



eetac

Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FIN DE GRADO

Título: MRO 4.0: Diseño de una aplicación para gestionar documentación relativa al mantenimiento de aeronaves

Autor: Laura Gonzalez Huguet

Director: José Antonio Castán Ponz

Fecha: 10 de Diciembre de 2020

Resumen

En los últimos años ha tomado fuerza la tendencia entre las aerolíneas comerciales de digitalizar la documentación de los aviones. Por ejemplo, gracias a acciones como la implementación del EFB a bordo se han podido cambiar unos 75 kg de papel por dos tabletas. Siguiendo esta línea, el objetivo de este proyecto es analizar el uso de los documentos impresos en los procedimientos relativos al mantenimiento de las aeronaves, con el propósito de desarrollar una aplicación que los sustituya. La finalidad no es sólo reducir la cantidad de papel empleada, sino también optimizar los procesos y mitigar las ineficiencias derivadas de la dependencia de registros físicos.

Primero, para entender cómo se lleva a cabo el mantenimiento de una flota comercial, se estudia el sistema de revisiones y la normativa aplicable, tanto de EASA como de ICAO. Además, para saber con qué tipo de documentación se trabaja, se hace un análisis y recopilación de los manuales y archivos más destacados.

El siguiente paso supone investigar qué programas existen actualmente, sus principales características, y el uso que hacen de ellos las aerolíneas.

Finalmente se desarrolla airDocs, una aplicación disponible para dispositivos Android y ordenadores, que permite llevar un buen control de las reparaciones e inspecciones, además de asociar cada una de ellas a la documentación pertinente, almacenada en Dropbox. Consta de cuatro módulos: ingeniería, TMA, tripulación y administración, todos ellos descritos con detalle en sus respectivos apartados. Así se tienen en cuenta todas las partes involucradas y se garantiza pleno acceso a la documentación actualizada, de modo que se reducen las posibilidades de que surja la necesidad de imprimir algún archivo.

En esencia, este proyecto contiene un estudio de los procesos que aseguran la aeronavegabilidad de una flota, y propone una alternativa que permite dar un paso más hacia la completa digitalización de la industria.

Title: MRO 4.0: Design of an app to deal with aircraft maintenance records

Author: Laura Gonzalez Huguet

Director: José Antonio Castán Ponz

Date: 10th of December 2020

Overview

In recent years, the tendency of digitizing aircraft documentation has increased among commercial airlines, with initiatives such as the EFB implementation on board, which has allowed to swap 75kg of paper for two tablets. Following this trend, the goal of this project is to analyze the use of physical records in aircraft maintenance, in order to develop an app meant to replace them. This will reduce the amount of paper used, optimize the process and mitigate the inefficiencies arisen from a paper-based system.

First, the checks system and the applicable regulation from both EASA and ICAO are studied, so as to understand how the maintenance of a fleet is carried out. Furthermore, the most important manuals and reports are analyzed with the purpose of understanding the kind of documentation involved.

The next step entails a research of the software in use, its main features and the utilization that airlines make of them.

Lastly, airDocs is designed and developed for both Android devices and computers. It enables a proper control of repairs and inspections, as well as an association to their documentation, which is stored in Dropbox. It is made up of four modules: engineering, TMA, cabin crew and administration, all explained in detail in their corresponding sections. In this way all parties involved are taken into account and are guaranteed full access to updated documentation. This decreases the possibilities of having to print any file.

Essentially, this project is a study of the processes that ensure the airworthiness of a fleet, and suggests an alternative that may bring the industry one step closer to full digitalization.

ÍNDICE

INTRODUCCIÓN	1
MRO 4.0	2
¿Por qué una aplicación para digitalizar registros?.....	2
CAPÍTULO 1. MANTENIMIENTO DE AVIONES EN LAS AEROLÍNEAS.	4
1.1. Procesos y tipos de revisiones.....	5
1.1.1. Mantenimiento de línea	6
1.1.2. Mantenimiento base	6
1.1.3. Tipos de revisiones.....	7
1.2. Departamentos y responsabilidades.....	8
1.3. Externalización como modelo de negocio.....	9
CAPÍTULO 2. REGISTROS DIGITALES	11
2.1. Regulación	12
2.1.1. ICAO	12
2.1.2. EASA	13
2.2. Documentación para el mantenimiento de aeronaves	13
CAPÍTULO 3. TECNOLOGÍA ACTUAL	17
CAPÍTULO 4. AIRDOCS: DISEÑO Y FUNCIONAMIENTO.....	20
4.1. Lenguaje y entorno de desarrollo	20
4.2. Estructuración de los datos	21
4.2.1. Tabla usuarios	22
4.2.2. Tabla aircraft.....	23
4.2.3. Tabla checks	24
4.2.4. Tabla damages.....	25
4.2.5. Conexión a la base de datos	26
4.3. Almacenamiento de archivos.....	27
4.4. Pantalla de inicio	28
4.5. Módulo de administración	29
4.6. Módulo de ingeniería.....	32
4.7. Módulo de TMA.....	35
4.8. Show Damages	36
4.8.1. Show Damages para ingeniería	37
4.8.2. Show Damages para TMA	38

4.9. Módulo de tripulación de cabina.....	39
4.10. AirDocs para ordenadores	40
CAPÍTULO 5. CONCLUSIONES	42
WEBGRAFÍA	43
ANEXO A. MANUAL DE USUARIO: VERSIÓN MÓVIL.....	44
ANEXO B. MANUAL DE USUARIO: VERSIÓN DE ESCRITORIO.....	54
ANEXO C. CÓDIGO DE AIRDOCS	66

ÍNDICE DE FIGURAS

CAPÍTULO 1. MANTENIMIENTO DE AERONAVES EN LAS AEROLÍNEAS

1.1	Resumen de la regulación aplicable al mantenimiento de aeronaves.....	4
1.2	Clasificación de los tipos de mantenimiento.....	6

CAPÍTULO 4. AIRDOCS: DISEÑO Y FUNCIONAMIENTO

4.1	Cuota de mercado dispositivos móviles.....	20
4.2	Ejemplo de estructura de carpetas de Dropbox.....	28
4.3	Pantalla de inicio.....	29
4.4	Módulo de administrador.....	30
4.5	Pantalla para visualizar usuarios.....	30
4.6	Pantalla de edición de usuarios.....	31
4.7	Pantalla para administrar la flota.....	32
4.8	Módulo de ingeniería.....	33
4.9	Pantalla para añadir y editar revisiones.....	34
4.10	Módulo de TMA.....	35
4.11	Show Damages para ingeniería.....	37
4.12	Show Damages para TMA.....	38
4.13	Módulo de tripulación.....	40
4.14	Pantalla para enviar mensajes.....	41

ÍNDICE DE TABLAS

CAPÍTULO 1. MANTENIMIENTO DE AERONAVES EN LAS AEROLÍNEAS

1.1	Tipos de revisiones.....	7
-----	--------------------------	---

CAPÍTULO 2. REGISTROS DIGITALES

2.1	Documentos oficiales.....	14
2.2	Manuales.....	15
2.3	Otros documentos.....	16

CAPÍTULO 3. TECNOLOGÍA ACTUAL

3.1	Tipos de software.....	19
-----	------------------------	----

CAPÍTULO 4. AIRDOCS: DISEÑO Y FUNCIONAMIENTO

4.1	Tipos de bases de datos.....	22
4.2	Usuarios.....	23
4.3	Aircraft.....	23
4.4	Checks.....	24
4.5	Damages.....	25

GLOSARIO

AD: Airworthiness Directive

AMC: Acceptable Means of Compliance

AOC: Air Operator Certificate

AOG: Aircraft on Ground

APU: Auxiliary Power Unit

CAMO: Continuing Airworthiness Maintenance Organization

CRS: Certificate of Release to Service

EASA: European Aviation Safety Agency

EFB: Electronic Flight Bag

FC: Flight Cycle

FH: Flight Hour

ICAO: International Civil Aviation Organization

MEL: Minimum Equipment List

MRO: Maintenance, Repair and Overhaul. Se suele referir a las empresas que ofrecen servicios de mantenimiento aeronáutico.

POA: Production Organization Approval

SB: Service Bulletin

SDT RB: Structural Damage Task, Report Back

SQL: Structured Query Language

TMA: Técnico de Mantenimiento de Aeronaves

INTRODUCCIÓN

En los últimos años, las aerolíneas comerciales han invertido dinero y esfuerzo en mejorar la experiencia del cliente, y también en facilitar el trabajo de sus empleados, mediante la progresiva digitalización de procesos. Así pues, los pasajeros pueden realizar el check-in sin necesidad de acudir al aeropuerto, a través de la web o aplicación de la aerolínea, y los pilotos ya no tienen que buscar entre infinitas páginas para encontrar la documentación que necesitan para realizar el vuelo, ya que el Electronic Flight Bag (EFB) les permite hacerlo en mucho menos tiempo. Sin embargo, se continúan generando cajas y cajas con los registros de mantenimiento de las aeronaves, que deben ser procesadas y guardadas, y en caso de que el avión en cuestión sea alquilado o vendido, la documentación tiene que entregarse con él, ya que certifica su aeronavegabilidad.

Esta dependencia del papel da lugar a grandes ineficiencias ya que, para empezar, es necesario imprimir y ordenar las instrucciones de todas las tareas que los mecánicos deben realizar. Además, hay que tener en cuenta que los ingenieros deben certificar y procesar todos estos documentos, con el inconveniente añadido de que no pueden usar motores de búsqueda como si de archivos digitales se tratase. Una práctica bastante extendida en la actualidad es escanear los papeles para trabajar con ellos desde el ordenador, pero supone una inversión de tiempo y esfuerzo adicional, y no garantiza una mayor agilidad en el procesado.

Si bien es cierto que algunos técnicos de mantenimiento de aeronaves (TMA) ya disponen de tabletas, su uso para visualizar y firmar documentación no está extendido todavía, y no se espera que el papel desaparezca del todo hasta la próxima década.

El objetivo de este trabajo es analizar las herramientas que hay en el mercado, el uso que hacen de ellas las aerolíneas y el grado de digitalización con el que trabajan. Además, se desarrolla una aplicación, airDocs, para gestionar dicha documentación, de acuerdo con la normativa vigente.

La presente memoria consta de dos partes. La primera se extiende entre los capítulos 1 y 3, y contiene toda la información necesaria para entender cómo se lleva a cabo el mantenimiento de la flota de una aerolínea. En el primer capítulo se describen los tipos de revisiones y las organizaciones involucradas, mientras que en el segundo se puede encontrar un resumen de la regulación aplicable y los tipos de documentos más importantes. El tercer capítulo es una recopilación del software más usado en la actualidad, y de sus características principales.

La segunda parte la forma el capítulo 4, y contiene una descripción de cómo se ha diseñado y cómo funciona airDocs. Incluye desde el lenguaje y entorno de desarrollo a las características de las pantallas de la aplicación, pasando por la estructuración de los datos y archivos.

Finalmente, en los anexos está el manual de usuario tanto para la versión móvil como para la versión de escritorio de airDocs, además del código implementado.

MRO 4.0

Se conoce como Cuarta Revolución Industrial o Industria 4.0 a la introducción en la industria de tecnologías como la Inteligencia Artificial, el Big Data y la interconexión masiva de dispositivos, es decir, la digitalización de las fábricas. Se trata de un proceso que acaba de empezar y se estima que durará por lo menos hasta 2030.

Si aplicamos este concepto al mantenimiento de aeronaves, MRO 4.0 engloba desde el uso de drones para realizar inspecciones hasta la completa digitalización de los datos y los procesos, dejando así de depender de la documentación en papel. También se incluye el uso de algoritmos para tratar con la información de manera automatizada y agilizar la toma de decisiones. Uno de los principales objetivos del Big Data en la industria aeronáutica es detectar patrones y predecir cuándo un componente va a fallar, para así poder realizar mantenimiento preventivo y evitar situaciones de AOG (Aircraft On Ground), los causantes de retrasos y gastos no previstos.

La aplicación desarrollada en este trabajo, airDocs, queda englobada dentro de la idea de MRO 4.0 ya que busca formar parte de esta transición eliminando los registros físicos. Ha sido diseñada pensando en todas las partes que intervienen en el proceso, de modo que todo el mundo esté incluido y sea poco probable que surja la necesidad de imprimir algún archivo.

¿Por qué una aplicación para digitalizar registros?

La idea de desarrollar una aplicación destinada a lidiar con los registros de mantenimiento de las aeronaves surge después de unas prácticas en el departamento de estructuras de la CAMO en Vueling. Durante seis meses estuve a cargo de actualizar los mapas de daños de los aviones de la flota después de las inspecciones tanto de línea como de mantenimiento mayor. Para ello debía analizar la documentación enviada por los TMAs y contrastar la información con los manuales pertinentes.

En el caso del mantenimiento de línea encontrar los archivos y datos necesarios no solía ser un problema, ya que lo habitual era trabajar con pocos daños y, en consecuencia, con un volumen pequeño de archivos. Sin embargo, cuando se trataba de mantenimiento mayor, al ser revisiones más exhaustivas, la cantidad de documentos aumentaba exponencialmente. Por ejemplo, al procesar una revisión tipo D, que se realiza cada 6 años aproximadamente, tuvimos que revisar alrededor de 10 archivos pdf de una media de 1500 páginas cada uno, con el problema añadido de que se trataba de una compilación de imágenes, ya que eran las copias en papel de los mecánicos

escaneadas. Esto en una flota de diez o veinte aviones puede ser un inconveniente, pero cuando se trata de flotas grandes como la de Vueling, que entonces contaba con algo más de 120 aviones, puede generar ineficiencias y ralentiza el proceso.

El objetivo de esta aplicación es acabar con las copias en papel de modo que toda la documentación sea digital, desde que se crean las instrucciones para los mecánicos hasta que se actualiza el mapa de daños. De esta manera se facilitará el trabajo de ingenieros, mecánicos e incluso de la tripulación.

Por un lado, el equipo de ingeniería tendrá menos problemas para procesar todos los datos después de las revisiones, ya que dispondrán de los motores de búsqueda adecuados y además no tendrán que esperar a que se escaneen todos los documentos.

Por otro lado, los mecánicos podrán disponer de tabletas con la aplicación instalada, hecho que abre un abanico de nuevas posibilidades, como por ejemplo poder adjuntar fotos y vídeos directamente a las instrucciones aportadas por el equipo de ingeniería. Además, ya no deberán cargar con todos los papeles ya que todo lo que puedan necesitar, de manuales hasta Work Packages, será accesible desde la tableta.

Por último, la tripulación podrá consultar el mapa de daños desde su dispositivo del mismo modo que consultan la cartografía o las checklists en el EFB. Así se eliminará la necesidad de tener que imprimir el mapa de daños y hacerlo llegar al avión cada vez que se realice algún cambio.

En un futuro, cuando la implementación de la aplicación se haya hecho de manera satisfactoria, es muy factible pensar en automatizar el proceso de modo que sea el sistema el que tome la información y actualice los mapas de daños en concordancia.

CAPÍTULO 1. MANTENIMIENTO DE AVIONES EN LAS AEROLÍNEAS.

De acuerdo con World Airlines Technical Operations Glossary se define el mantenimiento de una aeronave como *“aquel conjunto de acciones destinadas a restablecer o mantener un ítem en buenas condiciones para el servicio”*. Éstas incluyen reparación, modificación, revisión e inspección.

Controlar el estado de una aeronave no sólo es importante para garantizar la fiabilidad de la misma y la seguridad de los pasajeros, sino que también es un factor clave que puede contribuir a reducir o aumentar drásticamente los gastos de una aerolínea. Esto es debido a que se pueden prevenir posibles fallos o averías que podrían dejar el avión fuera de servicio, causando así potenciales retrasos e innumerables problemas. Además, un buen mantenimiento de todos los componentes ayuda a alargar su vida útil. Es por eso que todas las acciones que se toman son la suma de acciones preventivas, correctivas e inspecciones.

En este apartado se analizan los procedimientos a llevar a cabo durante la vida útil de una aeronave además de los modelos de negocio que siguen las aerolíneas, con el objetivo de entender cómo trabajan, para así posteriormente definir cómo debería ser la aplicación a desarrollar en este proyecto, y cuál sería la mejor manera de implementarla en la industria.

Para ello es necesario analizar qué establece la legislación vigente tanto de EASA como de ICAO, en cuyo caso se trata de directrices y recomendaciones. En la Figura 1.1 se muestra de manera resumida la regulación aplicable. Cabe destacar que, aunque la regulación número 1321/2014 de EASA es muy extensa, en este trabajo sólo se estudian las partes M y 145, que son las que atañen a los operadores y organizaciones de mantenimiento.

<u>EASA Reg. No.1321/2014</u>	<u>ICAO Doc. 9760</u>
<ul style="list-style-type: none"> - Parte M: Requerimientos para garantizar la aeronavegabilidad continuada. - Parte 145: Requerimientos para obtener la aprobación que autoriza a las organizaciones a realizar tareas de mantenimiento. - Parte 66: Certificación de ingenieros y mecánicos. Define los tipos de licencia. - Parte 147: Regula las escuelas de formación. - Parte T: Enmiendas y mitigación de riesgos. 	<p>Es el manual de aeronavegabilidad. Define todas las condiciones que se deben cumplir para garantizar la aeronavegabilidad de un avión. Cubre desde el diseño de la aeronave, hasta los procedimientos a seguir tanto por parte del operador, como por la organización de mantenimiento y la autoridad competente.</p>

Fig. 1.1 Resumen de la regulación aplicable al mantenimiento de aeronaves.

1.1. Procesos y tipos de revisiones

De acuerdo con el reglamento de la Comisión Europea no.1321/2014 Parte M, se requiere que el mantenimiento de las aeronaves se organice de acuerdo con el **Programa de Mantenimiento** (M.A.302). Este debe ser redactado por el operador, es decir, la organización que está en posesión de un certificado AOC, y aprobado por la autoridad competente. Todas las revisiones y enmiendas posteriores también deben ser aprobadas por dicha autoridad.

La Parte M también determina que si el Programa de Mantenimiento está basado en la lógica del Grupo de Coordinación de Mantenimiento (Maintenance Steering Group, MSG) o en el proceso de monitoreo continuado (Condition Monitoring), deberá incluir un programa de fiabilidad. La lógica MSG es un método usado para establecer intervalos de aplicación de las tareas de mantenimiento con el objetivo de evitar trabajos innecesarios y aumentar así la eficiencia. Se basa en la idea de que los sistemas que conforman un avión tienen unos mínimos de fiabilidad inherentes y que, en consecuencia, no requieren tareas de mantenimiento redundantes. Considera que estas aumentan la probabilidad de que ocurran errores humanos.

Por otro lado, el documento 9760 de la ICAO, define el Programa de Mantenimiento como el documento que describe las tareas necesarias para la operación continuada y segura de las aeronaves a las que aplica, así como la frecuencia con la que deben ser realizadas. Establece limitaciones de tiempo para reparaciones, inspecciones y revisiones del avión, los motores y todos sus componentes. Estas limitaciones se dan en horas de vuelo (FH), ciclos (FC) o días.

Todas las tareas recogidas en el Programa de Mantenimiento forman parte del **Mantenimiento Programado**, cuyo objetivo es anticipar y prevenir fallos. Generalmente se **agrupan las tareas** según la fecha en la que deben ser realizadas, para así llevarlas a cabo en una única **revisión**. De este modo podemos definir dos tipos de mantenimiento, **de línea y base**.

También se podría definir un tercer tipo de mantenimiento, el de componentes, que incluye únicamente los trabajos hechos en taller a componentes desmontados de la aeronave, como podrían ser asientos, motores o la APU. En general se lleva a cabo a la vez que el mantenimiento base y por la misma organización, aunque existen excepciones que requieren que el componente sea enviado a un taller especializado.

