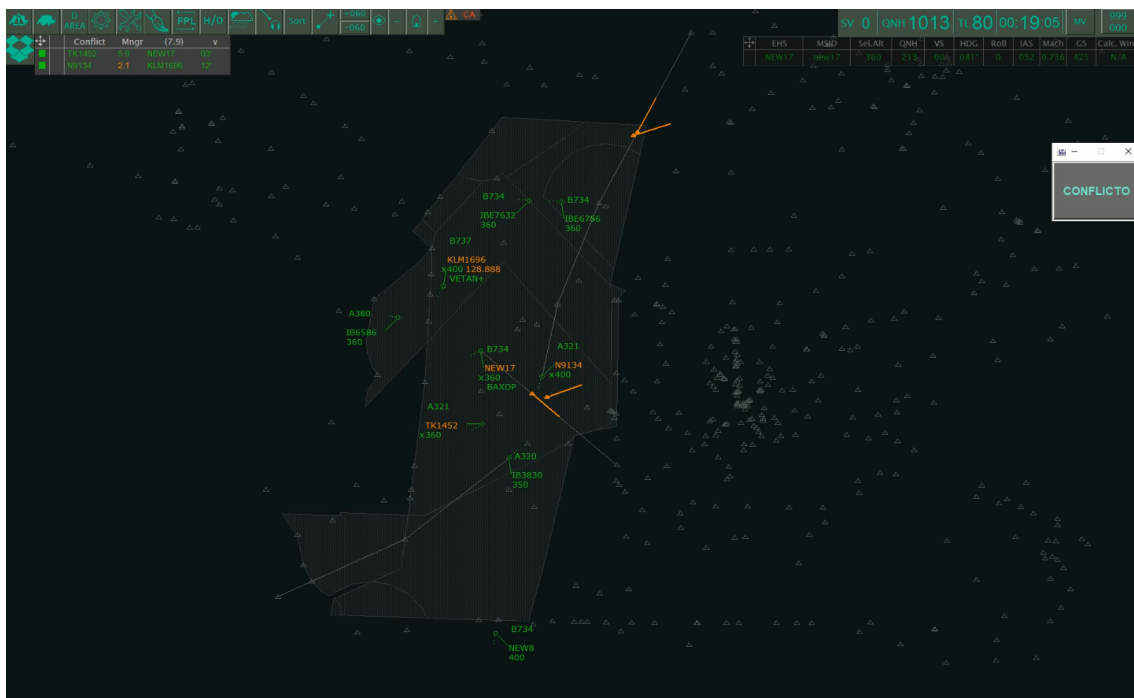


Figura 6.7: Interfaz de la simulación sin la resolución óptima de conflictos (21 de mayo)

- La segunda se ha realizado incorporando la nueva funcionalidad de resolución óptima de conflictos como se puede comprobar en la figura 6.8

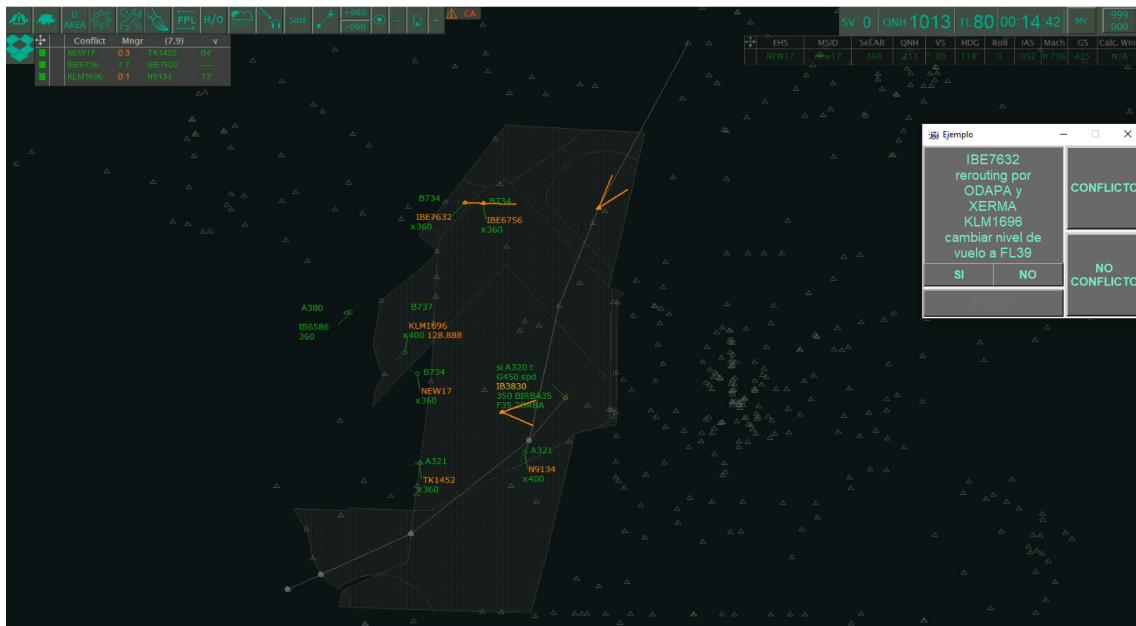


Figura 6.8: Interfaz de la simulación con la resolución óptima de conflictos (23 de junio)

Las dos pruebas han estado separadas en el tiempo por 33 días, para no condicionar los resultados de la prueba, pues se realiza el mismo ejercicio dos veces.



Figura 6.9: Desarrollo de la simulación



UNIVERSIDAD
POLITÉCNICA
DE MADRID

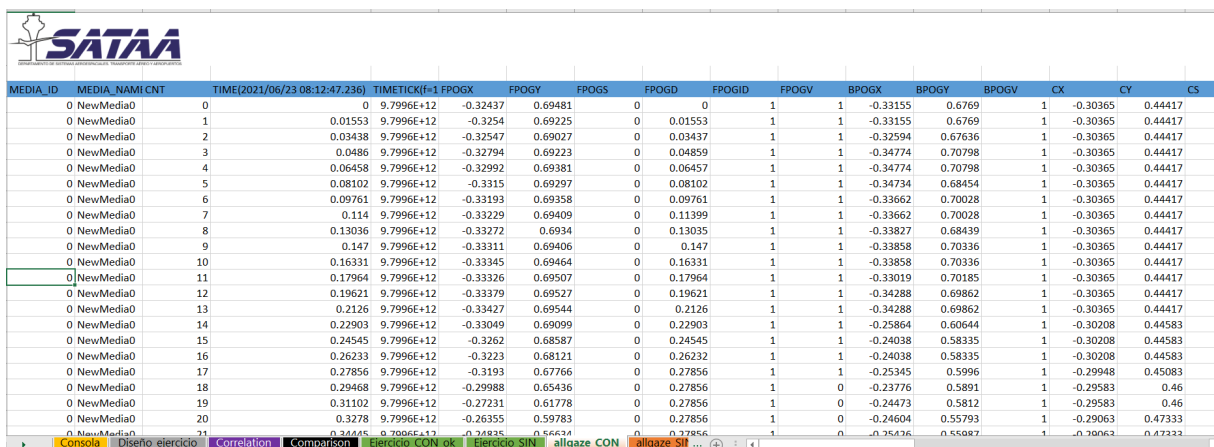


7. Análisis de resultados

Para la prueba de validación, entre las medidas posibles que se han descrito en el capítulo 5, se han utilizado medidas psicofisiológicas con el *Eye Tracker* y medidas de rendimiento.