Además del Mantenimiento Programado, se define el **Mantenimiento No Programado** como aquel conjunto de tareas que se realizan en el momento de detectar un fallo que pone en peligro la integridad y aeronavegabilidad del avión. No suele ser una porción elevada del total de trabajos ya que es una práctica habitual posponer la reparación de ciertas averías, siempre que los manuales lo permitan, para así realizarla durante una revisión. El tiempo que puede pasar hasta la reparación varía según el impacto y tipo de desperfecto, y está establecido en la MEL (Minimum Equipment List). Estos ítems reciben el

nombre de diferidos y se recogen en una lista que va a bordo y que debe estar siempre actualizada.

En la Figura 1.2 se muestra de manera resumida los tipos de mantenimiento mencionados, mientras que en los siguientes subapartados se describen con más detalle.

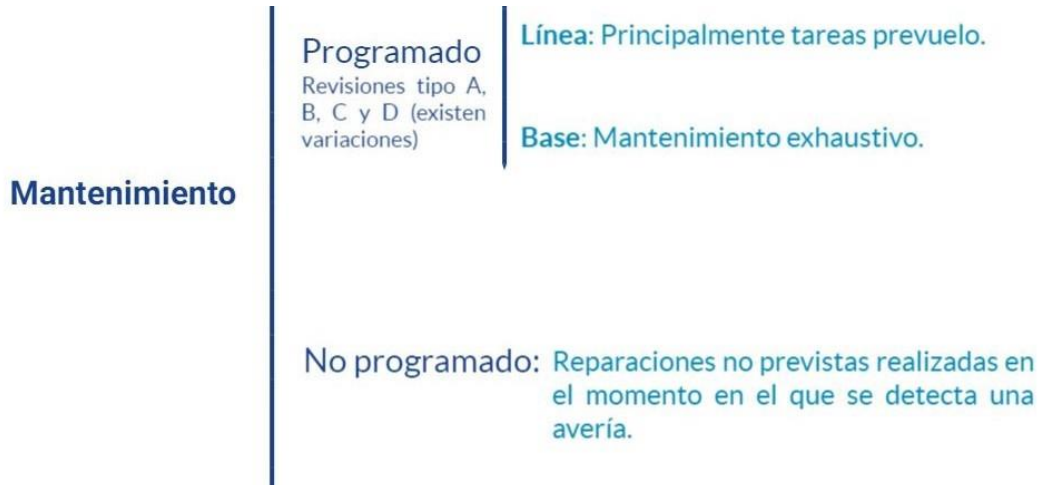


Fig. 1.2 Clasificación del mantenimiento de aeronaves.

1.1.1. Mantenimiento de línea

Según el AMC 145.A.10 de la regulación (EU) no.1321/2014, el Mantenimiento de Línea está conformado por todas aquellas tareas realizadas antes del vuelo con el objetivo de asegurar que el avión está en condiciones de realizarlo. Incluye solución de problemas, cambio de componentes, revisiones que no requieren de inspecciones en profundidad y a zonas de fácil acceso (como podrían ser la revisión diaria o la semanal) y reparaciones menores entre otros.

Se suele realizar fuera del hangar, normalmente en el mismo puesto de estacionamiento en el que se encuentra el avión antes de iniciar el rodaje hacia la pista.

1.1.2. Mantenimiento base

Según el AMC 145.A.10 (c) de la regulación (EU) no.1321/2014, todos aquellos trabajos que quedan fuera de la definición de Mantenimiento de Línea son Mantenimiento Base. Se suelen llevar a cabo dentro de un hangar, donde la atmósfera está controlada.

En este grupo se incluyen tareas a realizar con menos frecuencia y que requieren equipo especializado, ya que se trata de revisiones y reparaciones más en profundidad que las que se pueden realizar en el Mantenimiento de Línea. Se necesita mucho más tiempo para llevarlas a cabo y en algunos casos suponen desmontar y volver a montar el avión casi en su totalidad.

1.1.3. Tipos de revisiones

Las tareas de mantenimiento se suelen agrupar en revisiones según la frecuencia con la que se deben realizar. De este modo se definen los siguientes tipos de revisiones o checks:

Tabla 1.1 Tipos de revisiones

	Duración
A check	De 6 a 24 horas
B check	De 1 a 3 días
C check	De 1 a 3 semanas
D check	Alrededor de 1 mes

Los intervalos de tiempo en los que se llevan a cabo varían según la aeronave, siendo la A check la que se realiza más a menudo y la D la que menos. A modo de ejemplo, un Airbus A320 debe pasar una revisión A cada 120 días, 750 horas de vuelo o 750 ciclos, y una tipo C cada 730 días, 7500 horas de vuelo o 5000 ciclos, lo que se cumpla antes. La profundidad de las tareas a realizar aumenta con la letra, es decir, la D es la revisión más exhaustiva en la que se desmonta prácticamente todo el avión. Esta revisión también se conoce como la “gran parada” y se suele realizar cada 6 años.

Actualmente el sistema de revisiones ha adquirido mayor flexibilidad, de modo que el operador tiene mayor libertad para organizar los trabajos de mantenimiento. Por ello las revisiones B tienden a desaparecer en los aviones más modernos, quedando así sus correspondientes tareas repartidas entre los otros tipos. Además, aparecen varios tipos de revisión C, denominadas C1, C2..., que pueden englobar también tareas de la D check.

A los tipos de revisión descritos anteriormente hay que añadir las revisiones diarias (o daily) y las semanales (o weekly). En ellas se comprueban los puntos críticos tanto del interior como del exterior de las aeronaves para asegurar que están listas para efectuar un vuelo seguro.

1.2. Departamentos y responsabilidades

En una aerolínea comercial, el departamento a cargo del mantenimiento de la flota es la **CAMO**, Continuing Airworthiness Maintenance Organisation. De acuerdo con la **regulación EU no.1321/2014 Parte M Subparte G**, estas organizaciones requieren la aprobación de la autoridad competente. Para obtener dicha aprobación deben redactar el **CAME**, Continuing Airworthiness Maintenance Exposition. Este documento incluye un organigrama del departamento con las responsabilidades de cada uno de sus miembros, los procedimientos a seguir para garantizar la aeronavegabilidad de la flota, las acciones llevadas a cabo por los encargados del control de calidad y todas las organizaciones externas contratadas para llevar a cabo el mantenimiento (en caso de que las haya, el mantenimiento puede hacerlo la misma aerolínea si está autorizada para ello).

Una de las funciones principales de la CAMO es hacer de enlace entre la aerolínea y las autoridades, así como también con los proveedores de servicios de mantenimiento. Además, los miembros de su equipo de ingeniería deben ser conocedores de los distintos manuales para poder determinar la mejor manera de proceder ante las posibles averías. También deben estudiar las Directivas de Aeronavegabilidad (AD) y los Service Bulletins (SB) con el objetivo de determinar el impacto que tienen en la flota y qué acciones deben tomarse. El control de calidad y las auditorías también se encuentran entre sus cometidos y, aunque mayoritariamente se trata de trabajo de oficina, es posible que de vez en cuando se deban realizar estudios en las aeronaves mismas. Por último, en la CAMO debe haber personal destinado al procesado y almacenaje de los registros de mantenimiento siguiendo las pautas que marca la legislación vigente.

Otro de los actores principales son las organizaciones que cuentan con la aprobación contemplada en la **regulación EU no.1321/2014 Parte 145**, que las convierte en aptas para "*otorgar o mantener una aprobación de mantenimiento de aeronaves y elementos*". En otras palabras, están capacitadas para certificar que una reparación se ha llevado a cabo de manera correcta y que la aeronave está lista para ser puesta en servicio con total seguridad. Entre los requisitos para obtener dicha aprobación se encuentran la disponibilidad de unas instalaciones adecuadas tanto para realizar las tareas de mantenimiento (hangares y talleres) como las de certificación (oficinas), además de espacios de almacenamiento seguro de herramientas, material y equipo. En cuanto a personal, es necesario designar un responsable, y se debe demostrar que se dispone de empleados suficientes para llevar a cabo las tareas de mantenimiento y control de calidad de acuerdo con lo dispuesto en la legislación. Además, todos los trabajadores deben estar debidamente cualificados de acuerdo con las funciones que desempeñen y la normativa vigente.

Así como la CAMO debe redactar el CAME, las organizaciones Parte 145 deben redactar la **Memoria de la Organización de Mantenimiento** que describe las características de los trabajos para los que se solicita la aprobación. Debe incluir un organigrama con las responsabilidades de cada

miembro, la política de calidad y seguridad, una lista del personal certificador y de apoyo, y una descripción de los procedimientos establecidos.

Tal y como se explica en el siguiente apartado, las aerolíneas tienden a externalizar las tareas de mantenimiento a organizaciones que cuentan con la aprobación de la Parte 145, e incluso hay algunas que externalizan funciones de la CAMO. Si bien es cierto que se puede externalizar todo aquello relativo a la Parte 145, actualmente la normativa sólo contempla que se puedan externalizar algunas de las tareas de la CAMO, entre las cuales se incluye la conservación de registros, aunque se está estudiando que en un futuro se permita subcontratar más trabajos.

1.3. Externalización como modelo de negocio.

El 24 de octubre de 1978 se aprobó en Estados Unidos la Ley de Desregulación de Aerolíneas, cuyo objetivo era eliminar los controles que el gobierno ejercía sobre la aviación civil. De este modo se abrió la competencia y se crearon oportunidades para que nuevas aerolíneas pudieran entrar en el mercado. Quince años más tarde, el 1 de enero de 1993 entró en vigor el tercer paquete de medidas de liberalización del transporte aéreo en Europa, imitando así el modelo estadounidense.

Estas nuevas leyes supusieron un gran cambio en la industria, ya que se multiplicó el número de pasajeros, y además propició la aparición de las aerolíneas de bajo coste. El modelo de negocio que estas seguían, y siguen en la actualidad, se basaba en minimizar todos los gastos posibles mediante la externalización de muchas actividades que hasta entonces realizaba el operador por sí mismo. Esto condujo a que el mantenimiento de los aviones fuese llevado a cabo por terceros, de modo que la aerolínea se podía centrar en el núcleo de su negocio, llevar a los usuarios de un sitio a otro. Fue así como se rompió con la idea de que el mantenimiento de los aviones simplemente era parte del negocio, y empezó a surgir el MRO como un segmento viable de la industria.

Hoy en día la mayoría de aerolíneas externalizan al menos una parte del mantenimiento. Según la cantidad de trabajo subcontratado las podemos dividir en tres modelos:

- En el primer modelo, la **mayor parte de las tareas están realizadas por la aerolínea misma** salvo una pequeña parte que se subcontrata, como podría ser el mantenimiento de línea en aeropuertos a los que no se vuela a menudo o donde no sale rentable tener mecánicos. Un ejemplo sería American Airlines.
- En el segundo modelo, el grueso de los trabajos los realiza una **empresa subsidiaria** que suele tener como principal cliente la aerolínea matriz. Si la capacidad lo permite, también puede ofrecer servicios a

terceros. Igual que en el modelo anterior, una pequeña parte de las tareas se subcontrata. Iberia y Lufthansa son dos ejemplos.

- El tercer modelo es aquel que **externaliza todas las tareas de mantenimiento**. Un ejemplo sería Vueling, ya que el mantenimiento está a cargo de Iberia Maintenance entre otros.

Existe una gran variedad de motivos que puede llevar a las aerolíneas a tomar la decisión de subcontratar parte o incluso la totalidad de este servicio. Por un lado, están las startups y aerolíneas pequeñas que no tienen el capital necesario ni para la inversión inicial ni para mantener las instalaciones, el equipamiento específico que se requiere y el personal. Por otro están los operadores con flotas más grandes que, a pesar de disponer de hangar y mecánicos, no tienen la capacidad suficiente como para realizar todas las tareas que requieren sus aviones, de modo que deciden subcontratar una parte.

CAPÍTULO 2. REGISTROS DIGITALES

Con el objetivo de garantizar el cumplimiento de la legislación y asegurar que un avión es aeronavegable, es decir, satisface todos los requisitos para operar de manera segura, las autoridades requieren el almacenamiento de los documentos derivados de las tareas de mantenimiento.

A lo largo de los años ha ido cambiando la manera en la que las aerolíneas guardan dichos registros. Al principio se llenaban cajas y cajas de papeles que tenían que ser traspasadas de un operador a otro cuando la aeronave en cuestión se vendía o se alquilaba. El equipo de ingeniería tenía que buscar entre ellas todo lo necesario para iniciar el programa de mantenimiento, asegurar que se cumplían todas las directivas (ADs) y, en definitiva, adaptar el avión a su operativa.

En los últimos años se ha convertido en una práctica habitual escanear todos estos documentos para guardarlos en servidores externos. De este modo no solo se reduce el espacio físico que ocupa, sino que también se facilita el acceso a ellos ya que, por ejemplo, dos ingenieros pueden examinar el mismo documento a la vez, cada uno desde su ordenador.

Si bien esto supone una mejora respecto al método anterior, sigue distando de ser eficiente, principalmente porque un documento escaneado no deja de ser una imagen con fragmentos escritos a mano por los mecánicos, que no permite el uso de motores de búsqueda. Esto puede llevar en muchos casos a archivos de miles de páginas entre las que el equipo de ingeniería debe buscar a mano la información que necesita.

Así pues, el siguiente paso a seguir es la digitalización total de estos documentos, de manera que no sea necesario llegar a imprimirlos y que todo el personal implicado pueda acceder a ellos en cualquier momento. Esto aumentará la eficiencia y reducirá el tiempo invertido en todos los procesos, desde las reparaciones hasta las certificaciones. Además, esta nueva manera de proceder hará que todo el sistema sea más fácilmente escalable y se pueda adaptar rápidamente a cambios en las flotas y a un mayor volumen de datos.

De acuerdo con esta nueva tendencia, la ICAO ha redactado una guía para la aceptación de los registros digitales, incluida en su documento 9760, el Manual de Aeronavegabilidad. Está previsto que sea aplicable a partir de noviembre de 2020. Actualmente ya hay estados que permiten el uso de registros de mantenimiento electrónicos, EAMR por sus siglas en inglés, pero su uso no está todavía muy extendido. Entre las dificultades que presenta la implementación de los EAMR destaca la firma electrónica, la integridad y seguridad de los documentos, y la migración de un sistema a otro. El objetivo de la guía de la ICAO es tratar estos asuntos y proponer una solución lo más estandarizada posible para evitar así problemas de compatibilidad entre países.

2.1. Regulación

En este apartado se resumen las normas y directrices publicadas respecto a los registros electrónicos. Esto incluye principalmente la guía ya mencionada de la ICAO puesto que EASA no ha publicado por ahora ninguna regulación que trate específicamente los EAMR. Aun así, también se estudian los requisitos genéricos que establece EASA en cuanto al almacenamiento de registros. Es necesario recordar que la ICAO publica recomendaciones y directrices para que los estados miembros las adapten y conviertan en regulaciones.

2.1.1. ICAO

La ICAO establece que un sistema de conservación de registros electrónicos es aquel en que la información es introducida, compilada, guardada y recuperada de manera totalmente electrónica, sustituyendo así las copias en papel. Cualquier sistema que trabaje con EAMR debe cumplir ciertos requisitos establecidos por la autoridad competente y ser aprobado por la misma. Entre estos requisitos destaca el acceso sin restricciones para la realización de auditorías por parte de la autoridad, además de la capacidad de hacer llegar copias en papel de ciertos documentos a la misma. Asimismo, su implementación debe ir precedida de una formación adecuada para todo el personal.

Para una correcta autenticación de los usuarios se sugieren métodos como contraseñas, número PIN o tarjetas personales. El nivel de comprobación de que el usuario es quién dice ser deberá ser proporcional al tipo de tarea al que se le está autorizando. Los administradores del sistema deben ser responsables en todo momento de las autorizaciones que se les da a los usuarios, para así evitar un mal uso.

En cuanto a la firma electrónica, para que sea aceptable debe garantizar la identidad del usuario de la misma manera que lo haría una firma convencional. Incluye un sonido, símbolo, marca visible o proceso asociado a un documento y llevado a cabo por un usuario con el objetivo de firmarlo. Identifica y autentica electrónicamente al usuario en cuestión que introduce, verifica o audita registros digitales. La firma electrónica debe ser única y segura, y debe aportar trazabilidad.

Por otro lado, en lo que a seguridad e integridad del sistema se refiere, se deben definir un conjunto de políticas y procedimientos que respalden el sistema y los archivos que lo conforman. Se debe proteger tanto su integridad física como la información confidencial que pueda contener, y se deben evitar alteraciones no autorizadas. Además, se deben programar auditorías periódicas que comprueben que se mantiene la integridad del sistema. También se pueden llevar a cabo auditorías no previstas si se sospecha que la integridad del sistema ha sido comprometida. Es trabajo del operador

establecer los procedimientos para la transferencia segura de archivos a otro operador, organización o autoridad.

2.1.2. EASA

La normativa establece que la organización de mantenimiento sujeta a la parte M de la Regulación no.1321/2014 deberá guardar registros con los detalles de las tareas de mantenimiento llevadas a cabo, entre las que se incluyen reparaciones y modificaciones. Además, será necesario conservar todos aquellos documentos relacionados con la expedición de certificados de revisión de la aeronavegabilidad.

Tanto el operador como la organización de mantenimiento deberán disponer de copias de toda la información mencionada y conservarla durante tres años desde el momento en el que la aeronave o componente en cuestión es considerado apto para el servicio después de los trabajos. Además, cuando el avión sea retirado del servicio de manera definitiva, la documentación se deberá mantener durante los dos años siguientes. Todos los registros deben guardarse de manera segura para evitar daños, alteraciones no autorizadas o robos.

Por otro lado, las organizaciones que ofrecen servicios de mantenimiento (MRO), y que están sujetas a la parte 145 de la Regulación no.1321/2014, también deben guardar registros detallados de todos los trabajos de mantenimiento realizados. Como mínimo deben poder probar el cumplimiento de los requisitos establecidos para la expedición del certificado de aptitud para el servicio. Se conservarán dichos documentos hasta tres años después de la nueva puesta en servicio, siempre protegidos frente a daños, alteraciones no autorizadas y robos. Cuando una organización cese su actividad se distribuirán los documentos que se conserven de los tres últimos años al propietario más reciente de la aeronave o componente en cuestión, o se almacenarán como establezca la autoridad competente.

2.2. Documentación para el mantenimiento de aeronaves

El documento 9760 de la ICAO establece que los registros de mantenimiento deben dar una imagen general del estado de la aeronave, y deben demostrar que el mantenimiento se está llevando a cabo de manera correcta. El nivel de detalle, contenido, forma y soporte debe ser aceptado por la autoridad correspondiente.

Los registros deben indicar el estado de aplicación de los requisitos e instrucciones relativas a la aeronavegabilidad del avión, además del cumplimiento del programa de mantenimiento. Así pues, la documentación debe probar que se han llevado a cabo todas las reparaciones pertinentes y que se ha retirado del servicio cualquier componente que haya llegado al fin de

su vida útil. Todo ello debe demostrar la validez del Certificado de Aeronavegabilidad de la aeronave.

El personal de mantenimiento debe documentar todos los trabajos realizados al avión y sus componentes. Entre la información a incluir se encuentra la fecha de realización, la identificación de la aeronave o el componente sobre el que se trabaja (Part Number, PN, y Serial Number, SN), su antigüedad y el tiempo transcurrido desde la última reparación (en FH, FC o días), detalles de las tareas realizadas y referencias a los manuales aprobados en los que se basan, y el CRS (Certificate of Release to Service) junto con el nombre e identificación del personal de certificación. De la misma manera, se deberán documentar todos aquellos defectos o averías encontrados en las inspecciones, y las acciones correctivas que se llevan a cabo.

Además, según la Regulación Europea 1321/2014, se entiende como datos de mantenimiento cualquier documento aplicable emitido por la autoridad competente, AD aplicable e instrucciones o información de mantenimiento emitidas por organizaciones autorizadas. Deben ser accesibles y estar actualizados en todo momento y, con el objetivo de facilitar los procesos, se deben organizar por tareas en tarjetas o fichas de trabajo (JC, Job Cards).

A continuación, se detallan los documentos y manuales más importantes. Se muestran divididos en documentos oficiales (emitidos por la autoridad), manuales y otros documentos.

Tabla 2.1 Documentos oficiales.

Airworthiness Directive (AD)	Recoge mejoras o acciones correctivas a implementar en aeronaves. Son de obligado cumplimiento y tienen el objetivo de corregir o prevenir averías o eventos que puedan poner en peligro su integridad. El organismo encargado de emitir y aprobar una AD es la autoridad de aviación correspondiente, EASA en el caso europeo.
Service Bulletin (SB)	Es un documento emitido por los fabricantes de aeronaves, motores o cualquiera de sus componentes, y contiene información detallada sobre modificaciones aplicables a los mismos. Pueden ser mejoras de producto o pueden estar relacionadas con la seguridad operativa. En este último caso se emiten como "Alert SB" y deben ir acompañados de una AD emitida por la autoridad competente.
Formulario EASA 1	Es un certificado que acredita que un producto, componente o equipo ha sido fabricado de acuerdo con los datos de diseño establecidos y, por lo tanto, es aeronavegable. Se trata de un certificado estandarizado cuya plantilla es publicada por EASA, aunque lo puede emitir cualquier organización con un certificado POA.

Repair Design Approval Sheet (RDAS)	Es un documento emitido por el fabricante de aeronaves, por ejemplo Airbus, que contiene instrucciones para la reparación de un daño no recogido en el SRM (Structural Repair Manual).
--	--

Tabla 2.2 Manuales

Structural Repair Manual (SRM)	Es un documento publicado por el fabricante de la aeronave que describe los procedimientos a seguir para reparar daños estructurales. Está organizado de acuerdo con el sistema ATA 100, que divide los sistemas de la aeronave en capítulos. Los estructurales van del capítulo 51 al 57. Si no existe una reparación en el manual para cierto tipo de daño, se debe contactar con el fabricante para que estudie el caso y la facilite.
Aircraft Maintenance Manual (AMM)	Es un documento publicado por el fabricante de la aeronave que recoge las tareas de mantenimiento que deben llevarse a cabo en un avión. No incluye reparaciones y modificaciones estructurales, ya que éstas están recogidas en el SRM. También está organizado según el sistema ATA 100.
Minimum Equipment List (MEL)	Se trata de un documento que recoge las limitaciones operativas aplicables cuando algún componente no funciona o está defectuoso. Esto permite al personal decidir si posponer la reparación y mantener el avión en servicio bajo las condiciones especificadas, o si se debe quedar en AOG (Aircraft On Ground). Según la categoría del daño, este debe ser reparado en mayor o menor tiempo. A estos ítems pendientes de reparar se les conoce como “diferidos”, y están recogidos en una lista que va a bordo de la aeronave. La MEL debe ser aprobada por la autoridad correspondiente del país del operador, y es una adaptación de la MMEL (Master Minimum Equipment List) publicada por el fabricante del avión. La MEL es siempre más restrictiva que la MMEL.
Configuration Deviation List (CDL)	Es una lista que recoge las limitaciones operativas derivadas de la falta de alguna parte externa no estructural del avión. No se debe confundir con la MEL, ya que ésta cubre el mal funcionamiento de los sistemas, mientras que la CDL se ocupa de partes que no están, como podría ser un fairing o panel.
Illustrated Parts Catalogue (IPC)	Es un documento complementario que describe de forma detallada todos los componentes de una aeronave.

Tabla 2.3 Otros documentos.

Structural Damage Repair (SDR)	Se trata de un documento que recoge información detallada sobre un daño estructural. Se debe cumplimentar cada vez que se encuentre uno y debe ser enviado al equipo de ingeniería para su evaluación. Como es un documento interno de la aerolínea que da trazabilidad a todas las reparaciones estructurales, el formato y la información que se incluye depende de ellas. En general se requiere la posición del daño, su tamaño, categoría y referencias a las reparaciones incluidas en los manuales.
WorkOrder (WO)	Orden de trabajo que contiene las instrucciones a seguir para llevar a cabo una reparación o inspección. La genera el equipo de ingeniería y debe ser realizada por los TMAs.
Mapa de daños	Es un documento que recoge los daños y reparaciones de un avión. Se suelen incluir reparaciones estructurales realizadas en la parte externa de la aeronave, pero también pueden tenerse en cuenta elementos internos como los paneles del suelo de la bodega. Se trata de un documento interno de la aerolínea por lo que el formato puede variar de una a otra. El mapa de daños debe mantenerse siempre actualizado y la copia más reciente debe llevarse a bordo.
Entradas parte de vuelo (Logbook)	Recoge los detalles sobre el estado del mantenimiento de la aeronave. Cuando un mecánico o miembro de la tripulación encuentran un daño o defecto deben incluirlo en el parte de vuelo. Debe estar siempre a bordo.
Hold Item List (HIL)	Se trata de una lista que recoge todos los desperfectos pendientes de reparar. A estos elementos se les llama diferidos. El intervalo de tiempo que puede pasar entre que se encuentra una avería y se repara está establecido en la MEL, y variará según el impacto del mismo.

CAPÍTULO 3. TECNOLOGÍA ACTUAL

La idea de crear un programa que permita gestionar los datos y tareas relativas al mantenimiento de aeronaves no es nueva, sino que surgió a finales de los años 80 con software como el desarrollado por Crossair (predecesor de Swiss Airlines) que años después sentaría las bases de AMOS, una de las herramientas más usadas en la industria.

A lo largo de los años han ido surgiendo nuevos programas que cubren distintas necesidades, todos ellos con el objetivo de facilitar tanto a aerolíneas como a talleres y empresas MRO, la gestión de todos los datos requeridos por la autoridad para asegurar la aeronavegabilidad de las flotas.

A continuación, se detallan algunos de los programas más usados en la actualidad, con el objetivo de ver qué necesidades cubrirá la aplicación desarrollada en este proyecto.

- **AMOS:** Actualmente desarrollado por Swiss Aviation Software (spin off de Swiss International Air Lines, miembro de Lufthansa Group) y en constante evolución. Ofrece AMOSdesktop, aplicación de escritorio, y AMOSmobile (desde 2016) para tabletas y móviles. Permite trabajar con cualquier tipo de avión y es adecuado para flotas de muchos tamaños. Ofrece ocho módulos centrales: Material Management, Engineering, Planning, Production, Maintenance Control, Component Maintenance y Commercial and Quality Assurance. Además, hay dos módulos extra para gestionar Recursos Humanos y Finanzas, y otros tantos opcionales destinados a, por ejemplo, el mantenimiento de simuladores de vuelo. También ofrece la posibilidad de usar la firma electrónica (e-signature).
- **RAMCO Aviation Suite:** Parecido a AMOS con la diferencia de que cubre tanto aviación civil como defensa y además tiene un módulo dedicado al departamento de operaciones vuelo. Por otro lado, su desarrollo no está enfocado a los registros digitales sino a los bots, la Inteligencia Artificial y el Machine Learning.
- **TRAX eMRO:** También parecido a AMOS ya que tiene la misma estructura modular. Se trata de una aplicación para móviles y tabletas, con funcionalidades como el control de tareas, el control de stock... Incluye la posibilidad de implementar la firma digital. Además, otra característica destacable es que no necesita conexión a internet continua para trabajar. En caso de que esta se pierda, el programa se actualizará automáticamente cuando la recupere.
- **Quantum MX:** Desarrollado por FLY Online Tools. A diferencia de los dos anteriores no se trata de una aplicación de escritorio sino de una aplicación web. Además, si RAMCO y AMOS están más orientados a la CAMO de las aerolíneas, Quantum MX ha sido creado pensando en los talleres, por lo que quiere ser un punto de encuentro entre operadores y MRO externos.

- **AVIATAR:** Desarrollado por Lufthansa Technik junto con Swiss AS. Esta aplicación no gestiona registros, sino que se encarga de recopilar datos sobre el estado de la flota, los procesa y los muestra al equipo de ingeniería. De esta manera cuando hay una avería el personal puede saber dónde está el avión en cuestión y a dónde se dirige, permitiendo así que una vez aterrice esté todo preparado para solucionar el problema, y tener la aeronave parada el mínimo tiempo posible. También da una visión general de la flota y pone los fallos en contexto para detectar si, por ejemplo, conciernen a un tipo de avión concreto o a una fase de vuelo concreta.
- **FLYdocs:** Desarrollado por Lufthansa Technik se trata de una aplicación web que ordena y procesa los documentos escaneados derivados del mantenimiento de aeronaves. Su principal objetivo es minimizar y optimizar el tiempo de auditoría. Además, es capaz de crear un perfil para cada uno de los aviones de la flota para así facilitar las transiciones cuando el operador decida venderlos o cuando finalice el alquiler.
- **STREAM - Aerdata:** Desarrollado por Boeing. Permite visualizar y organizar archivos escaneados. También sirve como motor de búsqueda.
- **Skywise:** Desarrollado por Airbus y Palantir Technologies, se trata de una plataforma de open data que permite gestionar los datos de mantenimiento de una flota y obtener una visión general de la misma. El objetivo es ayudar a las aerolíneas a optimizar su estrategia de mantenimiento con el foco en el mantenimiento preventivo.
- **AirVault:** Desarrollado por GE Aviation. Es muy parecido a los anteriores. Es capaz de generar paneles de control (dashboard), reportes y estadísticas de varios tipos. Como valor añadido cabe destacar que es compatible con la e-signature de AMOS.
- **Maintenix:** Cuenta con distintas ediciones según el rol y las funciones a desempeñar. Entre ellas se encuentran la Operator Edition, Fleet Planner, Line Planner y eLogbook. La Operator Edition es la que se encarga de cubrir la parte de mantenimiento. También ofrece la posibilidad de implementar la firma digital.
- **SAP:** Aunque se trata de software genérico de gestión de recursos empresariales (ERP, Enterprise Resource Planning), también lo usan algunas aerolíneas (Iberia, por ejemplo) generalmente como complemento a otros tipos de programas.

De todas las herramientas analizadas se puede ver que, aunque hay algunas que ofrecen la posibilidad de implementar la firma digital, ninguna tiene como objetivo ayudar a las aerolíneas y empresas MRO a digitalizar del todo sus procesos. Todas ellas se pueden dividir en los tres grandes grupos que se muestran en la siguiente tabla:

Tabla 3.1 Tipos de software.

Software de gestión	Todos aquellos programas que facilitan el control de procesos como por ejemplo AMOS.
Software de documentación	Son programas que sirven para almacenar copias digitales de los registros en papel. Un ejemplo sería FLYdocs.
Software de análisis	Aquellos programas que procesan los datos y son capaces de generar gráficos, reportes... Un ejemplo sería Skywise.

Con el objetivo de ser todo lo accesible posible y de facilitar su implantación, la aplicación desarrollada en este proyecto no pretende sustituir a los programas que ya usan las aerolíneas, sino que busca ser un complemento a ellos. De este modo la transición será menos costosa y no requerirá de mucha formación por parte de los trabajadores.

Así pues, la aplicación servirá de puente entre el equipo de ingeniería, TMAs y también tripulaciones, permitiendo compartir la documentación necesaria sin necesidad de imprimirla y enviarla.

CAPÍTULO 4. AirDocs: DISEÑO Y FUNCIONAMIENTO

En este apartado se expone cómo se ha desarrollado la aplicación y cómo funciona. También se detallan y justifican todas las decisiones que se han ido tomando en el proceso. Cabe destacar que, aunque el principal objetivo de este trabajo es crear una aplicación móvil, también se ofrece una versión de escritorio. De esta manera se busca facilitar el trabajo del personal que se pueda beneficiar del uso de una tableta, teniendo también en cuenta a los trabajadores a los que les sea más conveniente usar un ordenador.

4.1. Lenguaje y entorno de desarrollo

Antes de empezar a diseñar la aplicación, es necesario ver qué opciones hay en cuanto a lenguajes de programación y entornos de desarrollo, ya que son factores claves que marcarán cómo será el resultado final.

Lo primero a tener en cuenta es cuáles son los sistemas operativos más usados hoy en día. En la Figura 4.1 se puede ver que Android domina el mercado de smartphones tanto a nivel europeo como mundial, y en cuanto a las tabletas también prevalece a nivel europeo, aunque con menos fuerza.

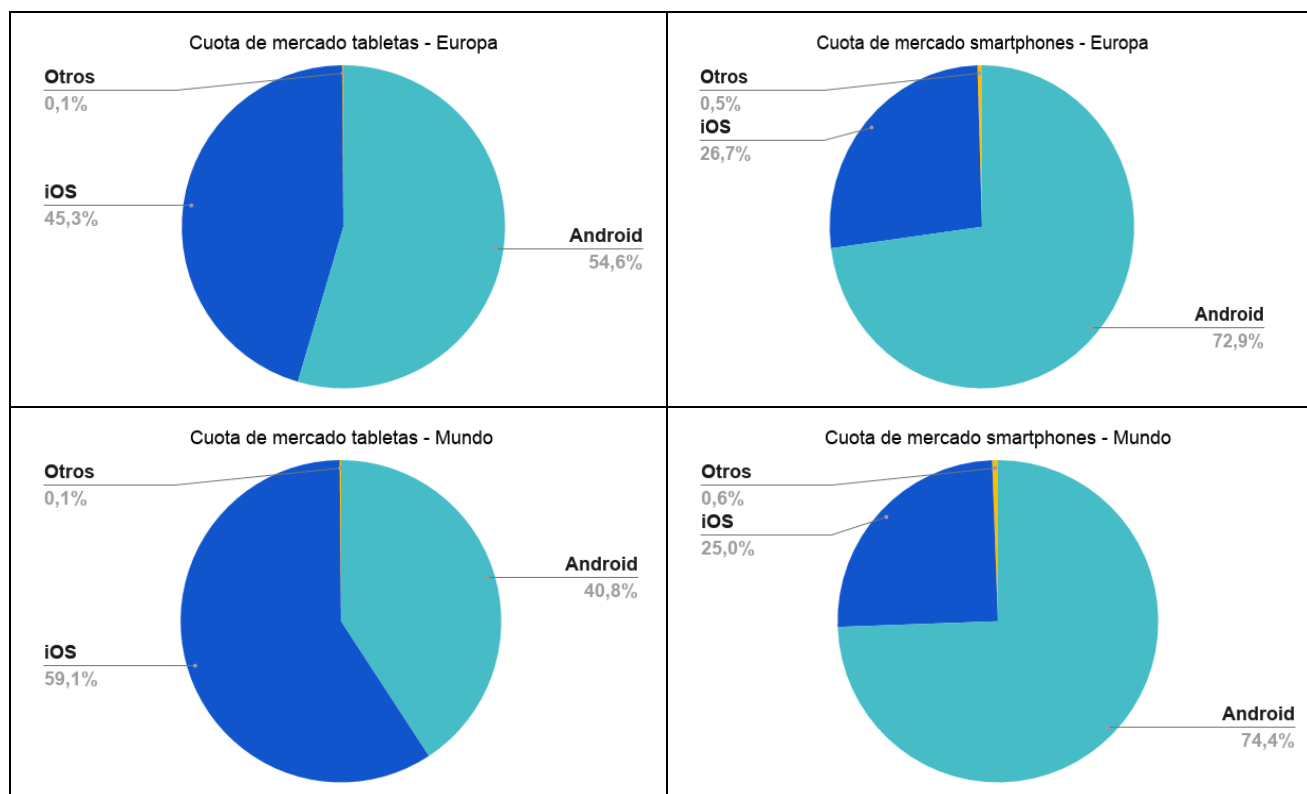


Fig. 4.1 Cuota de mercado dispositivos móviles. Fuente: StatCounter.

Es por eso que la idea inicial era desarrollar airDocs para Android. Sin embargo, después de investigar distintas opciones de plataformas de desarrollo surgió la posibilidad de crear la aplicación también para iOS, pero sin tener que volver a desarrollar todo el código. Esto es gracias a Xamarin, una plataforma de código abierto que permite desarrollar en C# aplicaciones para Android, iOS, Windows Phone y aplicaciones universales de Windows. Entre sus características principales destaca el hecho de que todas las aplicaciones creadas son nativas y que el código se comparte entre todas las plataformas, es decir, que se usa el mismo código para aplicaciones en distintos sistemas operativos, eliminando así la necesidad de tener que duplicar el trabajo para poder disponer de la aplicación en Android e iOS. También ofrece la posibilidad de generar código nativo para así personalizar la aplicación según el sistema operativo en caso de que sea necesario.

En cuanto a entorno de desarrollo, se usa Visual Studio 2019 principalmente porque es compatible con C# y permite instalar Xamarin. Además, incluye un simulador de dispositivos móviles, ideal para probar la aplicación mientras se programa sin tener que disponer de un dispositivo físico.

Aunque airDocs ha sido desarrollada para dos sistemas operativos diferentes, por ahora sólo se puede disponer de la versión en Android porque no se cuenta con los medios suficientes para probar y distribuir la versión en iOS.

La versión de escritorio de airDocs también está desarrollada en Visual Studio 2019, con Windows Forms y C#. Como que el lenguaje de programación es el mismo, se puede aprovechar la mayoría de la parte central del código generado para la aplicación. Así pues, las clases que estructuran los datos son exactamente iguales en todas las versiones del programa. Sin embargo, el código que se ejecuta en cada una de las pantallas sí que difiere un poco entre la versión móvil y la de ordenador, debido principalmente a que la naturaleza de ambos sistemas y entornos es diferente.

4.2. Estructuración de los datos

La función principal de airDocs es permitir que todos los usuarios puedan acceder de manera simultánea a la información, siempre actualizada, sobre el mantenimiento de la flota. Esto crea la necesidad de que los registros estén almacenados en una base de datos que se pueda consultar de manera remota. Así pues, lo primero que hay que considerar es cómo crear y gestionar dicha base de datos, y cómo accederá a ella la aplicación.

Actualmente existe una gran variedad de sistemas de gestión de bases de datos, y se pueden clasificar en dos grandes grupos: las relacionales y las no relacionales.

Tabla 4.1 Tipos de bases de datos.

Relacionales (SQL)	Se basan en la organización de la información en tablas , que se relacionan entre ellas mediante identificadores. Cada columna de la tabla es un campo y cada fila un registro. Entre sus ventajas destacan una buena definición de estándares gracias a las consultas SQL, y la eliminación de duplicidades, mientras que sus principales inconvenientes son la dificultad para realizar cambios en la estructura y los problemas de crecimiento. Ejemplos de sistemas de gestión de este tipo de bases de datos son mySQL, Oracle o SQL Server.
No relacionales (noSQL)	Su principal diferencia es que no establecen relaciones entre los datos y que usan un lenguaje diferente a las relacionales. Son útiles cuando no se tiene un esquema o idea exacta de lo que se quiere almacenar, o cuando se necesita guardar grandes cantidades de información para analizarla en un futuro (por ejemplo, aplicaciones de Big Data). Sus puntos fuertes son el manejo de grandes cantidades de información y la escalabilidad, mientras que entre sus inconvenientes está la baja estandarización y la falta de interfaz gráfica de muchas herramientas. Algunos ejemplos de sistemas de gestión son MongoDB, Redis o Elasticsearch.