7.1. Medidas psicofisiológicas

El *Eye tracker* proporciona una gran cantidad de parámetros relacionados con la atención del operador. En el Anexo B se encuentra el listado completo con todos los parámetros que se extraen del equipo. Estos parámetros se miden tomando 60 datos por segundo y el volumen de datos que se obtienen en una hora se han organizado en una tabla como la que muestra la siguiente figura:



MEDIA_ID	MEDIA_NAMICNT	TIME(2021/06/23 08:12:47.236)	TIMETICK(f-1 FPOGX	FPOGY	FPOGS	FPOGD	FPOGID	FPOGV	BPOGX	BPOGY	BPOGV	CX	CY	CS	
0	NewMedia0	0	9.7996E+12	-0.32437	0.69481	0	0	1	1	-0.33155	0.6769	1	-0.30365	0.44417	
0	NewMedia0	1	0.01553	9.7996E+12	-0.3254	0.69225	0	0.01553	1	1	-0.33155	0.6769	1	-0.30365	0.44417
0	NewMedia0	2	0.03438	9.7996E+12	-0.32547	0.69027	0	0.03437	1	1	-0.32594	0.67636	1	-0.30365	0.44417
0	NewMedia0	3	0.0486	9.7996E+12	-0.32794	0.69223	0	0.04859	1	1	-0.34774	0.70798	1	-0.30365	0.44417
0	NewMedia0	4	0.06458	9.7996E+12	-0.32992	0.69381	0	0.06457	1	1	-0.34774	0.70798	1	-0.30365	0.44417
0	NewMedia0	5	0.08102	9.7996E+12	-0.3315	0.69297	0	0.08102	1	1	-0.34734	0.68454	1	-0.30365	0.44417
0	NewMedia0	6	0.09761	9.7996E+12	-0.33193	0.69358	0	0.09761	1	1	-0.33662	0.70028	1	-0.30365	0.44417
0	NewMedia0	7	0.114	9.7996E+12	-0.33229	0.69409	0	0.11399	1	1	-0.33662	0.70028	1	-0.30365	0.44417
0	NewMedia0	8	0.13036	9.7996E+12	-0.33272	0.6934	0	0.13035	1	1	-0.33827	0.68439	1	-0.30365	0.44417
0	NewMedia0	9	0.147	9.7996E+12	-0.33311	0.69406	0	0.147	1	1	-0.33858	0.70336	1	-0.30365	0.44417
0	NewMedia0	10	0.16331	9.7996E+12	-0.33345	0.69464	0	0.16331	1	1	-0.33858	0.70336	1	-0.30365	0.44417
0	NewMedia0	11	0.17964	9.7996E+12	-0.33326	0.69507	0	0.17964	1	1	-0.33019	0.70185	1	-0.30365	0.44417
0	NewMedia0	12	0.19621	9.7996E+12	-0.33379	0.69527	0	0.19621	1	1	-0.34288	0.69862	1	-0.30365	0.44417
0	NewMedia0	13	0.2126	9.7996E+12	-0.33427	0.69544	0	0.2126	1	1	-0.34288	0.69862	1	-0.30365	0.44417
0	NewMedia0	14	0.22903	9.7996E+12	-0.33049	0.69099	0	0.22903	1	1	-0.25864	0.60644	1	-0.30208	0.44583
0	NewMedia0	15	0.24545	9.7996E+12	-0.3262	0.68587	0	0.24545	1	1	-0.24038	0.58335	1	-0.30208	0.44583
0	NewMedia0	16	0.26233	9.7996E+12	-0.3223	0.68121	0	0.26232	1	1	-0.24038	0.58335	1	-0.30208	0.44583
0	NewMedia0	17	0.27856	9.7996E+12	-0.3193	0.67766	0	0.27856	1	1	-0.25345	0.5996	1	-0.29948	0.45083
0	NewMedia0	18	0.29468	9.7996E+12	-0.29988	0.65436	0	0.27856	1	0	-0.23776	0.5891	1	-0.29583	0.46
0	NewMedia0	19	0.31102	9.7996E+12	-0.27231	0.61778	0	0.27856	1	0	-0.24473	0.5812	1	-0.29583	0.46
0	NewMedia0	20	0.3278	9.7996E+12	-0.26355	0.59783	0	0.27856	1	0	-0.24604	0.55793	1	-0.29063	0.47333
0	NewMedia0	21	0.34445	9.7996E+12	-0.24825	0.56634	0	0.27856	1	0	-0.25436	0.55087	1	-0.29063	0.47333

Figura 7.1: Datos exportados del Eye Tracker

El primer paso el tratamiento de los datos es la obtención de la línea temporal, para obtener esta línea se han seguido los siguientes pasos:

1. Primero se abren los datos en Excel y se localiza la columna **TIME**, que contiene la fecha y la hora del inicio de la grabación.
2. Una vez localizada esta columna, se debe crear la línea temporal de la simulación. Para ello, se deben borrar todos los datos tomados antes y después de la simulación, lo que es fácil de identificar gracias a los botones de las nuevas aplicaciones.
3. Una vez se han eliminado estos valores se crea la línea temporal del ejercicio

Entre todos los parámetros que se han obtenido, solamente se van a analizar tres de ellos: el número de fijaciones, la duración de las fijaciones y la frecuencia de parpadeo.