Para este proyecto, como se conocen los datos a almacenar y conviene hacerlo de manera ordenada, se ha optado por una base de datos relacional gestionada con mySQL. Una vez estén las tablas definidas, es poco probable que sea necesario cambiar su estructura, sólo hará falta ir añadiendo datos. Además, una base de datos no relacional no es necesaria para esta aplicación porque el volumen de datos con el que trabaja airDocs es relativamente pequeño, teniendo en cuenta que este tipo de bases de datos se recomienda para aplicaciones de Big Data.

Una vez elegida una base de datos tipo relacional, es necesario estructurar los datos. Es por eso que se crean cuatro tablas: usuarios, aircraft, checks y damages. Cabe destacar que la estructura de las clases definidas para la aplicación, y de los modelos de la API REST (ver apartado 4.2.5) es la misma que la de las tablas de la base de datos que se muestran a continuación.

4.2.1. Tabla usuarios

Esta tabla contiene todos los datos de los usuarios que están autorizados para utilizar airDocs. Los campos incluidos son los siguientes:

Tabla 4.2 Usuarios

fullName	Nombre completo del usuario.
userName	Nombre de usuario, para iniciar sesión.
password	Contraseña. En caso de pérdida, por motivos de seguridad, sólo los administradores del sistema pueden cambiarla
email	Correo electrónico al que se enviarán notificaciones del sistema y que desde el que se enviarán las consultas que el usuario haga a través de la aplicación.
position	Rol que desempeña el usuario. Puede ser admin (administrador/a), eng camo (ingeniero/a de la CAMO), eng mro (ingeniero/a del MRO), eng all (ingeniero/a con total acceso), tma (mecánico/a) o crew (miembro de la tripulación)
blocked	Parámetro que sirve para denegar el acceso al usuario de manera temporal si, por ejemplo, se detecta un uso indebido de una cuenta.
PIN	Número de identificación necesario para firmar cualquier acción o documento.

A modo de curiosidad, destacar que esta tabla recibe el nombre de “usuarios” y no “users”, en inglés como las demás, porque daba errores al crear la API REST. Fue por eso que se optó por crear un modelo de datos de nuevo con un nombre diferente.

4.2.2. Tabla aircraft

La tabla aircraft contiene la información más relevante sobre los aviones de la flota, en la que se incluye:

Tabla 4.3 Aircraft

flAcftReg	Matrícula de la aeronave.
manufDate	Fecha de fabricación.
acftType	Tipo de aeronave, por ejemplo A320neo.

4.2.3. Tabla checks

La tabla checks guarda la información sobre las revisiones, tanto las ya cerradas como las pendientes. Los campos que contiene son los siguientes:

Tabla 4.4 Checks

chAcftReg	Matrícula de la aeronave.
chType	Tipo de revisión (A,B,C...)
iniDate	Fecha de inicio.
endDate	Fecha fin.
chID	Identificador de la revisión. Se trata de un parámetro generado automáticamente por el sistema, y sigue un formato estándar: matrícula_tipo_fecha de inicio. Por ejemplo: EC-AAA_C_20201022
CRSdate	Fecha de puesta en servicio.
chStatus	Estado de la revisión. A elegir entre una lista predeterminada.
chComments	Comentarios o información relevante que no tiene cabida en ningún otro campo.
assignedTo	TMA's a los que se les asigna la revisión en cuestión.
allProcessed	Indica si la documentación ya ha sido totalmente revisada o no. Sirve para cerrar la revisión.

Cuando se crea o edita una revisión, se puede elegir entre uno de los siguientes estados:

- **Planned:** Programada para empezar un día concreto.
- **On-going:** Se está llevando a cabo.
- **Closed:** Cerrada. Todo el trabajo físico, certificaciones y revisiones ha sido llevado a cabo satisfactoriamente.
- **Delayed:** Se ha retrasado por algún motivo. Es recomendable especificarlo en los comentarios para mantener a todo el personal informado.
- **Overdue:** Caducada. Ha pasado el día previsto de finalización y nadie ha tomado ninguna acción.

- **To be processed:** El trabajo físico ha terminado, falta realizar el procesado posterior.
- **Pending documentation:** Faltan pocos días para que empiece la revisión y todavía no se han subido los paquetes de trabajo.

Esta lista se puede extender o reducir según las necesidades de las aerolíneas con sólo cambiar un parámetro en el código.

4.2.4. Tabla damages

La última tabla de la base de datos recoge los datos de los daños de las aeronaves. Es la más extensa de todas y contiene la siguiente información:

Tabla 4.5 Damages

damCat	<p>Categoría del daño:</p> <ul style="list-style-type: none"> • A: Daño que no requiere reparación o que ha sido reparado de manera permanente y no requiere ninguna acción futura. • B: Daño que no requiere reparación o que ha sido reparado pero que debe ser inspeccionado cada cierto intervalo de tiempo. • C: Daño que no ha podido ser reparado o que ha sido reparado de manera temporal y que requiere reparación permanente además de inspecciones periódicas.
ATA	Número del capítulo ATA correspondiente.
location	Breve descripción de la ubicación del daño.
damType	Tipo de daño, a elegir entre una lista predeterminada. Algunos ejemplos serían: dent, scratch, gouge...
damDimensions	Dimensiones del daño.
entryDate	Fecha en la que se encuentra.
closingDate	Fecha en la que se cierra, es decir, cuando ya no requiere más inspecciones ni reparaciones.
damReferences	Referencias a manuales y otros documentos.
PN	Número de pieza.

SN	Número de serie.
insplInterval	Intervalo de inspección. Sólo para daños categoría B y C.
intRepair	Tiempo máximo hasta la reparación. Sólo para daños categoría C.
damComments	Comentarios o información relevante que no tiene cabida en ningún otro campo.
TMAsigned	Indica si está firmado por un TMA o no.
signedBy	Nombre de usuario del TMA que ha firmado el daño
certified	Indica si ha sido certificado o no.
certifiedBy	Nombre de usuario del ingeniero/a que ha certificado el procedimiento.
damChecked	Indica si ha sido revisado y si se han tomado las acciones oportunas por parte del equipo de ingeniería. Especialmente útil cuando el mantenimiento se externaliza ya que la certificación la lleva a cabo un miembro del MRO y la revisión la hace el personal de la CAMO.
checkedBy	Nombre de usuario del ingeniero/a que ha revisado la documentación.
damAcftReg	Matrícula del avión al que pertenece el daño.
IDcheck	Identificador de la revisión en la que se encontró el daño.
isSDTrb	Parámetro de uso interno del sistema que indica si un daño es nuevo o el resultado de una tarea estructural (SDT).
Map	Indica si el daño está incluido no en el mapa de daños
damID	Identificador del daño de formato predeterminado: matrícula_número de daño. Por ejemplo: EC-AAA_dam1.

4.2.5. Conexión a la base de datos

Dado que la base de datos que proporciona la información para que airDocs funcione no es local, sino que todos los usuarios deben poder conectarse a ella de manera remota para obtener los datos actualizados, es necesario un intermediario entre la aplicación y la base de datos. Es por eso que se crea una API REST con LoopBack 3. Una API REST es una interfaz entre varios sistemas capaz de hacer peticiones HTTP. En el anexo C se pueden ver todas las peticiones que hace la API creada para airDocs, pero en general se pueden agrupar en tres grandes tipos:

- **GET:** Obtiene información de la base de datos.
- **POST:** Añade registros a la base de datos.
- **PUT:** Edita registros de la base de datos.
- **DELETE:** Elimina registros de la base de datos.

Para que la API funcione correctamente es necesario definir unos modelos de datos, que siguen la misma estructura que las tablas de la base de datos. Además, en el código de la aplicación hay que definir cuatro clases, una por tabla y modelo, que también siguen su misma estructura.

Es importante destacar que airDocs sólo funciona con conexión a Internet, ya que de lo contrario no es posible acceder a los datos.

4.3. Almacenamiento de archivos

Con el fin de asegurar que los documentos están siempre disponibles y cualquier usuario puede visualizarlos desde cualquier dispositivo, es necesario guardarlos en un servidor para acceder remotamente. Si bien teóricamente es posible guardar archivos en la misma base de datos MySQL donde se almacena el resto de la información, es muy ineficiente hacerlo. Además, recuperar la información no es tarea fácil. Es por eso que airDocs usa Dropbox para almacenar archivos. Se opta por esta plataforma y no por otra en primer lugar por sus estándares de seguridad (hoy en día ya hay aerolíneas que la usan), y también porque permite guardar cualquier tipo de archivo, desde hojas de Excel o archivos de texto, hasta imágenes, vídeos o mensajes de voz.

Para que la aplicación funcione de manera correcta y sea capaz de acceder a los archivos solicitados, es necesario que la cuenta de Dropbox asociada tenga una estructura concreta. Debe existir una carpeta por avión con su matrícula como nombre. Dentro de la misma debe haber una carpeta por cada revisión, cuyo nombre debe ser el mismo que el identificador de check que aparece en la base de datos. Por último, dentro de cada carpeta de revisión debe haber las carpetas “New Damages” donde se guarda la documentación de los daños nuevos, “SDT RB” donde se encuentran los archivos con los resultados de las inspecciones y reparaciones realizadas a los daños categoría B o C, “Work Packages” donde los TMA podrán encontrar las instrucciones de las tareas a realizar y “Other” con cualquier archivo que no pertenezca a las carpetas anteriores. En la Figura 4.2 se muestra un ejemplo de la estructura explicada.

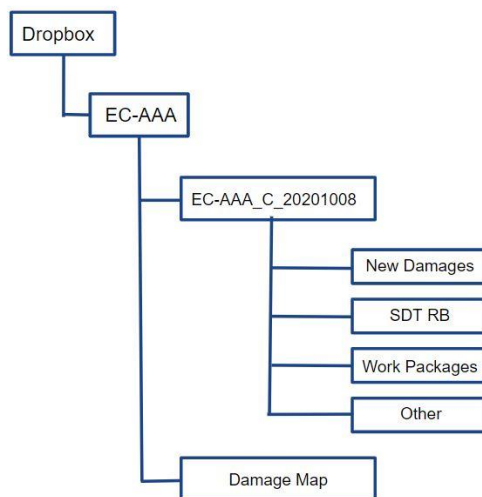


Fig. 4.2 Ejemplo de estructura de carpetas de Dropbox

Además, dentro de la carpeta de cada avión se debe incluir una carpeta llamada “Damage Map” donde debe estar guardado el mapa de daños en su versión más reciente. De este modo todos los usuarios podrán acceder a él a través de airDocs.

Para ayudar a los usuarios a llevar un buen control de las carpetas y evitar confusiones la aplicación recuerda cada vez que se crea una revisión o daño nuevo que es necesario crear su carpeta correspondiente en Dropbox, e indica el nombre que debe tener.

La organización de Dropbox no es del todo fija, sino que se puede adaptar a las necesidades de cada aerolínea añadiendo o quitando carpetas según sea conveniente. Lo único que hay que tener en cuenta es que se tendrán que aplicar pequeñas variaciones al código de acuerdo con los cambios realizados.

Aunque Dropbox es una muy buena opción para guardar archivos, si la aerolínea que quiere implementar airDocs ya usa otra plataforma y desea mantenerla, la aplicación está preparada para poder cambiar de Dropbox a otro proveedor siempre y cuando este tenga página web. Esto es así porque, tal y como se ha mencionado anteriormente, uno de los principales objetivos de airDocs es facilitar la transición y adaptarse al mayor número de escenarios posible.

4.4. Pantalla de inicio

Para acceder a airDocs es necesario disponer de un nombre de usuario y contraseña a introducir en una pantalla como la que se muestra en la Figura 4.3. para autenticarse. Según el trabajo que desempeñe el usuario, el sistema le mostrará uno u otro módulo de forma automática. Así pues, una ingeniera no puede acceder al módulo de TMA del mismo modo que un miembro de la

tripulación no está autorizado a entrar en el módulo de ingeniería. Solo los administradores tienen acceso a todos los módulos y a ciertas funcionalidades que otros usuarios no tienen.



Fig. 4.3 Pantalla de inicio

En caso de que el nombre de usuario o la contraseña no sean correctos el sistema avisará de ello al usuario, del mismo modo que si no hay conexión también se mostrará un aviso.

La pantalla inicial incluye un botón de ayuda en la parte superior izquierda que muestra una guía de usuario con una descripción de cómo funciona airDocs.

4.5. Módulo de administración

Los usuarios con rol de administrador son los únicos que pueden acceder a todos los módulos y funcionalidades disponibles en airDocs. Para ello disponen de una pantalla como la de la Figura 4.4 donde se muestran las distintas opciones.

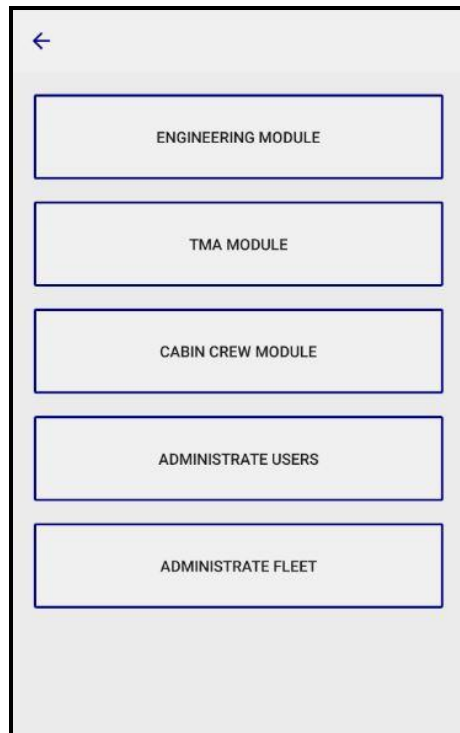


Fig. 4.4 Módulo de administrador

Un clic en “Engineering Module”, “TMA module” o “Cabin Crew Module” llevará al usuario a los módulos correspondientes. En los dos primeros, las funcionalidades de las que dispone el administrador son las mismas que las de los ingenieros o TMA (según el caso), pero con el añadido de que se pueden eliminar revisiones y daños. Solo está permitido a los administradores por razones de seguridad y para evitar una pérdida de información.

Para ver, editar y eliminar usuarios hay que hacer clic en “Administrate Users” y se abrirá una pantalla como la que se muestra en la Figura 4.5.

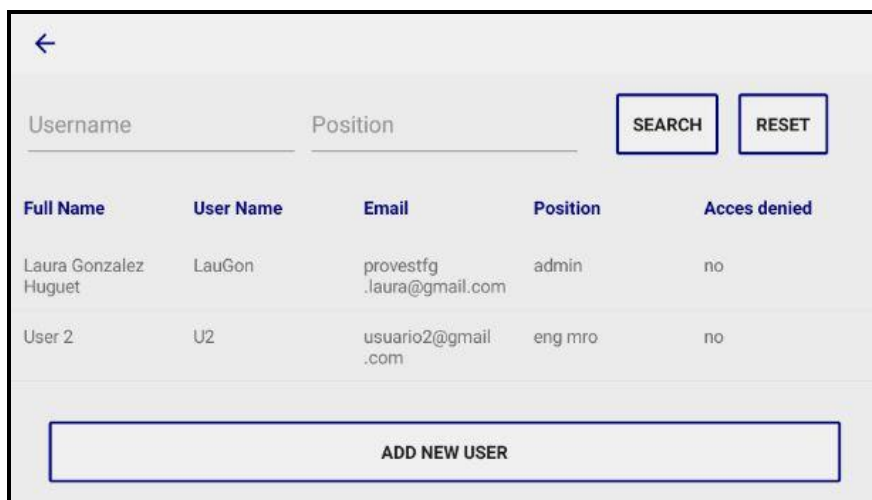


Fig. 4.5 Pantalla para visualizar usuarios

Cuando se selecciona un usuario de la tabla aparecen las opciones de editar o eliminar. Antes de proceder a suprimir un usuario de la base de datos el sistema pregunta para asegurar que la acción realmente se quiere llevar a cabo.

Si por el contrario se elige editar, se abrirá una ventana como la de la Figura 4.6 donde aparecerán los datos actualmente guardados. Tal y como se muestra en esta figura, se pueden editar todos los campos excepto el nombre de usuario, ya que es el parámetro que el sistema usa como identificador y cambiarlo podría llevar a errores o incoherencias.



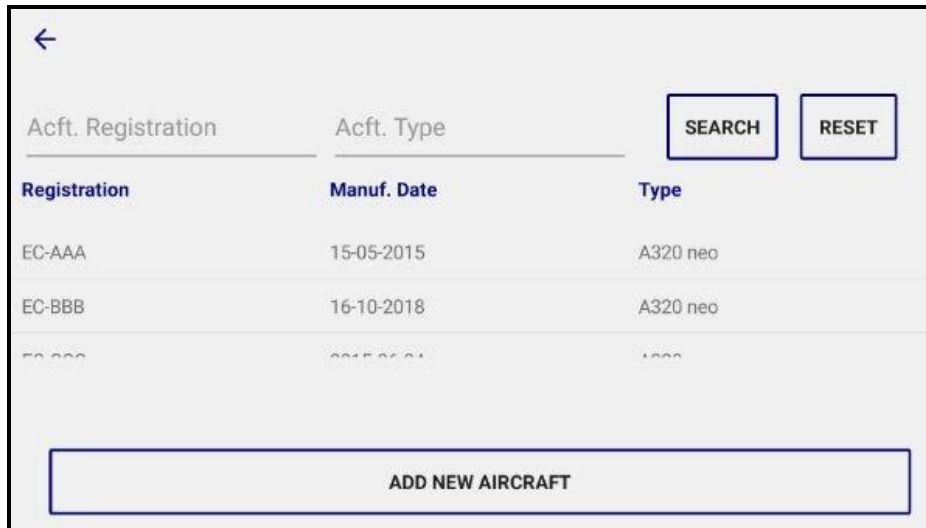
The screenshot shows a mobile application interface for editing a user. At the top left is a back arrow. Below it, several fields are listed with their current values: Full Name (User 2), Username (U2), Password (2222), Email (user2@gmail.com), Position (eng all), Acces Denied (no), and PIN (1234). Each field has a horizontal line below it, indicating it is an input field. At the bottom of the screen is a large rectangular button labeled "SAVE CHANGES".

Fig. 4.6 Pantalla de edición de usuarios

Para añadir un usuario, al seleccionar la opción “Add User” se muestra una ventana como la de edición, pero con los campos pendientes de rellenar. Si queda algún parámetro en blanco el sistema no permitirá proceder ya que deben estar todos cumplimentados.

Es importante destacar que para actualizar la tabla después de realizar cambios en los usuarios hay que deslizar la pantalla hacia abajo. Además, en la parte superior de la pantalla hay filtros para buscar usuarios según su nombre y posición.

Si el usuario selecciona la opción "Administrare Fleet" se abrirá una ventana como la que se muestra en la Figura 4.7 con una tabla con todos los aviones de la flota. De la misma manera que con los usuarios, si se selecciona uno se puede editar o eliminar. Si se desea suprimir una matrícula, el sistema pedirá una confirmación antes de proceder para evitar errores.



Acft. Registration	Acft. Type	
Registration	Manuf. Date	Type
EC-AAA	15-05-2015	A320 neo
EC-BBB	16-10-2018	A320 neo
EC-CCC	17-11-2019	A320 neo

SEARCH RESET

ADD NEW AIRCRAFT

Fig. 4.7 Pantalla para administrar la flota

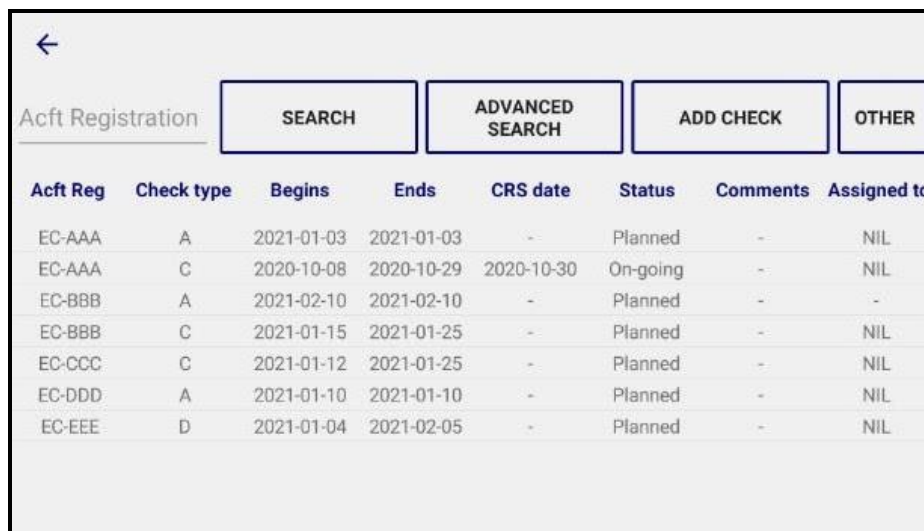
Al editar o añadir un elemento, no se abrirá una ventana diferente como en el caso de los usuarios, sino que en la parte inferior de la tabla aparecerán las opciones de edición. Esto es así porque hay muchos menos campos a rellenar y porque de este modo el sistema tiene una vista de contenido menos y puede compilar un poco más rápido.

Este módulo también incluye la opción de visualizar toda la documentación de una aeronave concreta, y permite filtrar las entradas de la tabla por matrícula y por tipo de avión. Además, en la parte superior izquierda de todas las pantallas hay una flecha que permite volver atrás.

4.6. Módulo de ingeniería

Este módulo está pensado tanto para aquellas aerolíneas que externalizan el mantenimiento como para aquellas que no. Por este motivo ha sido diseñado pensando tanto en el equipo de ingeniería de la CAMO como del MRO. En general la apariencia es la misma para ambos tipos de usuario, solamente cambian algunas funcionalidades concretas. Además, para aquellas aerolíneas que opten por realizar el mantenimiento ellas mismas, existe el tipo de usuario "eng all" que tiene permiso para realizar todas las acciones. De este modo airDocs puede ofrecer soluciones a las necesidades de cada aerolínea.

La pantalla principal de este módulo (Figura 4.8) consta de una tabla que muestra todas las revisiones almacenadas en la base de datos. En la parte superior de la misma se puede filtrar por matrícula y, además, para búsquedas más concretas, se pueden aplicar más filtros gracias a “Advanced search”. Al hacer clic en este botón se desplegarán más campos que permiten acotar la búsqueda según el estado de la revisión, su fecha de inicio y su fecha de fin.



Acft Reg	Check type	Begins	Ends	CRS date	Status	Comments	Assigned to
EC-AAA	A	2021-01-03	2021-01-03	-	Planned	-	NIL
EC-AAA	C	2020-10-08	2020-10-29	2020-10-30	On-going	-	NIL
EC-BBB	A	2021-02-10	2021-02-10	-	Planned	-	-
EC-BBB	C	2021-01-15	2021-01-25	-	Planned	-	NIL
EC-CCC	C	2021-01-12	2021-01-25	-	Planned	-	NIL
EC-DDD	A	2021-01-10	2021-01-10	-	Planned	-	NIL
EC-EEE	D	2021-01-04	2021-02-05	-	Planned	-	NIL

Fig. 4.8 Módulo de ingeniería

Para añadir una nueva revisión es necesario hacer clic en “Add check”. Esto llevará al usuario a una pantalla como la de la Figura 4.9 donde podrá rellenar toda la información y generar una nueva entrada en la base de datos. Antes de completar el proceso el sistema pide al usuario que cree una nueva carpeta destinada a la revisión, para así evitar posibles desajustes en la estructura de Dropbox. Es importante destacar que el sistema no permite que se dejen campos en blanco. Es por eso que, si no dispone de cierta información, el usuario deberá escribir un guion en el campo correspondiente.

Con el objetivo de facilitar el trabajo de los usuarios la mayoría de parámetros a cumplimentar disponen de listas desplegables de modo que, cuando se hace clic en ellos, aparecen alternativas entre las que elegir. Así el usuario no necesita escribir, puede simplemente seleccionar la opción que más se ajuste. Si ninguna de ellas es conveniente, el usuario siempre puede teclear el texto que crea pertinente.

Aircraft Registration	
Type of check	
Begins	11/20/2020
Ends	11/20/2020
CRS date	-
Status	-
Comments	-
Assigned to	-

ADD CHECK

Fig. 4.9 Pantalla para añadir y editar revisiones

El último de los botones que hay en la parte superior de la pantalla principal de este módulo (Figura 4.8), "Other", permite acceder a la documentación de una matrícula en concreto. Esto puede ser muy útil para consultar el histórico de archivos de un avión y ver todos los daños y reparaciones que han sido reportados a lo largo de su vida útil.

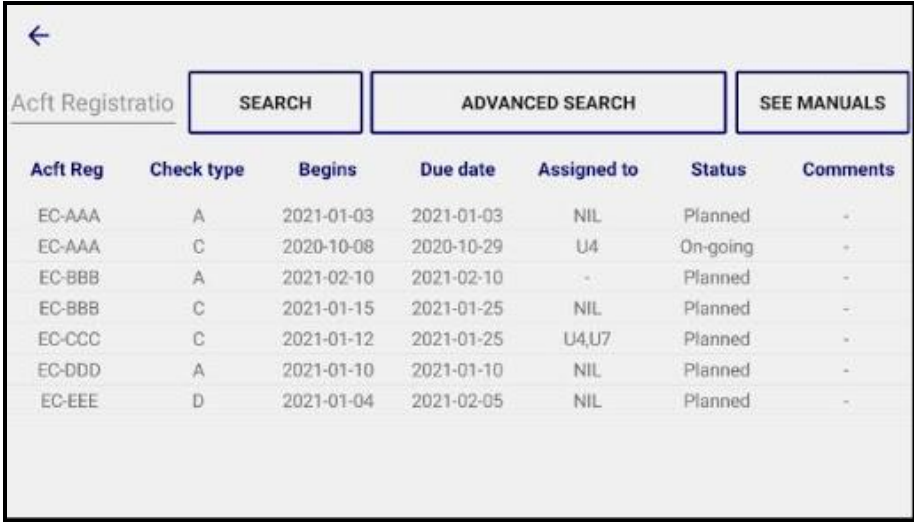
Para realizar acciones sobre las revisiones, el usuario debe seleccionar una de ellas en la tabla. Aparecerá un desplegable con las siguientes opciones:

- **Edit Check:** Permite editar los parámetros de la revisión en una pantalla como la mostrada en la Figura 4.9 pero con los datos de la misma.
- **See New Damages:** Permite ver todos los daños nuevos que han sido reportados por los TMA durante la revisión en cuestión. En el apartado 4.8. se explica en más profundidad las características de la pantalla donde se muestra el listado de daños.
- **See SDT RB:** Muestra los daños editados después de que se hayan realizado las tareas que se indican en la SDT correspondiente. En general son daños categoría B o C que han sido reparados o inspeccionados. Se muestran en una pantalla igual que la de los daños nuevos.
- **See Damage Map:** Muestra el mapa de daños del avión, que está almacenado en Dropbox.

- **See Work Packages:** Redirecciona al usuario a la carpeta de Dropbox donde se han subido las instrucciones de las tareas a realizar durante la revisión.
- **Upload files:** Redirecciona al usuario a la carpeta de Dropbox donde debe subir toda la documentación necesaria para llevar a cabo todos los trabajos requeridos.
- **Assign to TMAs:** Permite asignar la revisión a los TMAs para que la lleven a cabo.
- **Update Status:** Actualiza el estado de la revisión y lo guarda en la base de datos.
- **Query:** Permite enviar mensajes a otros usuarios para resolver dudas o reportar errores. Para ello usa el correo electrónico que está configurado en el dispositivo.
- **Delete check:** Permite eliminar la revisión seleccionada. Por motivos de seguridad y para evitar pérdida de información, sólo los administradores pueden llevar a cabo esta acción.

4.7. Módulo de TMA

El módulo de TMA está diseñado para que los mecánicos vean las revisiones que hay programadas y puedan reportar todos los daños que van encontrando durante el proceso. Para ello en la vista principal del módulo se muestra una tabla que recoge todas las revisiones, tal y como se puede observar en la Figura 4.10.



The screenshot shows a mobile application interface for the TMA module. At the top left is a back arrow. Below it is the text 'Acft Registratio'. To the right are three buttons: 'SEARCH', 'ADVANCED SEARCH', and 'SEE MANUALS'. Below these is a table with the following data:

Acft Reg	Check type	Begins	Due date	Assigned to	Status	Comments
EC-AAA	A	2021-01-03	2021-01-03	NIL	Planned	-
EC-AAA	C	2020-10-08	2020-10-29	U4	On-going	-
EC-BBB	A	2021-02-10	2021-02-10	-	Planned	-
EC-BBB	C	2021-01-15	2021-01-25	NIL	Planned	-
EC-CCC	C	2021-01-12	2021-01-25	U4,U7	Planned	-
EC-DDD	A	2021-01-10	2021-01-10	NIL	Planned	-
EC-EEE	D	2021-01-04	2021-02-05	NIL	Planned	-

Fig. 4.10 Módulo de TMA

En la parte superior de la pantalla hay un filtro que permite visualizar solamente las revisiones de la matrícula seleccionada. Para realizar búsquedas más concretas el botón “Advanced Search” permite filtrar según a qué usuarios se ha asignado, el estado, la fecha de inicio y la fecha de finalización.

A la derecha de los filtros está el botón “See Manuals”, que permite visualizar los manuales necesarios para llevar a cabo las tareas asignadas. Por defecto se abre la página de AirNavX, dónde Airbus posibilita la visualización en línea de manuales como el SRM, el AMM o el IPC entre muchos otros. Sin embargo, si la aerolínea lo desea, se puede cambiar esta web por cualquier otra sin ninguna complicación.

Cuando el usuario selecciona una revisión, aparece un desplegable con varias opciones entre las que elegir:

- **See damages:** Permite ver y reportar daños nuevos. En el apartado 4.8. se explica con más detalle las características de esta pantalla.
- **SDT RB:** Permite reportar los cambios realizados en un daño después de llevar a cabo una SDT. Se abre una ventana como la anterior, aunque con funcionalidades un poco diferentes (ver apartado 4.8.).
- **See Work Packages:** Redirecciona al usuario a la carpeta de Dropbox que contiene todos los WP de la revisión en cuestión.
- **See Damage Map:** Muestra el mapa de daños almacenado en la carpeta correspondiente de Dropbox.
- **Upload file:** Permite al TMA subir archivos a la carpeta de la revisión. Está pensado principalmente para documentos que deben ir a la carpeta “Others” aunque permite guardarlos en cualquiera de las cuatro carpetas.
- **Query:** Sirve para enviar mensajes a otros usuarios, principalmente al equipo de ingeniería, con el objetivo de resolver dudas y reportar errores o incoherencias. Usa el correo configurado en el dispositivo.

4.8. Show Damages

Para mostrar los daños y trabajar con ellos se ha diseñado la pantalla “Show Damages”. Se puede acceder a ella desde el módulo de ingeniería y desde el módulo de TMA, aunque tanto su aspecto como las acciones que permite realizar cambian levemente. La parte que es común a ambos incluye una tabla donde se muestran los parámetros más importantes de cada daño.

4.8.1. Show Damages para ingeniería

Cuando un miembro del equipo de ingeniería selecciona la opción “See New Damages” o “See SDT RB” aparece una pantalla con todos los daños o los resultados de las SDT, según el caso. Su aspecto varía en función de si se externaliza o no el mantenimiento. En el ejemplo de la Figura 4.11 no hay externalización, es decir es lo que ven los usuarios “eng all” y “admin”. En caso de que la hubiera se mostraría sólo el botón “Show only not certified” para los ingenieros MRO y sólo “Show only not checked” para los de la CAMO, ya que los primeros se encargan de certificar los daños y los segundos de comprobar que todo es correcto y que no hay errores ni incoherencias.

Description	Dimensions	Cat.	ATA	PN	SN	References	Comments
dent in fan cowl RH eng.2 outer skin	10x18	C	54-20	642-3002-521	8497001	SRM 54-20-00 REV.134	-
dent in forward fuselage skin FR 12-13 STRG 34-36 RH	80x65	C	53-11	N/A	N/A	SRM 53-11- 11-283-012 REV.134	-

Fig. 4.11 Show Damages para ingeniería

Aquí, en lugar de filtros manuales, encontramos botones que permiten visualizar sólo los daños con ciertas características. “Show only not certified” muestra únicamente los daños pendientes de certificar mientras que “Show only not checked” hace lo propio, pero con los que no están revisados. Además, “Show only map” muestra aquellos daños que se incluyen en el mapa de daños y “Show only no map” muestra los que quedan excluidos. Por último, “Show all damages” vuelve a mostrar todas las entradas en la tabla.

Si el usuario selecciona un daño concreto, aparece un desplegable con las siguientes opciones:


- **Certify:** Permite marcar el daño como certificado. Si hay externalización sólo lo pueden hacer usuarios “eng mro” o “admin”.
- **Check:** Marca el daño como revisado. Si hay externalización sólo lo pueden llevar a cabo usuarios “eng camo” o “admin”.

- **Further information:** Muestra una ventana emergente con información adicional sobre el daño, que no se muestra en la tabla por razones de espacio y distribución.
- **See documentation:** Lleva al usuario a la carpeta de Dropbox que contiene todos los archivos relacionados con el daño.
- **Set as map/ Set as no map:** Permite decidir si incluir o no el daño en el mapa de daños.

Para completar el proceso de certificación o verificación, el sistema pide al usuario que introduzca su PIN. Este es un número de identificación personal que verifica que la persona que lleva a cabo la acción es quién dice ser. Además, en la base de datos se almacena el nombre del usuario que ha llevado a cabo la acción para así tener trazabilidad.

4.8.2. Show Damages para TMA

Para reportar daños o el resultado de SDT, los TMA son redireccionados a esta pantalla desde su módulo. Su aspecto es un poco diferente al que ve el equipo de ingeniería, ya que las acciones a realizar también son distintas.



Description	Dimensions	Cat.	ATA	PN	SN	References	Comments
dent in fan cowl RH eng.2 outer skin	10x18	C	54-20	642-3002-521	8497001	SRM 54-20-00 REV.134	-
dent in forward fuselage skin FR 12-13 STRG 34-36 RH	80x65	C	53-11	N/A	N/A	SRM 53-11- 11-283-012 REV.134	-
Other: missing sealant in fwd	N/A	Cx	51-00	N/A	N/A	-	Sealant replaced

CREATE NEW DAMAGE

Fig. 4.12 Show Damages para TMA

Tal y como se puede ver en la Figura 4.12, desaparecen los botones de filtrado y se sustituyen por el botón “Create New Damage” o “Report Back SDT” según si se ha seleccionado la opción “See damages” o “SDT RB” en el módulo de TMA. En ambos casos, al hacer clic se abre una nueva pantalla que permite introducir la información sobre el daño. Si se trata de uno nuevo, el formulario estará en blanco, mientras que si se trata de un Report Back el usuario primero deberá seleccionar el daño al que aplica el SDT, y a continuación se mostrarán los campos rellenos con los datos de dicho daño, listos para ser editados.

Para facilitar el trabajo de los usuarios, muchos de los campos disponen de listas desplegables con opciones entre las que elegir.

Igual que anteriormente, si el usuario selecciona un daño de la tabla, aparecerá un desplegable con las siguientes opciones:

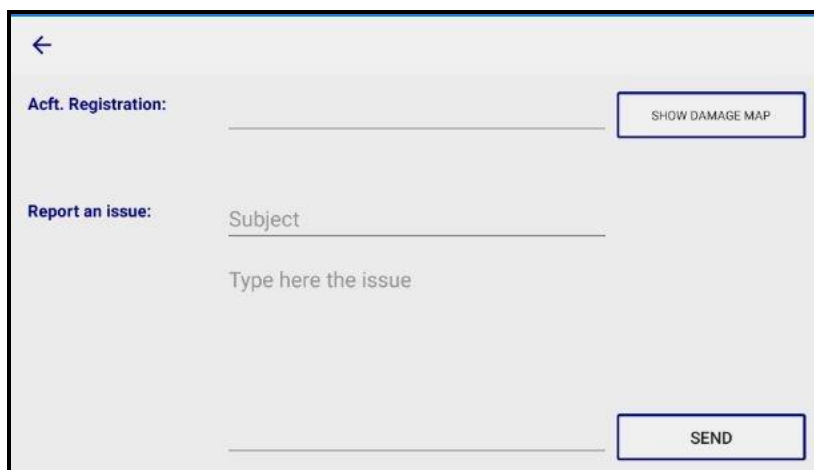
- **Further information:** Muestra información adicional sobre el daño.
- **Add file:** Permite subir archivos a la carpeta que corresponde al daño. Se pueden añadir documentos Excel o Word, imágenes, vídeos o notas de voz entre otros.
- **See documentation:** Muestra al usuario la carpeta de Dropbox con toda la documentación relativa al daño.
- **Edit damage:** Abre una nueva ventana dónde se puede editar el daño seleccionado. Para guardar los cambios el sistema pide al usuario que introduzca su PIN y almacena su nombre en la base de datos para dar trazabilidad a la operación. Sucede lo mismo cuando se crea un daño nuevo.
- **Delete damage:** Permite eliminar el daño seleccionado. Por seguridad, sólo los administradores tienen permiso para realizar esta acción.

Cuando se crea un daño o Report Back nuevo, el sistema pide al usuario que cree una carpeta en Dropbox con el nombre que corresponde al identificador del daño para así evitar desajustes en la estructura de carpetas. Es importante tener en cuenta que el sistema no permite que queden campos en blanco, de modo que, si no se dispone información suficiente, antes de guardar se deberá escribir un guion en los campos que falte cumplimentar.

4.9. Módulo de tripulación de cabina

Para acabar de cerrar el círculo, el módulo de tripulación de cabina permite al personal de abordaje acceder al mapa de daños de las aeronaves de la flota sin la necesidad de tener la copia física. De este modo no sólo se reduce la cantidad de papel usada y, en consecuencia, los costes, sino que también se puede acceder a los mapas de daños actualizados en todo momento. Esto no sucede con las copias en papel ya que desde que el equipo de ingeniería las actualiza hasta que se imprimen y un TMA las lleva al avión pueden pasar días.

Adicionalmente, tal y como se muestra en la Figura 4.13, este módulo permite a las tripulaciones reportar problemas directamente a los ingenieros y a los TMA.



The screenshot displays a mobile application interface for a crew module. At the top left, there is a back arrow icon. Below it, the 'Acft. Registration:' section features a text input field and a 'SHOW DAMAGE MAP' button. The 'Report an issue:' section includes a 'Subject' text input field, a larger text area with the placeholder 'Type here the issue', and a 'SEND' button at the bottom right.