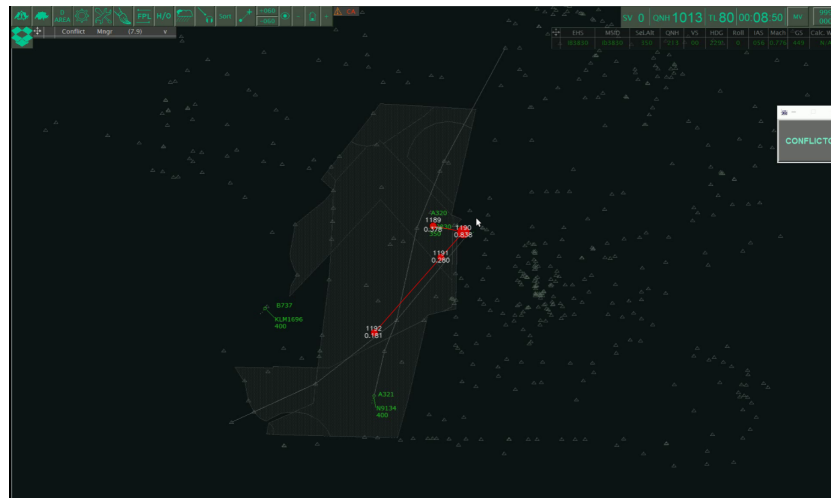
En la tabla 7.1 se resume la variación esperada de los parámetros al aumentar la carga de trabajo, de esta manera al obtener los resultados se puede comprobar si estos parámetros aumentan o disminuyen y por lo tanto obtener una conclusión del estudio.

Equipo	Parámetros	Indicador
<i>Eye tracker</i>	Número de fijaciones	↑
	Duración de las fijaciones	↓
	Frecuencia de parpadeo	↓

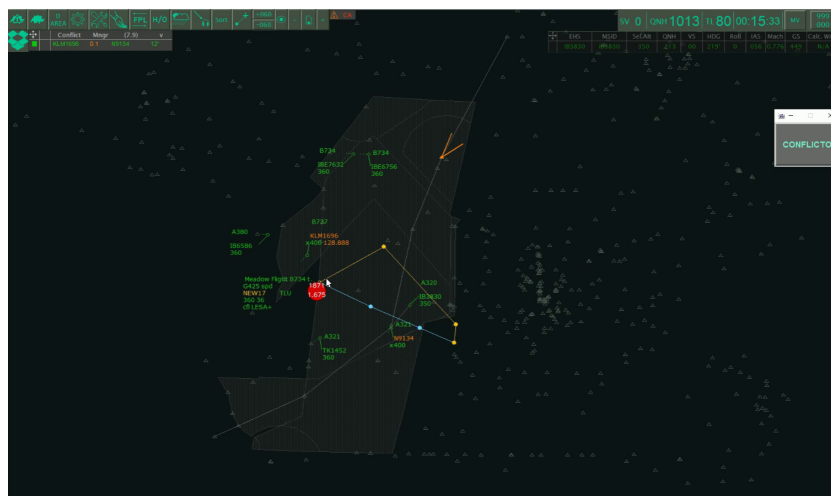
Tabla 7.1: Variación de los parámetros al aumentar la carga de trabajo

Para verificar estas tendencias, se ha realizado un análisis de correlaciones de estos parámetros con la carga de trabajo asociada a la monitorización de aeronaves. El resultado de este análisis ha demostrado que las variables varían siguiendo esa lógica, pese a que los valores numéricos del análisis no son representativos. Este análisis se puede utilizar una vez que la carga de trabajo total esté calculada e integrada temporalmente.

En la figura 7.2 se presenta un ejemplo de la duración de las fijaciones en dos momentos diferentes de la simulación. En la figura superior las fijaciones son más cortas y hay un mayor número de ellas, mientras que en la figura inferior la duración es más larga porque la atención se ha centrado en ese punto.



(a) Fijaciones cortas



(b) Fijaciones largas

Figura 7.2: Fijaciones con alta y baja carga de trabajo durante la simulación

A continuación se incluyen las gráficas que se han obtenido comparando las dos soluciones para los tres parámetros analizados.

Número de fijaciones

La figura 7.3 representa el número de fijaciones por minuto a lo largo de la simulación. Comparando las dos gráficas se puede intuir que la simulación con nueva funcionalidad es más independiente de la carga de trabajo que la simulación sin la nueva funcionalidad, que como se ve en la línea de tendencia azul, sigue la misma forma que la carga de trabajo.

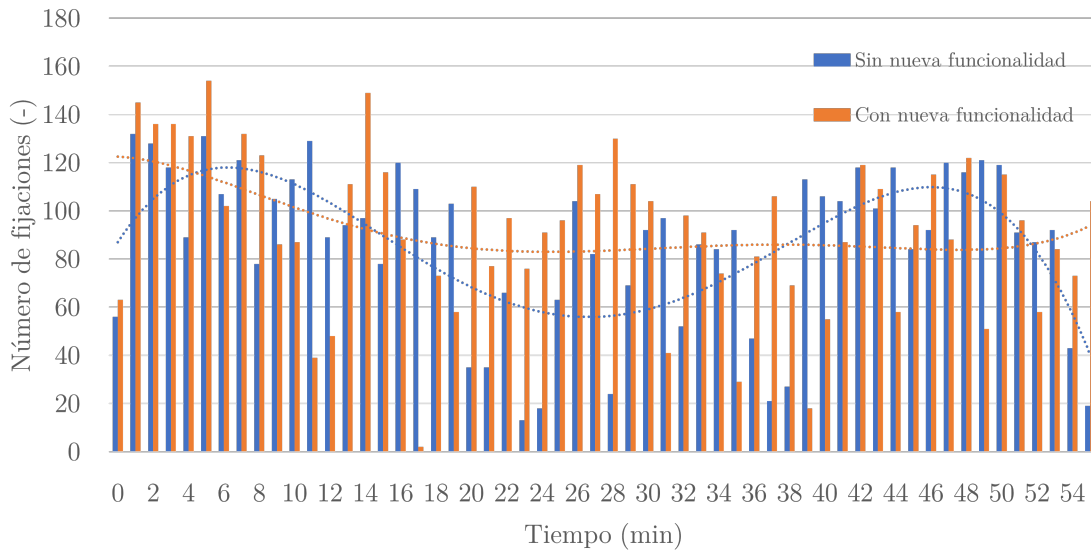


Figura 7.3: Número de fijaciones por minuto

Duración de las fijaciones

La figura 7.4 representa la duración de las fijaciones a lo largo de todo el ejercicio. Se puede observar que en determinados momentos el operador ha mirado fijamente a la pantalla durante mas tiempo, esto puede deberse a tener que buscar el nombre del *waypoint* de la solución propuesta en la aplicación. No obstante, en general la duración de las fijaciones ha sido ligeramente menor en el ejercicio con la nueva funcionalidad incorporada.