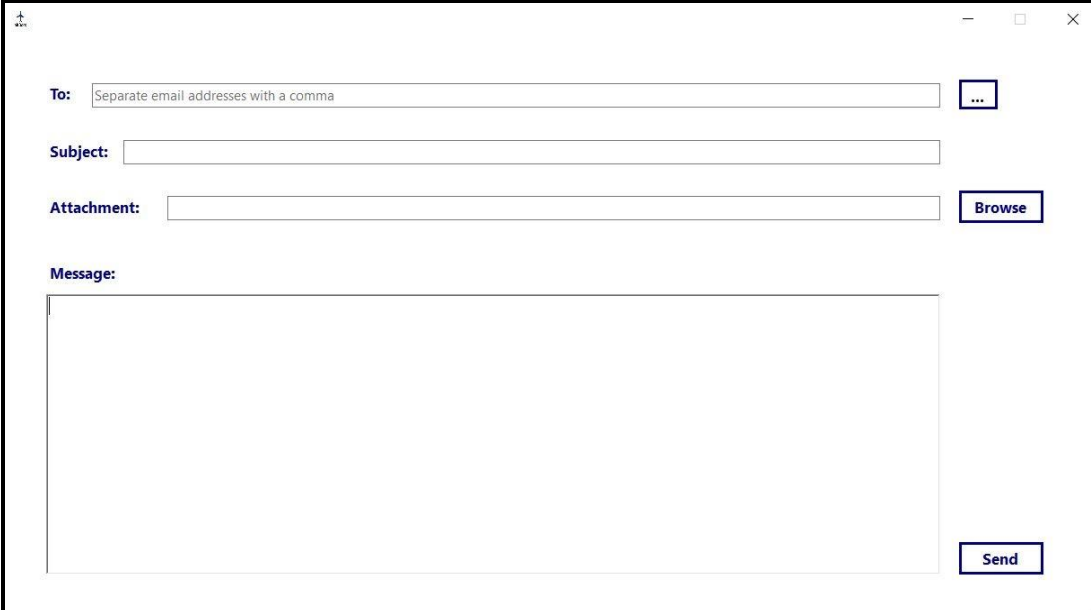
Fig. 4.13 Módulo de tripulación

4.10. AirDocs para ordenadores

Con el objetivo de ofrecer a cada usuario la solución que más se adapta a sus necesidades, también se ha desarrollado una versión de escritorio de airDocs. Así mientras los TMAs pueden aprovechar las ventajas que les ofrece trabajar con tabletas, aquellos miembros del personal que trabajan en oficinas pueden acceder a airDocs desde su ordenador habitual de trabajo.

El funcionamiento de ambas versiones es muy similar, aunque su apariencia y algunos procesos cambian ligeramente. Una de las características que funciona de forma diferente en la versión de escritorio es el uso del correo electrónico para enviar reportes o consultas. En lugar de abrir la aplicación de mensajería instalada en el dispositivo como sucede con la versión móvil, aparece una pantalla propia de la aplicación (Figura 4.14) donde el usuario puede introducir el destinatario, el asunto y el mensaje, y también adjuntar un archivo. Al hacer clic en enviar, el sistema pide la contraseña de la dirección de correo asociada a su usuario y envía el mensaje directamente.

Junto a la casilla "To" hay un botón que muestra una lista de contactos con todos los usuarios del sistema, su rol y su correo electrónico. Es importante que, si se desea enviar el mensaje a más de un destinatario, sus direcciones estén separadas solamente por una coma, sin ningún espacio en blanco. De lo contrario el sistema mostrará un aviso y no permitirá proceder con el envío.



The image shows a screenshot of a desktop application window titled "airDocs" with standard window controls (minimize, maximize, close). The interface is for sending an email and contains the following elements:

- To:** A text input field with the placeholder text "Separate email addresses with a comma" and a small "..." button to its right.
- Subject:** A text input field.
- Attachment:** A text input field with a "Browse" button to its right.
- Message:** A large, empty text area for composing the email body.
- Send:** A button located at the bottom right of the message area.

Fig. 4.14 Pantalla para enviar mensajes

AirDocs para ordenadores contiene una funcionalidad de la que la aplicación móvil no dispone, "Export Damages". Este botón se muestra en el desplegable que aparece en el módulo de ingeniería cuando el usuario hace clic en "Options" después de seleccionar una revisión de la tabla. Genera un archivo de texto (.txt) que se guarda donde el usuario indique, y que contiene toda la información de los daños que se han encontrado en la revisión seleccionada. Esta característica es muy útil para aquellas aerolíneas que deseen combinar el uso de airDocs con otro software de mantenimiento, como podría ser AMOS, ya que el archivo se puede cargar en el sistema y así no es necesario entrar los daños uno a uno. El formato del archivo se puede variar según sea necesario cambiando ligeramente el código.

CAPÍTULO 5. CONCLUSIONES

La idea para este trabajo, y parte de los conocimientos necesarios para llevarlo a cabo, surgieron del aprendizaje adquirido durante las prácticas en una aerolínea. El objetivo principal era desarrollar una aplicación que permitiera reducir la dependencia del papel en las revisiones de mantenimiento, para así poder agilizar las tareas relacionadas y asentar las bases de la futura digitalización de todos los procesos.

Para conseguir dicha meta, lo primero fue investigar cómo son los trabajos mediante los que se garantiza que un avión esté en perfecto estado para volar, además de estudiar la normativa aplicable. Asimismo, analizar las herramientas disponibles en el mercado también resultó ser un factor clave para definir las características de la aplicación diseñada.

AirDocs está formada por cuatro módulos: el de administración, el de ingeniería, el de TMA y el de la tripulación. Así todo el personal dispone de un entorno personalizado que le permite llevar a cabo las tareas propias de su puesto de trabajo. Además, la pantalla “Show Damages” permite reportar, editar y certificar los daños encontrados durante las revisiones. Adicionalmente, en la versión de escritorio, estos daños se pueden exportar a un archivo de texto para cargarlos en otros programas, brindando así la posibilidad de compatibilizar airDocs con otros tipos de software.

Buscando ofrecer soluciones que se puedan adaptar a todos los usuarios, se ha desarrollado una versión móvil y una versión de escritorio de airDocs. De este modo, tanto el personal que trabaja con un ordenador fijo como aquellos que necesitan usar una tableta o incluso un móvil, pueden disponer de la aplicación.

Con la mirada en el futuro, el siguiente paso a dar sería poner a prueba airDocs en un entorno real. Esto permitiría recibir opiniones y críticas de los usuarios, fundamentales para acabar de ajustar las características de la aplicación y asegurar que su uso perdura. También se podría estudiar añadir funcionalidades extra como, por ejemplo, un módulo para editar el mapa de daños o un servidor propio, de manera que no sea necesario depender de proveedores externos como Dropbox.

Hasta conseguir la completa digitalización de la industria todavía queda camino por recorrer, pero todo apunta a que sucederá en los próximos años. En cuanto las aerolíneas se recuperen de la crisis actual, podrán invertir más en este tipo de tecnologías, que ya empiezan a despegar de la mano de herramientas destinadas al mantenimiento preventivo. De este modo no sólo se reducirán los costes, sino que, además, se podrá ir un paso por delante y contribuir a alargar la vida útil tanto de las aeronaves como de sus componentes.

WEBGRAFÍA

[1]: Skybrary, diccionario de términos y conceptos aeronáuticos : <https://www.skybrary.aero>

[2]: Going paperless in the hangar: <https://www.avm-mag.com/going-paperless-in-the-hangar/>

[3]: Documentación de Xamarin: <https://docs.microsoft.com/es-es/xamarin/>

[4]: Documentación de Loopback 3: <https://loopback.io/doc/en/lb3/>

[5]: Documentación mySQL: <https://dev.mysql.com/doc/>

Anexo A. Manual de usuario: versión móvil

AirDocs es una aplicación para la gestión de la documentación relativa al mantenimiento de una flota de aeronaves. El presente documento expone el funcionamiento de airDocs para dispositivos móviles y busca resolver todas aquellas dudas que puedan surgir durante su uso. Siempre que sea posible, es muy recomendable usar airDocs en una tableta o un móvil grande ya que de lo contrario es posible que la disposición de los elementos de ciertas pantallas se vea alterada. Si no existe esta posibilidad, se recomienda cambiar la orientación del dispositivo según convenga con el fin de encontrar el modo que mejor se ajuste a la pantalla que se está visualizando.

Inicio de sesión

Para iniciar sesión es necesario disponer de un nombre de usuario y una contraseña que deberán ser introducidos en una pantalla como la que se muestra en la Figura A.1



The image shows a mobile application login screen for 'airDocs'. At the top left, there is a 'HELP' button. In the center, there is a blue silhouette of an airplane with vertical lines underneath it, representing a landing gear or maintenance points. Below the airplane icon, the text 'airDocs' is displayed in a large, bold, black font. Underneath the logo, there are two input fields: 'Username' and 'Password', each with a horizontal line for text entry. At the bottom center, there is a 'LOGIN' button.

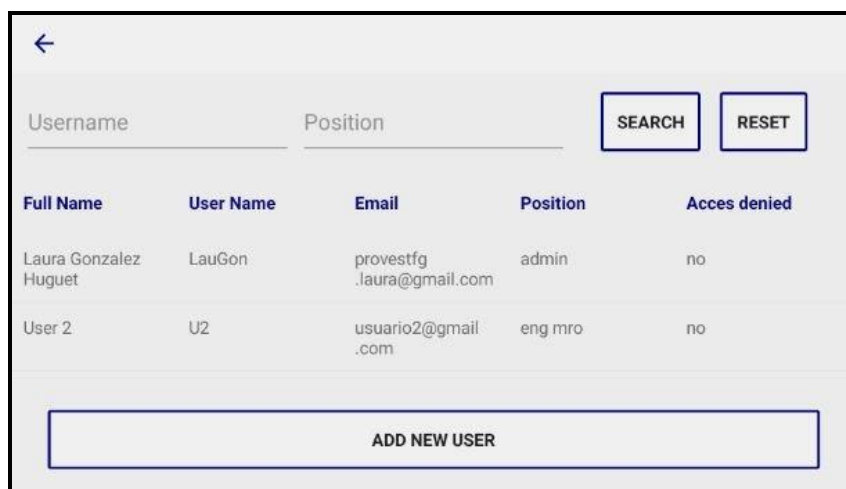
Fig. A.1 Inicio de sesión

El sistema muestra una alerta si el usuario no existe, la contraseña es incorrecta o si no hay conexión. En este último caso es posible que tarde un

poco más en aparecer el aviso que en los dos primeros. En caso de pérdida de la contraseña es necesario contactar con los administradores del sistema, ya que son ellos los únicos que pueden cambiarla.

Trabajar con usuarios: añadir, editar y eliminar

Todas aquellas acciones relacionadas con los usuarios sólo las pueden llevar a cabo los administradores. Desde su módulo deben hacer clic en “Administrate Users” y se abrirá una pantalla como la de la Figura A.2.

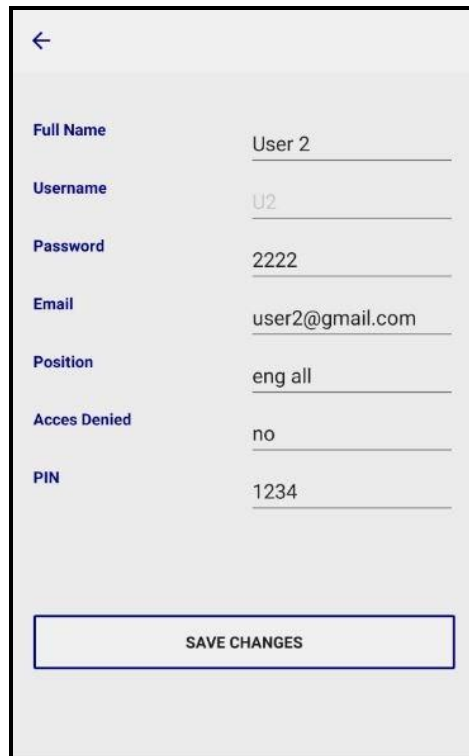


The screenshot shows a mobile application interface for user management. At the top left is a back arrow. Below it are two search filters: 'Username' and 'Position', followed by 'SEARCH' and 'RESET' buttons. A table lists two users with columns for Full Name, User Name, Email, Position, and Acces denied. At the bottom is a large 'ADD NEW USER' button.

Full Name	User Name	Email	Position	Acces denied
Laura Gonzalez Huguet	LauGon	provestfg.laura@gmail.com	admin	no
User 2	U2	usuario2@gmail.com	eng mro	no

Fig. A.2 Administrar usuarios

Para **añadir** un usuario nuevo hay que hacer clic en el botón “Add New User”, mientras que para **editar** uno hay que seleccionarlo en la tabla, y hacer clic en la opción “Edit User” presente en el desplegable que aparece. En ambos casos se abrirá una pantalla como la que se muestra en la Figura A.3 con los campos por rellenar o ya cumplimentados y listos para editar. En dicha Figura también se puede observar que el nombre de usuario no es editable, ya que se trata de un parámetro clave para el funcionamiento del sistema. Para realizar cambios en este parámetro es necesario crear un usuario nuevo y eliminar el antiguo.



The image shows a mobile application screen for adding or editing a user. At the top left is a back arrow. Below it are seven form fields, each with a label on the left and a text input on the right. The fields are: Full Name (User 2), Username (U2), Password (2222), Email (user2@gmail.com), Position (eng all), Acces Denied (no), and PIN (1234). At the bottom center is a rectangular button with the text "SAVE CHANGES".

Full Name	User 2
Username	U2
Password	2222
Email	user2@gmail.com
Position	eng all
Acces Denied	no
PIN	1234

SAVE CHANGES

Fig. A.3 Añadir o editar usuario

Solo cuando estén todos los campos rellenos el sistema procederá a guardar la información en la base de datos. Una vez de vuelta a la pantalla de la Figura A.2, si no se ha actualizado la tabla solamente hay que deslizarla hacia abajo y se cargarán los datos de nuevo.

Para **eliminar** un usuario hay que seleccionarlo en la tabla y hacer clic en "Delete User" en el desplegable. Antes de proceder el sistema preguntará si se desea llevar a cabo la acción para evitar así una pérdida accidental de información.

Trabajar con aviones: añadir, editar, eliminar y visualizar documentación

Tal y como sucede con los usuarios, todas las acciones relacionadas con los datos de la flota sólo las pueden llevar a cabo los administradores. Para ello deben seleccionar en su módulo la opción "Administrate Fleet" y el sistema les llevará a una ventana como la mostrada en la Figura A.4.

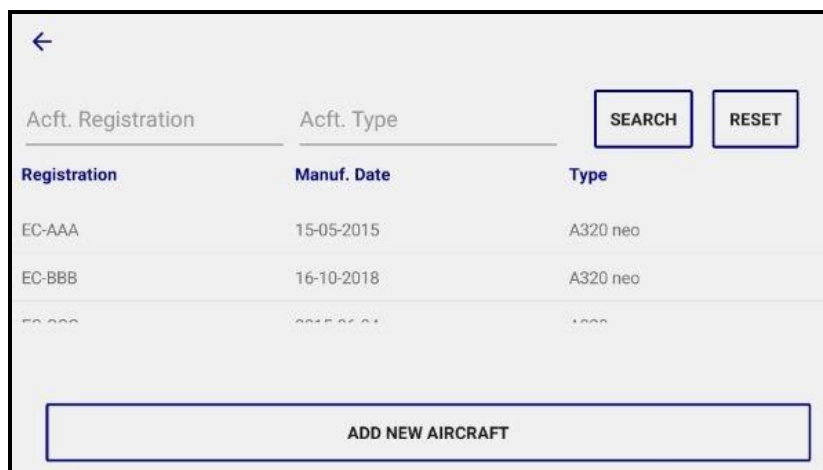


Fig. A.4 Administrar flota

Si se desea **añadir** un nuevo avión hay que hacer clic en la opción “Add New Aircraft”. Aparecerán tres ventanas emergentes en las que el usuario deberá introducir la matrícula, la fecha de fabricación y el tipo de aeronave. En cualquier momento se puede interrumpir el proceso, pero no es posible volver atrás, es decir, si se comete un error no es posible corregirlo volviendo a la ventana anterior, sino que hay que esperar a completar el proceso para editarlo después o bien empezar de nuevo.

Para **editar** un avión hay que seleccionarlo en la tabla y hacer clic en “Edit Aircraft” en el desplegable. En lugar de una pantalla nueva, los campos a rellenar aparecen en la parte inferior de la tabla, tal y como se muestra en la Figura A.5. Es necesario tener en cuenta que algunos dispositivos muy pequeños pueden presentar problemas al mostrar todos los elementos, por lo que se recomienda cambiar su orientación según convenga.



Fig. A.5 Editar avión

Es importante destacar que los cambios en la matrícula no se guardarán, sino que darán lugar a una nueva entrada en la tabla. Esto es debido a que se trata de un parámetro clave que no se puede editar.

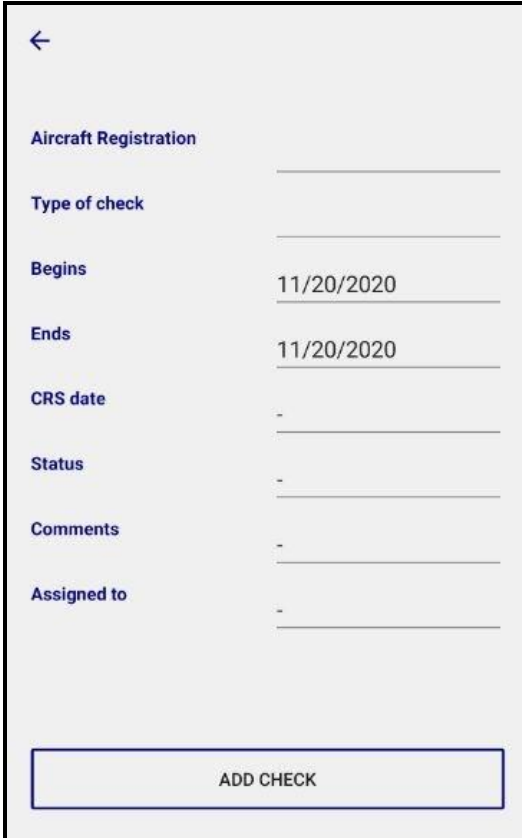
Para **eliminar** un avión del sistema sólo hay que seleccionarlo y hacer clic en la opción "Delete Aircraft" en el desplegable. Antes de proceder el sistema pedirá al usuario que confirme que realmente quiere llevar a cabo la acción para evitar errores.

Por último, para **visualizar** toda la **documentación** relativa a una matrícula hay que seleccionarla en la tabla y hacer clic en la opción "See Documentation". A continuación, el usuario será redireccionado a la carpeta de Dropbox correspondiente en el visor web.

Trabajar con revisiones: añadir, editar, eliminar y otras acciones

Las revisiones son uno de los ejes centrales del mantenimiento de las aeronaves. Se puede acceder a ellas desde los módulos de ingeniería y de TMA y, en consecuencia, también desde el módulo de administración. Sin embargo, sólo pueden añadirlas y editarlas los miembros del equipo de ingeniería y los administradores, quienes adicionalmente pueden eliminar las que convenga.

Para **añadir** una nueva revisión hay que hacer clic en el botón "Add Check" visible en el módulo de ingeniería. La información requerida es la que se puede observar en la Figura A.6. El sistema sólo permite guardar la revisión cuando todos los campos están cumplimentados. En caso de que no se disponga de cierta información, como podría ser la fecha de puesta en servicio, es suficiente con escribir un guión. Más adelante se podrá **editar** este parámetro y todos los demás haciendo clic en la opción "Edit Check" del desplegable que aparece cuando se selecciona una revisión en la tabla.



The screenshot shows a mobile application interface for adding or editing a check. At the top left, there is a back arrow. Below it, the title "Aircraft Registration" is displayed. The form consists of several fields, each with a label on the left and a corresponding input field on the right. The fields are: "Type of check", "Begins" (with the value "11/20/2020"), "Ends" (with the value "11/20/2020"), "CRS date", "Status", "Comments", and "Assigned to". At the bottom of the form, there is a large button labeled "ADD CHECK".

Fig. A.6 Añadir o editar revisiones

Los administradores tienen la capacidad de **eliminar** una revisión con un clic en "Delete Check" en el mismo desplegable en el que se encuentra la opción "Edit Check".

Como **acciones adicionales**, gracias a las opciones que aparecen al seleccionar una revisión, los ingenieros pueden:

- Ver los daños encontrados durante la revisión con un clic en "**See New Damages**" (ver siguiente apartado para más detalles)
- Ver los resultados de las SDT llevadas a cabo seleccionando "**See SDT RB**" (ver siguiente apartado para más detalles)
- Ver el mapa de daños almacenado en la carpeta correspondiente de Dropbox con la opción "**See Damage Map**".
- Visualizar y añadir en Dropbox los paquetes de trabajo con las instrucciones a llevar a cabo clicando en "**See Work Packages**".
- Subir a Dropbox cualquier tipo de archivo con seleccionar "**Upload Files**". Antes de redireccionar al usuario a la página web, el sistema preguntará en qué carpeta se desea guardar la documentación para así ir directamente a ella.

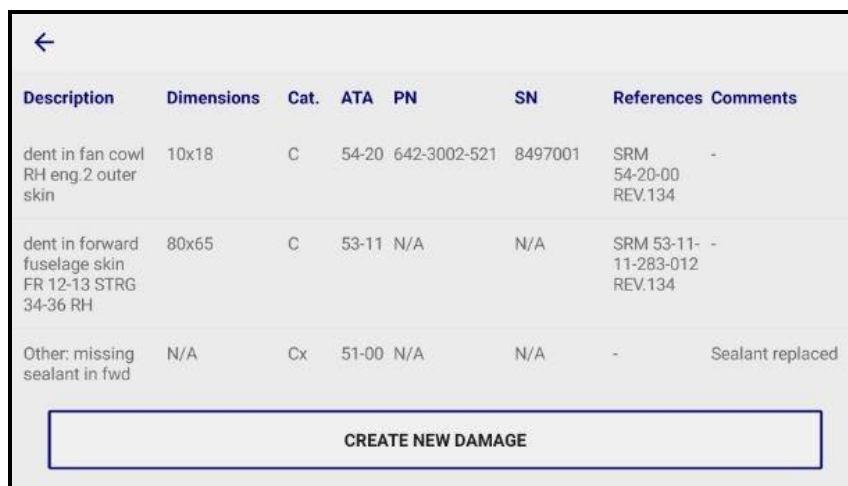
- Asignar las revisiones a los TMAs para que así estos sepan los trabajos que deben realizar. Para ello hay que seleccionar la opción **“Assign to TMAs”**. Aparecerá una ventana con un desplegable con los nombres de todos los mecánicos. Para seleccionar más de uno hay que introducir sus nombres de usuario separados solamente por una coma.
- Actualizar el estado de la revisión gracias a la opción **“Update Status”**. Al hacer clic en ella aparecerá una ventana con un listado de estados entre el que seleccionar. Si, por ejemplo, se retrasa la revisión es recomendable exponer los motivos en los comentarios para que así todos los usuarios puedan estar informados.
- Reportar errores, incoherencias o preguntar dudas a otros usuarios con un clic en **“Query”**. En la versión móvil el sistema pide al usuario que introduzca las direcciones de los destinatarios y abre la aplicación de correo electrónico que esté configurada en el dispositivo, de modo que el cuerpo del mensaje se puede escribir directamente allí.

Por otro lado, los TMAs disponen de menos características adicionales en su módulo ya que debido a las particularidades de su trabajo sólo requieren las opciones **“See New Damages”** y **“See SDT RB”** donde podrán crear nuevos daños, y **“See Work Packages”**, **“See Damage Map”**, **“Upload File”** y **“Query”** que funcionan de la misma manera que en el módulo de ingeniería,

Trabajar con daños: añadir, editar, eliminar y otras acciones

Los daños y las reparaciones que conllevan son el segundo eje central del mantenimiento de una flota. Se pueden visualizar en la pantalla **“Show Damages”** o **“Show SDT RB”** según si son daños nuevos o el resultado de una tarea SDT. Tanto el equipo de ingeniería como los TMAs y los administradores del sistema pueden acceder a ellos desde sus respectivos módulos, seleccionando la opción correspondiente en los desplegables que se generan al hacer clic en una revisión. Es conveniente tener en cuenta que la disposición de la pantalla y las funcionalidades disponibles varían según si se accede desde el módulo de ingeniería o desde el de TMAs.

Sólo el equipo de TMAs, desde la pantalla que muestra la Figura A.7, puede **añadir** daños haciendo clic en **“Create New Damage”** o bien en **“Report Back SDT”** según convenga. A continuación, se abrirá una ventana nueva donde introducir todos los datos. El sistema no permitirá guardar el daño hasta que todos los campos estén completos. En caso de no disponer de cierta información es suficiente escribir un guión, ya que más adelante se puede editar con los datos actualizados. Para completar la inserción del daño en la base de datos el usuario deberá introducir su PIN y, además, el sistema guardará su nombre para asegurar trazabilidad en el futuro.



Description	Dimensions	Cat.	ATA	PN	SN	References	Comments
dent in fan cowl RH eng.2 outer skin	10x18	C	54-20	642-3002-521	8497001	SRM 54-20-00 REV.134	-
dent in forward fuselage skin FR 12-13 STRG 34-36 RH	80x65	C	53-11	N/A	N/A	SRM 53-11- 11-283-012 REV.134	-
Other: missing sealant in fwd	N/A	Cx	51-00	N/A	N/A	-	Sealant replaced

CREATE NEW DAMAGE

Fig. A.7 Show Damages para TMA

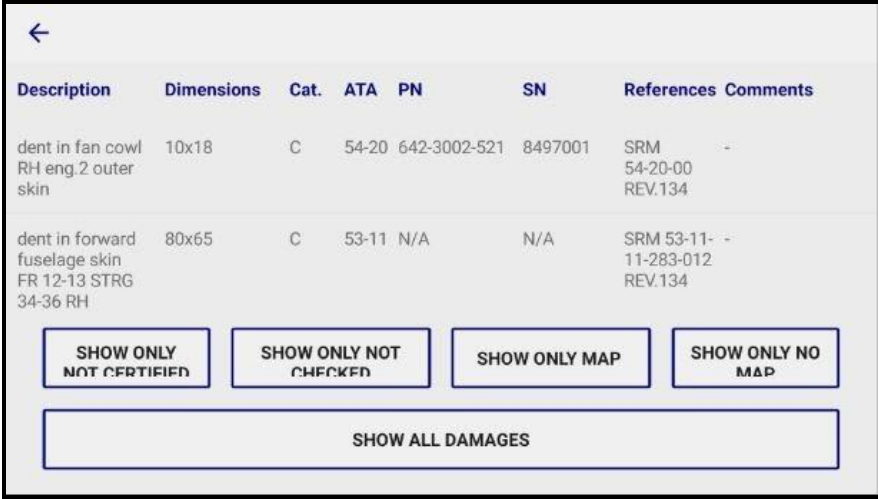
Editar los daños también es una acción reservada a los TMAs. Se puede realizar seleccionando uno en la tabla y haciendo clic en la opción “Edit Damage” del desplegable que aparece. Seguidamente se abrirá una ventana con toda la información que está en ese momento guardada en la base de datos. Una vez estén todos los cambios realizados, y siempre y cuando no queden campos en blanco, el usuario deberá introducir su PIN y el sistema guardará el daño actualizado y su nombre, para así aportar trazabilidad y consistencia al proceso.

Tal y como sucede con las revisiones, los usuarios y la flota, sólo los administradores pueden **eliminar** daños. Para ello deben seleccionar el daño que corresponda y hacer clic en la opción “Delete Damage”. Antes de completar el proceso el sistema preguntará para asegurar que no se trata de una acción no intencionada. Es conveniente, aunque no imprescindible, eliminar la carpeta de Dropbox cuando se elimina un daño porque es posible que el sistema use su identificador para un daño nuevo.

Como **acciones adicionales** los TMA al seleccionar un daño pueden:

- Visualizar información adicional a la que aparece en la tabla con la opción “**Further Information**”.
- Añadir archivos directamente a la carpeta que pertenece al daño haciendo clic en “**Add File**”.
- Visualizar toda la documentación relativa al daño seleccionando “**See Documentation**”.

En la Figura A.8 se muestra la apariencia de la tabla de daños para los usuarios “eng all” y “admin”. Para los usuarios con el rol “eng mro” desaparece la opción “Show Only Not Checked” ya que no entra dentro de sus funciones, mientras que para “eng camo” no es visible el botón “Show Only Not Certified” debido a que no es una de sus tareas.



The screenshot shows a mobile application interface with a table of damage records and filter buttons. The table has columns for Description, Dimensions, Cat., ATA, PN, SN, References, and Comments. Below the table are four filter buttons: 'SHOW ONLY NOT CERTIFIED', 'SHOW ONLY NOT CHECKED', 'SHOW ONLY MAP', and 'SHOW ONLY NO MAP'. At the bottom is a large button labeled 'SHOW ALL DAMAGES'.

Description	Dimensions	Cat.	ATA	PN	SN	References	Comments
dent in fan cowl RH eng.2 outer skin	10x18	C	54-20	642-3002-521	8497001	SRM 54-20-00 REV.134	-
dent in forward fuselage skin FR 12-13 STRG 34-36 RH	80x65	C	53-11	N/A	N/A	SRM 53-11- 11-283-012 REV.134	-

Buttons:

- SHOW ONLY NOT CERTIFIED
- SHOW ONLY NOT CHECKED
- SHOW ONLY MAP
- SHOW ONLY NO MAP
- SHOW ALL DAMAGES

Fig. A.8 Show Damages para ingeniería

Todos los botones que se ven en la Figura A.8 son filtros para visualizar unos daños u otros según sea necesario. “Show Only Not Certified” muestra todos aquellos daños pendientes de certificar mientras que “Show Only Not Checked” muestra los que todavía no han sido revisados por algún miembro de la CAMO. Por otro lado, los botones “Show Only Map” y “Show Only No Map” filtran los daños según si están incluidos en el mapa de daños o no.

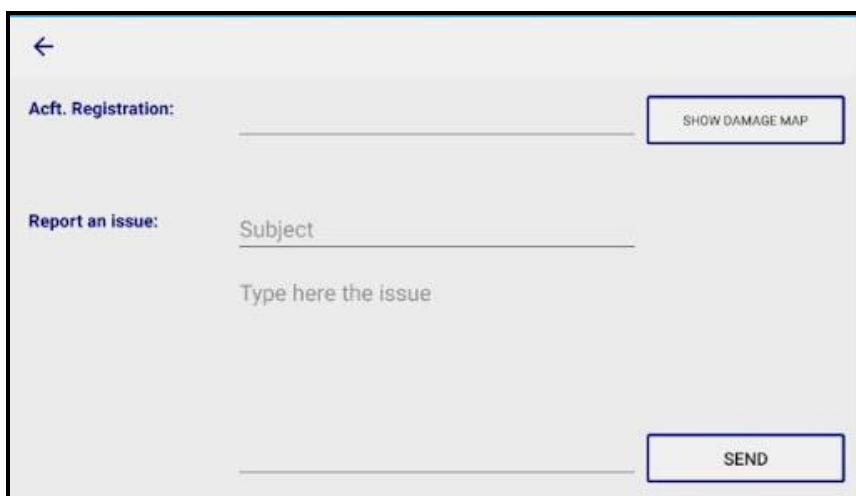
En cuanto a las **acciones adicionales**, el equipo de ingeniería puede:

- Certificar un daño con la opción “**Certify**”. Esta acción la pueden llevar a cabo todos los tipos de ingeniero excepto los de la CAMO. Para completar el proceso de certificación el usuario debe introducir su PIN y el sistema guarda su nombre conforme ha sido él el que ha realizado la acción.
- Revisar un daño con la opción “**Check**”. Esta acción la pueden llevar a cabo todos los tipos de ingeniero excepto los del MRO. Igual que para la certificación, el usuario debe introducir su PIN y el sistema guarda su nombre para así aportar trazabilidad al proceso.
- Visualizar información adicional haciendo clic en “**Further information**”.
- Visualizar toda la documentación relativa al daño con la opción “**See Documentation**”.
- Incluir o excluir un daño del mapa de daños gracias a la opción “**Set As Map / Set As No Map**”.

Módulo de tripulación

Para asegurar que el proceso está del todo digitalizado, las tripulaciones disponen de su propio módulo (Figura A.9). Desde él se puede acceder a los mapas de daños actualizados de toda la flota con sólo seleccionar una matrícula de la lista que aparece al hacer clic en “Acft. Registration”.

Además, también pueden enviar reportes al equipo de ingeniería y TMAs. Igual que las “Queries” de los otros módulos, los mensajes que se generen aquí se enviarán a través de la aplicación de correo electrónico configurada en el dispositivo.



The screenshot displays a mobile application interface for the crew module. At the top left, there is a back arrow icon. Below it, the section 'Acft. Registration:' is followed by a horizontal input field and a button labeled 'SHOW DAMAGE MAP'. Further down, the section 'Report an issue:' is followed by a 'Subject' input field, a larger text area with the placeholder 'Type here the issue', and a 'SEND' button at the bottom right.

Fig. A.9 Módulo de tripulación

Anexo B. Manual de usuario: versión de escritorio

AirDocs es una aplicación para la gestión de la documentación relativa al mantenimiento de una flota de aeronaves. El presente documento expone el funcionamiento de airDocs para ordenadores y busca resolver todas aquellas dudas que puedan surgir durante su uso.

Inicio de sesión

Para iniciar sesión en airDocs es necesario disponer de un nombre de usuario y una contraseña que deberán ser introducidos en la pantalla que se muestra en la Figura B.1. El sistema mostrará un aviso cuando el usuario no exista en la base de datos, la contraseña sea incorrecta y cuando se pierda la conexión con el servidor. En este último caso es posible que tarde un poco más en aparecer la alerta.

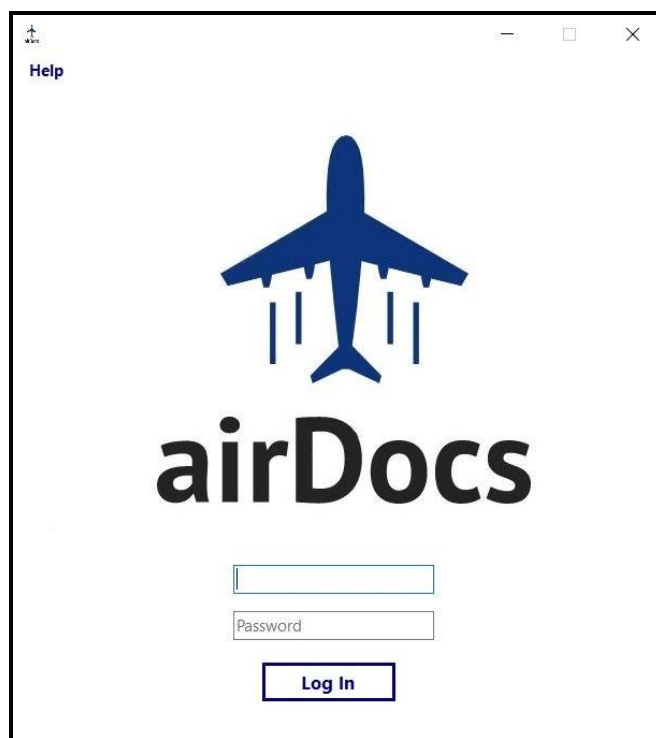


Fig. B.1 Inicio de sesión

En caso de pérdida de contraseña es necesario contactar con los administradores ya que, por motivos de seguridad, sólo ellos pueden proceder a cambiarla.

Trabajar con usuarios: añadir, editar y eliminar

Únicamente los administradores tienen permiso para alterar los datos de los usuarios del sistema. Desde su módulo deben seleccionar la opción “Administrate Users”, y se abrirá una ventana como la que se muestra en la Figura B.2. A la derecha de la misma hay filtros que permiten buscar por nombre de usuario y por posición. Además, en la parte superior de la pantalla está “Refresh table” que permite al usuario actualizar la tabla en cualquier momento, aunque cada vez que se realiza algún cambio el programa lo hace automáticamente.

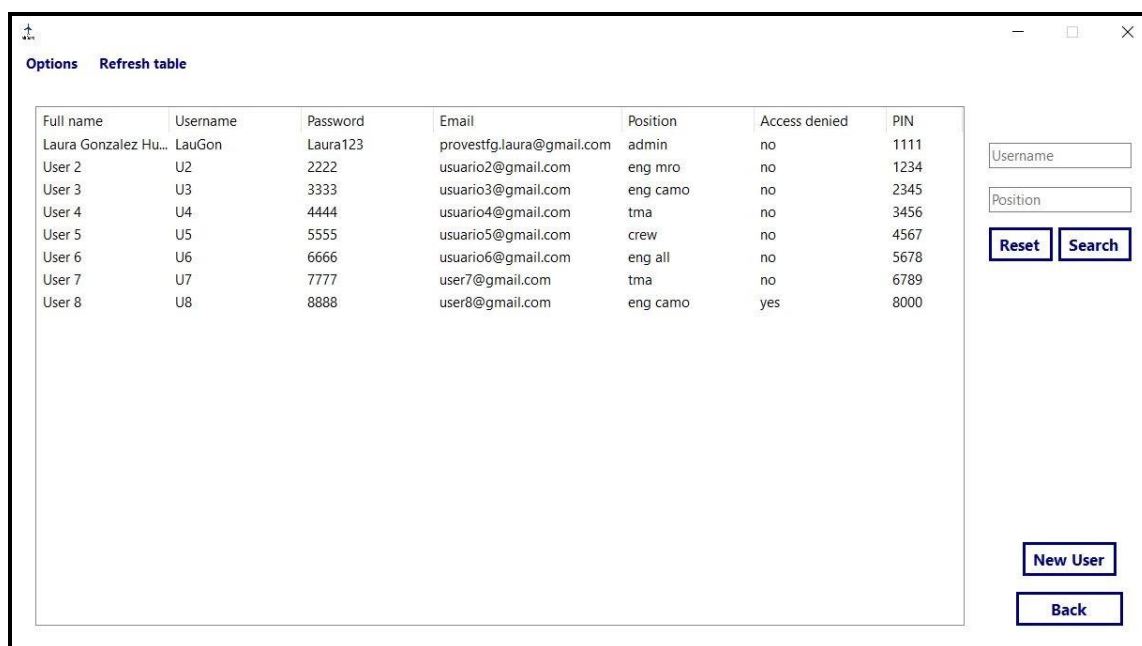


Fig. B.2 Administrar usuarios

Es posible **añadir** un nuevo usuario haciendo clic en “New User” abajo a la derecha. En cambio, para poder **editar** un usuario hay que seleccionarlo en la tabla, hacer clic en el botón “Options” que se encuentra en la parte superior izquierda, y luego elegir la opción “Edit Damage” en el desplegable. En ambos casos se abre una ventana nueva, como la que se muestra en la Figura B.3, con todos los datos requeridos. Si se desea editar un daño aparecerán los campos rellenos con la información que esté guardada en la base de datos.