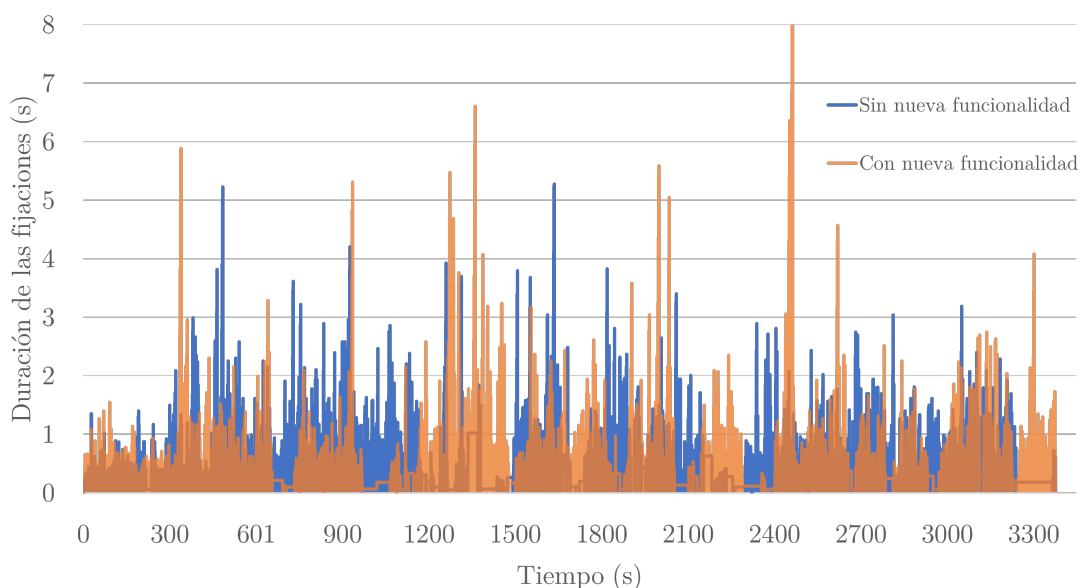


Figura 7.4: Tiempo de duración de las fijaciones

Frecuencia de parpadeo

El último parámetro analizado es la frecuencia de parpadeo. Se puede observar variabilidad en los resultados de la frecuencia de parpadeo para ambas simulaciones, sin embargo en general los valores de parpadeo para la simulación sin funcionalidad son mayores que para la simulación con funcionalidad.

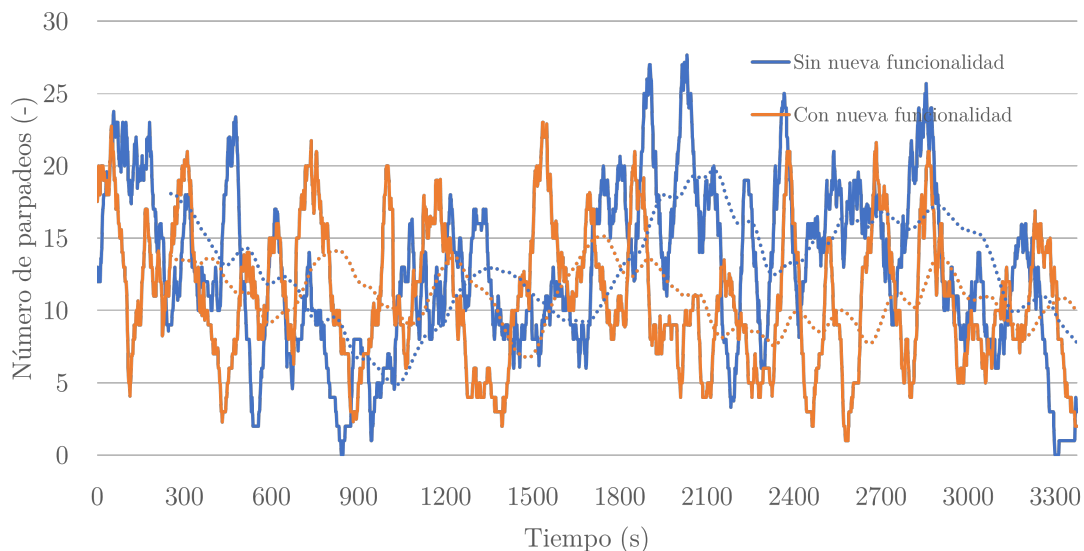


Figura 7.5: Frecuencia de parpadeo

7.2. Medidas de rendimiento

La medida de rendimiento que se va a utilizar es el número de conflictos resueltos durante los ejercicios de simulación. Sólo se utiliza esta variable porque el número de aeronaves asumidas y transferidas se supone el mismo en el ejercicio sin nueva funcionalidad y con nueva funcionalidad.

En la figura 7.6 se pueden observar los archivos que se generan con la información de los botones. A la izquierda se puede comprobar que el sujeto resolvió 11 conflictos mientras que en la figura de la derecha resolvió 7 conflictos. Como en ambos casos el ejercicio está inicialmente programado con 7 conflictos, cuando se realizó el ejercicio sin la resolución óptima, se causaron 4 conflictos adicionales al intentar evitar los anteriores. Sin embargo, cuando se realizó el ejercicio con la ventana, se resolvieron 7, todos ellos utilizando la solución recomendada (pulsar el botón SI).

Los clicks en los botones CONFLICTO y NO CONFLICTO se deben a un error de sincronización entre el simulador y los mensajes, pues se le indicó que los pulsara en caso de

conflicto sin solución o solución sin conflicto.

Ejercicio:Ej_Ariadna_1	Ejercicio:Con funcionalidad ejercicio 2
CONFLICTO-8:40:41	INICIO-8:38:38
CONFLICTO-8:41:47	NO CONFLICTO-8:41:13
CONFLICTO-8:42:59	SI-8:41:14
CONFLICTO-8:44:16	SI-8:43:14
CONFLICTO-8:46:21	NO CONFLICTO-8:43:15
CONFLICTO-8:48:10	CONFLICTO-8:45:36
CONFLICTO-8:57:31	SI-9:4:2
CONFLICTO-9:3:11	NO CONFLICTO-9:4:4
CONFLICTO-9:7:12	SI-9:8:49
CONFLICTO-9:15:12	NO CONFLICTO-9:8:50
CONFLICTO-9:17:41	SI-9:9:8
	SI-9:18:15
	NO CONFLICTO-9:18:16
	SI-9:20:10

(a) Sin la nueva funcionalidad

(b) Con la nueva funcionalidad

Figura 7.6: Resultado de los reports de los botones

8. Conclusiones y líneas futuras

El aumento de la demanda de transporte aéreo de los últimos años, a pesar de las situaciones de crisis económicas o sanitarias que han ralentizado el crecimiento, hace que sea necesario continuar mejorando los sistemas y optimizando el uso del espacio aéreo. Uno de los principales problemas al que tiene que enfrentarse el sistema ATM es a la congestión de la red.

Este problema afecta directamente a los controladores aéreos que cada vez tienen que asumir un mayor volumen de aeronaves, lo que conlleva un aumento de su carga de trabajo. Para afrontar la congestión de la red han surgido numerosas iniciativas para modernizar el sistema ATM, siendo SESAR la más relevante. El programa SESAR ha iniciado numerosos proyectos para ayudar a los controladores en el desempeño de su labor. Dentro de estos proyectos se encuentra uno de los objetivos de este trabajo, que es el diseño de una herramienta que proporcione al controlador la resolución óptima de los conflictos entre aeronaves.

El impacto que tiene introducir una nueva herramienta en las posiciones de control puede afectar a la carga de trabajo positiva o negativamente desde el punto de vista de los factores humanos, y es por ello que aunque una solución sea óptima para el sistema ATM, es posible que dificulte las tareas del controlador y les aumente la carga de trabajo en lugar de disminuirla.

La implementación de la aplicación en el simulador ha comprobado que teniendo la resolución óptima de los conflictos el ejercicio obtiene mejores resultados en número de conflictos, lo que es positivo para el sistema ATM. Sin embargo, la validación de que esta herramienta aumenta o disminuye la carga de trabajo se hará siguiendo el programa de validación, realizando pruebas con un mayor número de sujetos de acuerdo al programa diseñado en el capítulo 5.

Las principales líneas de trabajo futuro que se han identificado durante la realización de este proyectos son las siguientes:

- La necesidad de una metodología para asignar la carga de trabajo de los ejercicios. Si desde las medidas obtenidas de las simulaciones se establecen correlaciones entre estas medidas y los eventos de los ejercicios se podrían obtener puntuaciones de medida de la carga de trabajo más precisas, así como una carga de trabajo integrada temporalmente en vez de la curva discretizada.
- Automatizar el procesamiento de datos para que un mayor número de alumnos realicen las simulaciones y de esta manera obtener resultados representativos.

Referencias

- [1] Aricò, P.; Borghini, G.; Di Flumeri, G.; Bonelli, S. y Golfetti, A (2017). Human Factors and Neurophysiological Metrics in Air Traffic Control: a Critical Review. *IEEE Reviews in Biomedical Engineering* doi: 10.1109/RBME.2017.2694142
- [2] Borst, C.; d'Engelbronner, J.G.; Ellerbroek, J.; Mulder, M. y van Paassen, M. M. (2015). Solution Space Based Analysis of Dynamic Air Traffic Controller Workload. *Journal of Aircraft* doi: 10.2514/1.C032847
- [3] Borst, C.; Mulder, M.; Van Paassen, M. M. y Visser, R. M. (2019). Exploring Short-Term Training Effects of Ecological Interfaces: A Case Study in Air Traffic Control. *IEEE Transactions on Human-Machine Systems*, 49(6), 623-632. [8735791]. <https://doi.org/10.1109/THMS.2019.2919742>
- [4] Cain, B. (2007). *A review of the mental workload literature. (NATO Report No. RTO-TR-HFM-121-Part II)*. Toronto, Canada: Defence Research and Development.
- [5] EMOTIV <https://www.emotiv.com/product/emotiv-insight-5-channel-mobile-eeg/>
- [6] ENAIRE <https://www.enaire.es/home>
- [7] ENAIRE (2018) *Plan de Vuelo 2020*.
- [8] Endsley, M. R.; Jones, D. G. (2004) *Designing for Situation Awareness. An approach to Used-Centered Design. 2nd Edition*. CRC Press. Taylor & Francis Group
- [9] EUROCONTROL <https://www.eurocontrol.int/>
- [10] EUROCONTROL (2019). *Performance Review Report 2019*.
- [11] EUROCONTROL (2015). *Performance Review Report 2015*.
- [12] EUROCONTROL (2010). *Human Performance in Air Traffic Management Safety. A White Paper*
- [13] EUROCONTROL (2003). *Pessimistic Sector Capacity Estimation*. EUROCONTROL Experimental Centre.
- [14] EUROCONTROL (2019). *LSSIP 2019 Spain Local Single Sky Implementation*.
- [15] Gazept <https://www.gazept.com/>
- [16] Genovés Pérez, Eduardo (2021) *Análisis de factores humanos para la mejora del rendimiento del controlador aéreo*

- [17] Hancock, Peter. A. (Ed.) (1999) *Human Performance and Ergonomics*. Academic Press.
- [18] Herrero Sanz, Javier (2020) *Propuesta de plan de formación ATC basado en simuladores para un entorno automatizado*
- [19] ICAO www.icao.int
- [20] ICAO (1992) *Air Traffic Services Planning Manual*. Document Sales Unit ICAO. Doc 9426-AN/924
- [21] ICAO (2001) *Annex 11 of the Convention on International Civil Aviation*. 13th Edition.
- [22] ICAO (2021) *Effects of Novel Coronavirus (COVID-19) on Civil Aviation: Economic Impact Analysis*. Economic Development - Air Transport Bureau 1 June 2021.
- [23] ICAO (2016). *Global Air Navigation Plan (GANP)*.
- [24] GARTEUR Action Group FM AG13 (2003). *GARTEUR Handbook of Mental Workload Measurement*.
- [25] IFATCA (2018). *TECHNICAL AND PROFESSIONAL MANUAL*. International Federation of Air Traffic Controllers' Associations.
- [26] IFATCA (2019) *Remote Towers*. International Federation of Air Traffic Controllers' Associations. <https://www.ifatca.org/remote-towers-guidance/>
- [27] Loura, J. y Yadav, A. S. (2013). Air Traffic Control Concept from a Human Factors Perspective-A Review. *International Journal of Trends in Human Resource Management* 2, issue 7.
- [28] Majumdar, A. y Ochieng, W. (2002) The factors affecting air traffic controller workload: a multivariate analysis based upon simulation modelling of controller workload. *Publisher: Transportation Research Board*, 1788, 58-69 doi: 10.3141/1788-08
- [29] Martins, A.P.G. (2016). A review of important cognitive concepts in aviation. *Taylor & Francis. Volumen 20(2): 65-84*. doi: 10.3846/16487788.2016.1196559
- [30] Miller, S. (2001) *Literature review: workload measures*. National Advanced Driving Simulator, University of Iowa.
- [31] Mulder, M.; Yang, L.; Borst, C. y Van Paassen, M. M. (2019). Solution-Space-based ATC support for 4DT heterogeneous aircraft-mix control. *IEEE International Conference on Systems, Man and Cybernetics, SMC 2019* Vol.2019-October, 3192-3197. <https://doi.org/10.1109/SMC.2019.8914657>

- [32] Murcia Palazón, Juan Antonio (2021). *Plan de Actuación y Evaluación de Nuevas Funcionalidades en un Sistema ATC*. Universidad Politécnica de Madrid.
- [33] Nolan, Michael S. (2011). *Fundamentals of Air Traffic Control. 5th Edition*.
- [34] <https://www.python.org/>
- [35] Ribeiro, M.; Ellerbroek, J. y Hoekstra, J. (2020). Review of conflict resolution methods for manned and unmanned aviation. *Aerospace*, 7(6), 79. <https://doi.org/10.3390/AEROSPACE7060079>
- [36] Rodríguez González, Pelayo (2021) *Viabilidad del empleo de técnicas de electroencefalografía en el control del tráfico aéreo*
- [37] ROSE Simulation (2018). *ROSE Handbook*. Version 2.23.772.
- [38] Rubio Sobrino, A. (2020). *Análisis sobre la aplicación efectiva de sistemas optimizados de gestión de información y de planificación de la capacidad en el espacio aéreo español y red de aeropuertos de Aena: propuesta de solución integrada*.
- [39] Ruitenbergh, B. (1999) Ten Years of Human Factors in Air Traffic Control. *The Controller. Journal of Air Traffic Control. IFATCA*, 38, 24-27
- [40] SESAR JU https://www.atmmasterplan.eu/depl/essip_objectives/map?objId=1000378
- [41] SESAR JU <https://www.sesarju.eu/>
- [42] SESAR JU <https://www.atmmasterplan.eu/>
- [43] SESAR JU (2020). *European ATM master plan. Executive view*. Publications Office of the European Union Luxembourg. doi:10.2829/695700
- [44] SESAR JU (2012). *Concept of Operation Step 1*. Project ID: B4.2
- [45] SESAR JU (2016). *Sesar Solutions Catalogue* Publications Office of the European Union Luxembourg. doi:10.2829/84570
- [46] SESAR JU (2016). *Competence and Training Requirements. AUTOPACE D3.2 H2020-SESAR-2015-1*
- [47] SESAR JU (2018) *Autopace: Final Project Results Report*
- [48] Tobaruela Arnedo, Gonzalo (2015). *A framework to assess the ability of automation to deliver capacity targets in European airspace*. Imperial College London, UK.

- [49] Universidad Politécnica de Madrid (2017) *Grupo de Investigación de Navegación Aérea (GINA). Líneas de investigación*
- [50] SATAA (2020) *Manual de uso del laboratorio ATC*
- [51] Wickens, C. D.; Mavor, A. S. y McGee, J. P. (Eds). (1997) *Flight to the future: Human factors in air traffic control*. Washington, D.C.: National Academy Press. <https://doi.org/10.17226/5493>

A. Código de Python de la nueva funcionalidad

En este anexo se encuentra el código completo de la nueva funcionalidad. Este código esta dividido en las siguientes partes:

1. Importación de librerías
2. Declaración de variables globales
3. Primera clase: ventana principal
4. Segunda clase: ventana de salida
5. Declaración de la función principal

```

'''
Ayuda a la resolución de conflictos en posiciones ATC
Ariadna Gómez Rozalén
'''

#Librerías
import tkinter as tk          #ventanas emergentes
from tkinter import ttk      #módulo ttk para usar widgets visuales
import time                   #permite manejar variables en el tiempo
import threading              #gestionar operaciones concurrentes dentro de un proceso (varios hilos de ejecución)
import queue                  #implementar colas multi-productor y multi-
                              #consumidor (para intercambiar información entre subprocesos)
from datetime import datetime #Capturar la fecha y hora
from reportlab.pdfgen import canvas #Creación de reportes en pdf
from reportlab.lib.pagesizes import A4
import os                    #Creación de .txt

#Definición de variables globales que se pueden usar en diferentes clases
global cont
global time

#Primera clase: Ventana principal
class Ventana_principal(tk.Frame):

    #Aquí se inicializan los atributos de cada clase
    def __init__(self, master):
        super(Ventana_principal, self).__init__(master)
        master.title("Ejemplo") #Título de la ventana
        master.columnconfigure(1, weight=1) #Configuración de las columnas de la ventana
        master.columnconfigure(1, weight=1)
        master.rowconfigure(1, weight=1) #Configuración de las filas de la ventana
        master.rowconfigure(1, weight=1)
        self.queue = queue.Queue() #Cola de actualizar ventana
        self.create_widgets() #Creación de widgets
        self.time=[0]*200 #Variable que almacena la información de los botones
        self.cont=0 #Variable contador de clicks de los botones

        self.master = master
        self.master.protocol("WM_DELETE_WINDOW", self.on_closing)

    @staticmethod
    def mensajes(cola, self): #Esta función me permite programar los mensajes en el tiempo
        cola.put(" ")
        time.sleep(4)
        cola.put("Solución: subir NEW 2 a FL160")
        self.YES_button['state'] = tk.NORMAL
        self.NO_button['state'] = tk.NORMAL
        time.sleep(10)
        cola.put(" ")
        self.YES_button['state'] = tk.DISABLED
        self.NO_button['state'] = tk.DISABLED
        time.sleep(4)
        cola.put("Solución: desviar NEW 3 por ...")
        self.YES_button['state'] = tk.NORMAL
        self.NO_button['state'] = tk.NORMAL

```

```

def update_status(self):    #Esta función inicia la ventana de mensajes
    threading.Thread(target=self.mensajes, args=(self.queue,self)).start()
    self.after(100, self.process_queue)
    self.START_button['state'] = tk.DISABLED
    self.time[self.cont]= "INICIO-"
" + str(datetime.now().hour) + ":" + str(datetime.now().minute) + ":" + str(datetime.now().second)
    self.cont+=1

def process_queue(self):    #Esta función procesa la cola
    try:
        data = self.queue.get_nowait()
        self.text.set(data)
    except queue.Empty:
        pass
    self.master.after(100, self.process_queue) #100 ms

#Funciones que capturan en el vector self.time los clicks en los botones
def CONFLICT(self):
    self.time[self.cont]= "CONFLICTO-"
" + str(datetime.now().hour) + ":" + str(datetime.now().minute) + ":" + str(datetime.now().second)
    self.cont+=1

def NOCONFLICT(self):
    self.time[self.cont]= "NO CONFLICTO-"
" + str(datetime.now().hour) + ":" + str(datetime.now().minute) + ":" + str(datetime.now().second)
    self.cont+=1

def YES(self):
    self.time[self.cont]= "SI-"
" + str(datetime.now().hour) + ":" + str(datetime.now().minute) + ":" + str(datetime.now().second)
    self.cont+=1

def NO(self):
    self.time[self.cont]= "NO-"
" + str(datetime.now().hour) + ":" + str(datetime.now().minute) + ":" + str(datetime.now().second)
    self.cont+=1

#Función que crea el aspecto de la ventana
def create_widgets(self):

    # Inicia el estilo
    self.s = ttk.Style()

    #Crea el estilo por defecto en todos los marcos
    self.s.configure('TFrame', background='dimgray')
    #Crea el estilo del primer marco
    self.s.configure('Frame1.TFrame', background='dimgray')

    #Parte izquierda
    self.left_pane = tk.PanedWindow(root, orient=tk.VERTICAL)
    self.left_pane.grid(column=0, row=0, rowspan=1, sticky=(tk.N, tk.W, tk.E, tk.S))
    self.left_pane.rowconfigure(0, weight=1)

```

```

self.left_pane.columnconfigure(0, weight=1)
self.left_upperframe = ttk.Frame(root, relief='groove', borderwidth=1, style='Frame1.TFrame')
self.left_upperframe.grid(column=0, row=0, sticky=(tk.N, tk.W, tk.E, tk.S))
self.left_upperframe.columnconfigure(0, weight=1)
self.left_upperframe.rowconfigure(0, weight=1)
self.left_pane.add(self.left_upperframe, height=200, width=200)
self.left_bottomframe = ttk.Frame(self.left_pane, relief='groove', borderwidth=1)
self.left_bottomframe.grid(column=0, row=1, sticky=(tk.N, tk.W, tk.E, tk.S))
self.left_bottomframe.columnconfigure(0, weight=1)
self.left_bottomframe.rowconfigure(0, weight=1)
self.left_pane.add(self.left_bottomframe, height=40)

#Parte derecha
self.right_pane = tk.PanedWindow(root, orient=tk.VERTICAL)
self.right_pane.grid(column=1, row=0, rowspan=1, sticky=(tk.N, tk.W, tk.E, tk.S))
self.right_pane.rowconfigure(0, weight=1)
self.right_pane.columnconfigure(0, weight=1)
self.right_upperframe = ttk.Frame(root, relief='groove', borderwidth=2)
self.right_upperframe.grid(column=0, row=0, sticky=(tk.N, tk.W, tk.E, tk.S))
self.right_upperframe.columnconfigure(0, weight=1)
self.right_upperframe.rowconfigure(0, weight=1)
self.right_pane.add(self.right_upperframe, width=110, height=120)
self.right_bottomframe = ttk.Frame(self.right_pane, relief='groove', borderwidth=2)
self.right_bottomframe.grid(column=1, row=1, sticky=(tk.N, tk.W, tk.E, tk.S))
self.right_bottomframe.columnconfigure(0, weight=1)
self.right_bottomframe.rowconfigure(0, weight=1)
self.right_pane.add(self.right_bottomframe, width=110, height = 100)

#Botón START
self.START_button = tk.Button(self.left_bottomframe, fg="aquamarine2", bg="dimgray", font="arial 14 bold")
self.START_button["text"] = "START"
self.START_button["command"] = self.update_status
self.START_button.grid(column=0, row=0, sticky=(tk.W, tk.S, tk.E, tk.N))

#Texto programado antes de iniciar
self.labl=tk.Label(self.left_upperframe)
self.text = tk.StringVar()
self.text.set("Pulsa START \n")
self.labl.config(textvar=self.text, font="arial 14", fg="aquamarine2", bg="dimgray")
self.labl.grid(column=0, row=0, sticky="n", pady=4, padx=5)
self.labl.bind("<Configure>", self.on_label_resize)

#Botón CONFLICTO
self.CONFLICT_button = tk.Button(self.right_upperframe, fg="aquamarine2", bg="dimgray", font="arial 13 bold")
self.CONFLICT_button["text"] = "CONFLICTO"
self.CONFLICT_button["command"] = self.CONFLICT
self.CONFLICT_button.grid(column=0, row=0, sticky=(tk.W, tk.S, tk.E, tk.N))

#Botón NO CONFLICTO
self.NOCONFLICT_button =tk.Button(self.right_bottomframe, fg="aquamarine2", bg="dimgray", font="arial 13 bold")
self.NOCONFLICT_button["text"]="NO CONFLICTO"
self.NOCONFLICT_button.bind("<Configure>", self.on_label_resize_2)

```

```

self.NOCONFLICT_button["command"] = self.NOCONFLICT
self.NOCONFLICT_button.grid(column=0, row=0, sticky=(tk.W, tk.S, tk.E, tk.N))

#Botón SI
self.YES_button= tk.Button(self.left_upperframe,fg="aquamarine2", bg="dimgray", font="arial 13 bold", width=9)
self.YES_button["text"]="SI"
self.YES_button.bind("<Configure>", self.on_label_resize_2)
self.YES_button["command"] = self.YES
self.YES_button.grid(column=0, row=0, sticky=("sw"))
self.YES_button['state'] = tk.DISABLED

#Botón NO
self.NO_button= tk.Button(self.left_upperframe,fg="aquamarine2", bg="dimgray", font="arial 13 bold", width= 9)
self.NO_button["text"]="NO"
self.NO_button.bind("<Configure>", self.on_label_resize_2)
self.NO_button["command"] = self.NO
self.NO_button.grid(column=0, row=0, sticky=("se"))
self.NO_button['state'] = tk.DISABLED

#Tamaño del texto del botón START 150 puntos
def on_label_resize(self, event):
    event.widget.configure(wraplength=150)

#Tamaño del texto de los demás botones de 100 puntos
def on_label_resize_2(self, event):
    event.widget.configure(wraplength=100)

#Función que llama a la ventana de salida
def on_closing(self):
    d = MyDialog(root,self.time,self.cont)
    self.master.wait_window(d.top)

#Segunda clase: Ventana de salida
class MyDialog:
    def __init__(self, parent, time, cont):
        self.top = tk.Toplevel(parent)
        self.parent = parent
        self.top.title(" ")
        self.top.configure(bg='dimgray')
        self.top.wm_attributes("-topmost", True)
        self.top.iconbitmap('c:\\Users\\Ariadna\\Documents\\UNIVERSIDAD\\TFM\\Mi_nueva_funcionalidad\\Logo_SATAA.ico')
        self.top.resizable(width=False, height=False)

        self.time = time
        self.cont = cont

        tk.Label(self.top, text="Nº ejercicio",fg="aquamarine2", bg="dimgray", font="arial 10 bold").grid(row=0, column=0)

        self.ejercicio=tk.Entry(self.top)
        self.ejercicio.place(x=100, y=50)
        self.ejercicio.grid(row=0, column=1, padx=5, pady=5)

        tk.Label(self.top, text="Nombre",fg="aquamarine2", bg="dimgray", font="arial 10 bold").grid(row=1, column=0)
        tk.Label(self.top, text="Apellido",fg="aquamarine2", bg="dimgray", font="arial 10 bold").grid(row=1, column=1)

```

```

self.nombre=tk.Entry(self.top)
self.nombre.place(x=50, y=50)
self.nombre.grid(row=2, column=0, padx=5, pady=5)

self.apellido=tk.Entry(self.top)
self.apellido.place(x=100, y=50)
self.apellido.grid(row=2, column=1, padx=5, pady=5)

self.SAVEANDCLOSE_button = tk.Button(self.top, text="Guardar y salir", command=self.salir,fg="aquamarine2", bg
="dimgray", font="arial 10")
self.CANCEL_button = tk.Button(self.top, text="Cancelar", command=self.minimizar,fg="aquamarine2", bg="dimgray
", font="arial 10")
self.SAVEANDCLOSE_button.grid(row=3, column=0, padx=5, pady=5)
self.CANCEL_button.grid(row=3, column=1, padx=5, pady=5)

#Función que al salir guarda la información en txt y pdf
def salir(self):
    self.nombre_1=self.nombre.get()
    self.apellido_1=self.apellido.get()
    self.ejercicio_1=self.ejercicio.get()
    c=canvas.Canvas("C:\\Users\\Ariadna\\Documents\\UNIVERSIDAD\\TFM\\Mi_nueva_funcionalidad\\reports\\Report -
Ejercicio "+self.ejercicio_1+ " - "+ self.nombre_1 +" "+ self.apellido_1 +".pdf",pagesize=A4)
    w, h = A4
    x_offset = 50
    y_offset = 50
    c.setFillColorRGB(0,0,0) #Color de la fuente
    c.setFont("Helvetica", 12) #Tipo de fuente y tamaño
    c.drawString(x_offset,h-
y_offset,"Ejercicio:"+ self.ejercicio_1 +" Alumno: " + self.nombre_1 +" "+ self.apellido_1)
    c.drawString(x_offset,h-y_offset+20,"Fecha:"+str(datetime.now().day)+"-"+str(datetime.now().month)+"-"
+str(datetime.now().year))

    for i in range(self.cont):
        c.drawString(x_offset,h-y_offset-20*(i+1),self.time[i])
    c.showPage()
    c.save()

    f = open ("C:\\Users\\Ariadna\\Documents\\UNIVERSIDAD\\TFM\\Mi_nueva_funcionalidad\\reports\\Report -
Ejercicio "+self.ejercicio_1+ " - "+ self.nombre_1 +" "+ self.apellido_1 +".txt",'w')

    f.write("Ejercicio:"+ self.ejercicio_1 +" Alumno: " + self.nombre_1 +" "+ self.apellido_1 +"\n")

    for i in range(self.cont):
        f.write(self.time[i]+"\n")
    f.close()
    self.top.destroy()
    self.parent.destroy()

def minimizar(self):
    self.top.destroy()

#Declara que la función superior es la ventana principal

```

```
if __name__ == "__main__":  
    root = tk.Tk()  
    root.resizable(width=False, height=False) #Fija el tamaño de la ventana  
    root.wm_attributes("-topmost", True) # Esta línea es importante: fija la ventana y se puede trabajar por debajo  
    root.iconbitmap('c:\\Users\\Ariadna\\Documents\\UNIVERSIDAD\\TFM\\Mi_nueva_funcionalidad\\Logo_SATAA.ico')  
    app = Ventana_principal(master=root)  
    app.mainloop()
```


B. Parámetros del Eye Tracker

En la siguiente tabla se presentan los parámetros exportados del *Eye Tracker*. Estos datos se recogen con una frecuencia de 60 datos por segundo desde el momento que se inicia la grabación.

Nombre	Descripción
MEDIA_ID	Identificador numérico
MEDIA_NAME	Nombre del elemento
CNT	Cada uno de los datos que toma el sistema
TIME	Tiempo de cada toma de datos
TIMETICK (f=1000000)	Momentos registrados para sincronizar con otras aplicaciones
FPOGX	Coordenada X de fijación como porcentaje del ancho de pantalla
FPOGY	Coordenada Y de fijación como porcentaje del altura de pantalla
FPOGS	Tiempo de inicio de la fijación en segundos
FPOGD	Tiempo de la duración de la fijación
FPOGID	La fijación en número para distinguir
FPOGV	Parámetro del FPOG indicando si es válido o no
BPOGX	Coordenada X de la media de datos no filtrados en el ancho de la pantalla
BPOGY	Coordenada Y de la media de datos no filtrados en la altura de la pantalla
BPOGV	Indica si BPOG es un dato válido o no
CX	Coordenada X de la posición del ratón en la pantalla
CY	Coordenada Y de la posición del ratón en la pantalla
CS	Estado del ratón
USER	Campo de datos personalizado
LPCX	Coordenada X en porcentaje de la pupila izquierda en la pantalla
LPCY	Coordenada Y en porcentaje de la pupila izquierda en la pantalla
LPD	Diámetro de la pupila del ojo izquierdo en milímetros
LPS	Factor de escala de la pupila del ojo izquierdo en milímetros
LPV	Indica si el parámetro de la pupila izquierda es correcto o no
RPCX	Coordenada X en porcentaje de la pupila derecha en la pantalla
RPCY	Coordenada Y en porcentaje de la pupila derecha en la pantalla
RPD	Diámetro de la pupila del ojo derecha en milímetros

Tabla B.1: Parámetros exportados del *Eye Tracker* (I)

Nombre	Descripción
RPS	Factor de escala de la pupila del ojo derecha en milímetros
RPV	Indica si el parámetro de la pupila derecha es correcto o no
BKID	Indicador del parpadeo, marca 0 cuando no se produce parpadeo
BKDUR	Duración del parpadeo anterior en segundos
BKPMIN	Número de parpadeos en los últimos 60 segundos
LPMM	Diámetro de la pupila izquierda en milímetros
LPMMV	Indicador del LPMM si es válido o no
RPMM	Diámetro de la pupila derecha en milímetros
RPMMV	Indicador del RPMM si es válido o no
DIAL	Posición de la persona
DIALV	Indicador del DIAL si es válido o no

Tabla B.2: Parámetros exportados del *Eye Tracker* (II)