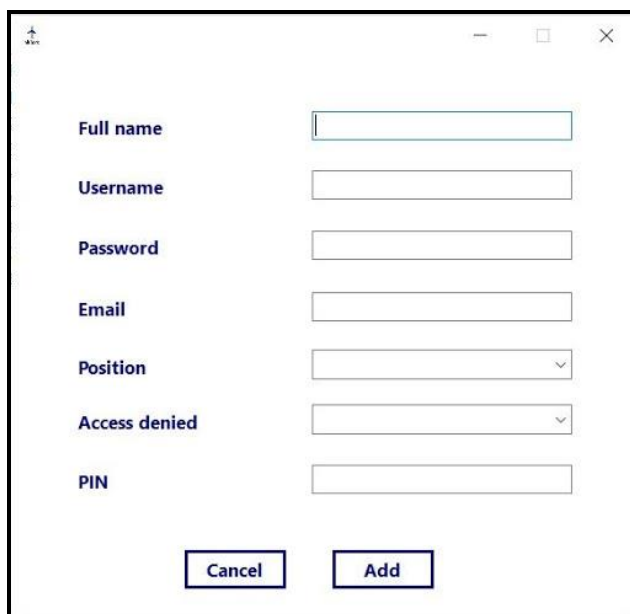
El formulario muestra un cuadro de diálogo con un título de ventana y botones de control. Contiene los siguientes campos: 'Full name' (campo de texto), 'Username' (campo de texto), 'Password' (campo de texto), 'Email' (campo de texto), 'Position' (lista desplegable), 'Access denied' (lista desplegable) y 'PIN' (campo de texto). En la parte inferior hay dos botones: 'Cancel' y 'Add'.

Fig. B.3 Añadir o editar usuarios

Es importante tener en cuenta que el nombre de usuario no se puede editar, ya que al tratarse del identificador del usuario no se guardaría correctamente, y se generaría una entrada nueva en la tabla. Si se desea cambiar el nombre de usuario es necesario crear uno nuevo y eliminar el anterior.

El sistema únicamente procederá a guardar la información una vez esté todo completado, ya que todos los parámetros son necesarios. Para facilitar el proceso, hay campos como "Position" y "Access denied" que disponen de listas desplegables con las opciones entre las que escoger.

Por último, para **eliminar** un usuario es necesario seleccionarlo y en el desplegable de "Options" hacer clic en "Delete User". Antes de proceder el sistema preguntará para asegurar que realmente se quiere llevar a cabo la acción.

Trabajar con aviones: añadir, editar, eliminar y visualizar documentación

Tal y como sucede con los usuarios, sólo los administradores pueden realizar acciones sobre los datos de la flota. Para ello deben acceder a la ventana que se muestra en la Figura B.4, seleccionando "Administrate Fleet" desde su módulo. Una vez aquí pueden filtrar las entradas de la tabla por matrícula y tipo de aeronave, y pueden actualizar los datos gracias a "Refresh Table" situado en la parte superior de la pantalla.

The screenshot shows a web application window titled "Administrar flota". At the top left, there are "Options" and "Refresh table" links. Below these are two input fields labeled "Registration" and "Type", followed by "Search" and "Reset" buttons. The main area contains a table with the following data:

Registration	Manuf. date	Type
EC-AAA	15-05-2015	A320 neo
EC-BBB	16-10-2018	A320 neo
EC-CCC	2015-06-04	A320 neo
EC-DDD	2014-03-12	A320 neo
EC-EEE	2014-12-21	A320 neo

To the right of the table are three input fields: "Acft. Registration", "Manuf. Date", and "Acft. Type". Below these are "Save" and "Cancel" buttons. At the bottom right, there are "Back" and "Add aircraft" buttons. A scrollbar is visible at the bottom of the table area.

Fig. B.4 Administrar flota

Para **añadir** un avión nuevo hay que hacer clic en "Add Aircraft" en la parte inferior derecha, y se mostrarán los campos a rellenar, ya que por defecto no aparecen. Si por el contrario se quiere **editar** un avión existente, hay que seleccionarlo en la tabla, ir a "Options" y hacer clic en "Edit Aircraft" en el desplegable. Esto hará que aparezcan los datos actuales en la parte derecha de la pantalla. Es importante tener en cuenta que, si se cambia la matrícula no se sobrescribirá correctamente, sino que se creará una entrada nueva en la tabla. Esto es debido a que la matrícula es el parámetro que identifica la aeronave y, por tanto, si se desea cambiarla hay que crear un avión nuevo y eliminar el antiguo. Además, el sistema sólo procederá a guardar los cambios cuando estén todos los campos cumplimentados.

Para **visualizar** la **documentación** correspondiente a una aeronave concreta, sólo hay que seleccionarla en la tabla y luego hacer clic en "Options" y "See Documentation". El usuario será redireccionado a la página de Dropbox donde se encuentra la carpeta del avión en cuestión.

Finalmente, para **eliminar** una aeronave hay que seleccionarla en la tabla, hacer clic en "Options" y en el desplegable elegir "Delete Damage". Antes de completar el proceso, el sistema pedirá al usuario que confirme la acción para así asegurar que no se produzca ninguna pérdida accidental de información.

Trabajar con revisiones: añadir, editar, eliminar y otras acciones

Las revisiones son uno de los ejes centrales del mantenimiento de las aeronaves. Se puede acceder a ellas desde los módulos de ingeniería y de TMA y, en consecuencia, también desde el módulo de administración. Sin embargo, sólo pueden añadirlas y editarlas los miembros del equipo de ingeniería y los administradores, quienes adicionalmente pueden eliminar las que convenga.

Para **añadir** una nueva revisión es necesario hacer clic en el botón “Add Check”, visible en la parte inferior derecha del módulo de ingeniería (Figura B.5), mientras que para **editarla** hay que seleccionarla en la tabla, hacer clic en “Options” y luego en “Edit Check” en el desplegable.

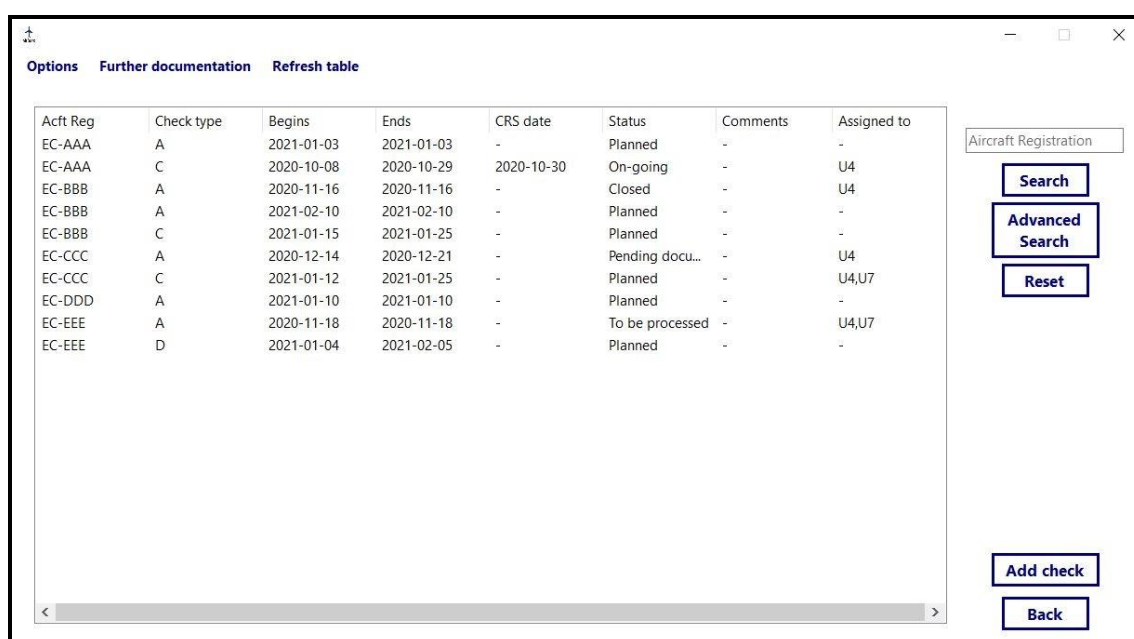
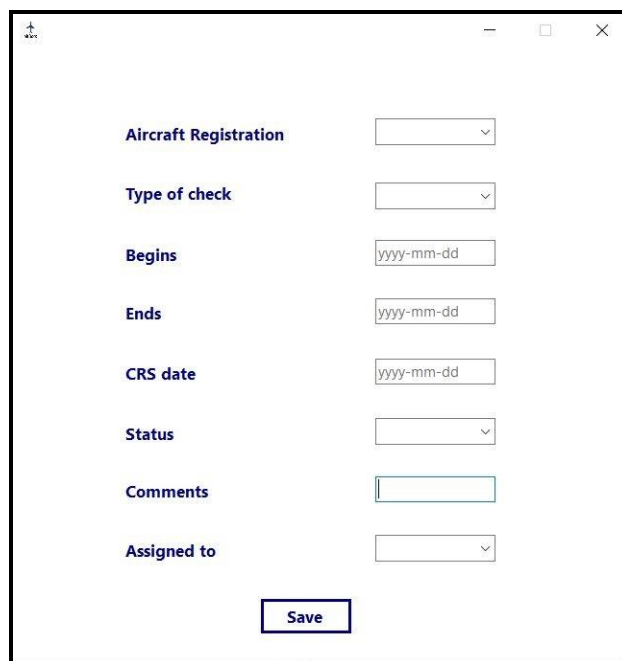


Fig. B.5 Módulo de ingeniería

Tanto para añadir como para editar revisiones, el sistema usa la misma ventana (Figura B.6). Según el caso, aparecerá por completar o con la información almacenada en la base de datos. El sistema sólo procederá a guardar los cambios cuando todos los campos estén completados. Si no se dispone de algún dato, como podría ser la fecha de puesta en servicio (CRS Date), es suficiente con escribir un guion y ya se editará más adelante. Para facilitar el trabajo, la mayoría de los parámetros tienen listas desplegables con opciones entre las que elegir, aunque el usuario siempre puede escribir a mano la información. Es importante recordar que las fechas deben introducirse con el formato “año-mes-día”, y tanto el mes como el día deben ser dos dígitos, por ejemplo 02 y no 2.



The screenshot shows a web form with the following fields:

- Aircraft Registration: dropdown menu
- Type of check: dropdown menu
- Begins: text input field with placeholder 'yyyy-mm-dd'
- Ends: text input field with placeholder 'yyyy-mm-dd'
- CRS date: text input field with placeholder 'yyyy-mm-dd'
- Status: dropdown menu
- Comments: text input field
- Assigned to: dropdown menu

A 'Save' button is positioned at the bottom center of the form.

Fig. B.6 Añadir o editar revisión

El proceso a seguir para **eliminar** una revisión es muy parecido al que hay que seguir para eliminar usuarios o aviones, y también sólo puede ser llevado a cabo por administradores. Lo primero es seleccionar la revisión en la tabla, luego hay que hacer clic en “Options” y finalmente hay que elegir “Delete Check”. El sistema pedirá al usuario una confirmación antes de proceder a suprimir definitivamente los datos.

Las **acciones adicionales** disponibles en el módulo de ingeniería son visibles en hacer clic en “Options” e incluyen las siguientes:

- **“See New Damages”**: Visualizar los daños encontrados durante la revisión (ver siguiente apartado para más detalles)
- **“See SDT RB”**: Visualizar los resultados de las SDT llevadas a cabo (ver siguiente apartado para más detalles)
- **“See Work Packages”**: Visualizar y añadir en Dropbox los paquetes de trabajo con las instrucciones a llevar a cabo.
- **“Upload Files”**: Subir a Dropbox cualquier tipo de archivo. Antes de redireccionar al usuario a la página web, el sistema preguntará en qué carpeta se desea guardar la documentación para así ir directamente a ella.
- **“Assign to TMAs”**: Asignar las revisiones a los TMAs, para que así sepan los trabajos que deben realizar. Aparecerá una ventana con un desplegable con los nombres de todos los mecánicos. Para seleccionar

más de uno hay que introducir sus nombres de usuario separados solamente por una coma.

- **“Update Status”**: Actualizar el estado de la revisión. Al hacer clic en ella aparecerá una ventana con un listado de estados entre el que seleccionar. Si, por ejemplo, se retrasa la revisión es recomendable exponer los motivos en los comentarios para que así todos los usuarios puedan estar informados.
- **“See Damage Map”**: Visualizar el mapa de daños almacenado en la carpeta correspondiente de Dropbox.
- **“Export Damages”**: Genera un archivo de texto (.txt) cuyo nombre es el identificador de la revisión, y que contiene los datos de todos los daños encontrados durante la misma. El formato en el que se muestra la información se puede editar fácilmente realizando pequeños cambios en el código. Esta opción es especialmente útil para cargar los daños en otros sistemas que use la aerolínea sin tener que hacerlo uno a uno.
- **“Query”**: Reportar errores, incoherencias o preguntar dudas a otros usuarios (ver apartado “Reportes y mensajes” para más información).

Además, junto a “Options” se encuentra “Further Documentation” que permite acceder a la documentación de cualquiera de las matrículas de la flota.

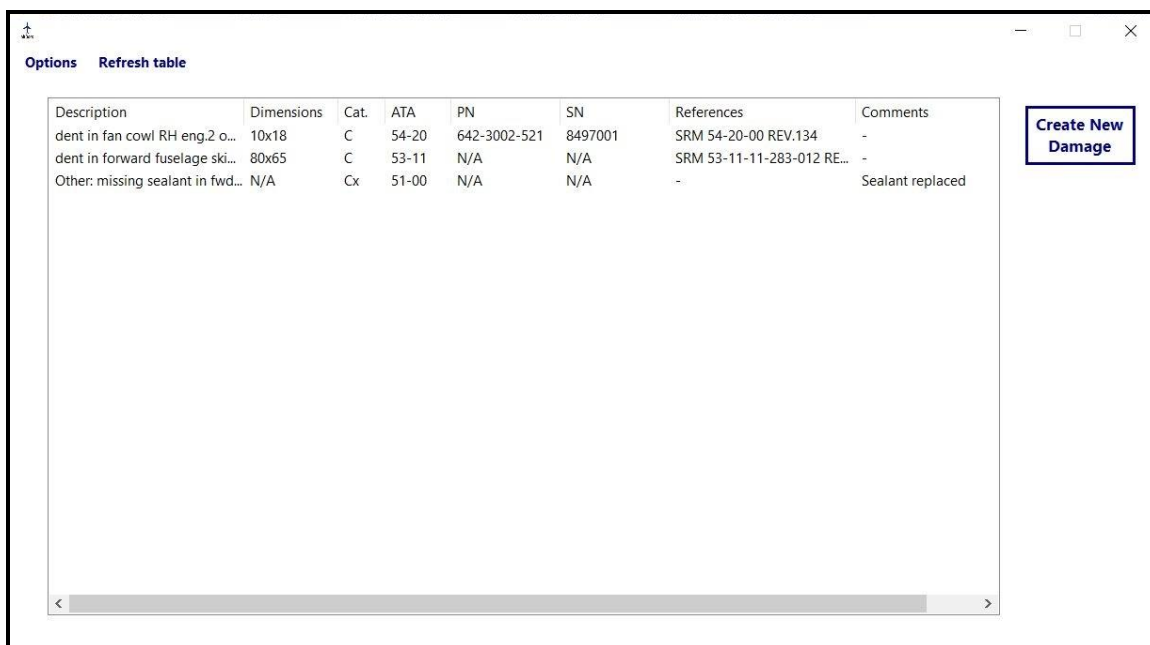
Las opciones adicionales de las que dispone el módulo de TMAs no son tan extensas como las del equipo de ingeniería. Se incluyen **“See Damages”** y **“See SDT RB”** donde podrán crear daños nuevos, y **“See Work Packages”**, **“Upload Files”**, **“Query”** y **“See Damage Map”** que funcionan igual que en el módulo de ingeniería. Adicionalmente, los TMAs pueden acceder a los manuales en línea gracias a **“See Manuals”** que se encuentra junto a **“Options”** en la parte superior de la pantalla.

Trabajar con daños: añadir, editar, eliminar y otras acciones

Los daños y las reparaciones que conllevan son el segundo eje central del mantenimiento de una flota. Se pueden visualizar en la pantalla “Show Damages” o “Show SDT RB”, según si son daños nuevos o el resultado de una tarea SDT. Tanto el equipo de ingeniería como los TMAs y los administradores del sistema pueden acceder a ellos desde sus respectivos módulos, seleccionando la opción correspondiente en los desplegados que se generan al hacer clic en “Options”. Es conveniente tener en cuenta que la disposición de la pantalla y las funcionalidades disponibles varían según si se accede desde el módulo de ingeniería o desde el de TMAs.

Los únicos que pueden **añadir** daños son los miembros del equipo de TMAs. Para ello, deben acceder a la pantalla que se muestra en la Figura B.7 haciendo clic en “Show Damages” o “Show SDT RB” desde su módulo. A

continuación, deben seleccionar “Create New Damage” o “Report Back SDT” según corresponda.



Description	Dimensions	Cat.	ATA	PN	SN	References	Comments
dent in fan cowl RH eng.2 o...	10x18	C	54-20	642-3002-521	8497001	SRM 54-20-00 REV.134	-
dent in forward fuselage ski...	80x65	C	53-11	N/A	N/A	SRM 53-11-11-283-012 RE...	-
Other: missing sealant in fwd...	N/A	Cx	51-00	N/A	N/A	-	Sealant replaced

Fig. B.7 Show Damages para TMA

Seguidamente se abrirá una ventana nueva donde introducir todos los datos (Figura B.8). El sistema sólo procederá a guardar el daño cuando todos los campos estén completos. En caso de no disponer de cierta información es suficiente escribir un guión, ya que más adelante se puede editar con los datos actualizados. Los campos que se pueden dejar con un guión son los que aparecen por defecto de color gris. Para cumplimentar alguno de estos campos hay que seleccionar la caja que hay a su derecha y aparecerá disponible para editar. Además, hay algunos parámetros que tienen listas desplegadas con opciones entre las que elegir. Por último, para completar la inserción del daño en la base de datos, el usuario deberá introducir su PIN y, además, el sistema guardará su nombre para asegurar trazabilidad en el futuro.

Fig. B.8 Añadir o editar daño

Editar los daños también es una acción reservada a los TMAs. Sólo hay que seleccionar el daño en cuestión, ir a “Options” y hacer clic en “Edit Damage”. A continuación, se abrirá una ventana como la de la Figura B.8 con toda la información que está guardada en la base de datos. Una vez estén todos los cambios realizados, el usuario deberá introducir su PIN y el sistema guardará su nombre para así aportar trazabilidad y consistencia al proceso.

Tal y como sucede con las revisiones, los usuarios y la flota, sólo los administradores pueden **eliminar** daños. Para ello deben acceder al módulo de TMAs, visualizar los daños de una revisión concreta, seleccionar el daño que corresponda y hacer clic en la opción “Delete Damage”. Antes de completar el proceso, el sistema pedirá al usuario que confirme la acción para evitar una pérdida accidental de datos. Es conveniente, aunque no imprescindible, eliminar la carpeta de Dropbox cuando se elimina un daño, porque es posible que el sistema use su identificador para un daño nuevo.

Como **acciones adicionales** los TMA pueden elegir entre las siguientes opciones desde el desplegable que se genera al hacer clic en “Options”:

- **“Further Information”**: Permite visualizar en una ventana emergente toda la información del daño seleccionado. Especialmente útil si hay datos que quedan cortados por el tamaño de la tabla.
- **“Add File”**: Para añadir archivos directamente a la carpeta que pertenece al daño.
- **“See Documentation”**: Permite visualizar toda la documentación relativa al daño.

En la Figura B.9 se puede ver la apariencia de la tabla de daños para los usuarios “eng all” y “admin”. Para los usuarios con el rol “eng mro” desaparece la opción “Show Only Not Checked” ya que no entra dentro de sus funciones, mientras que para “eng camo” no es visible el botón “Show Only Not Certified” debido a que no es una de sus tareas.

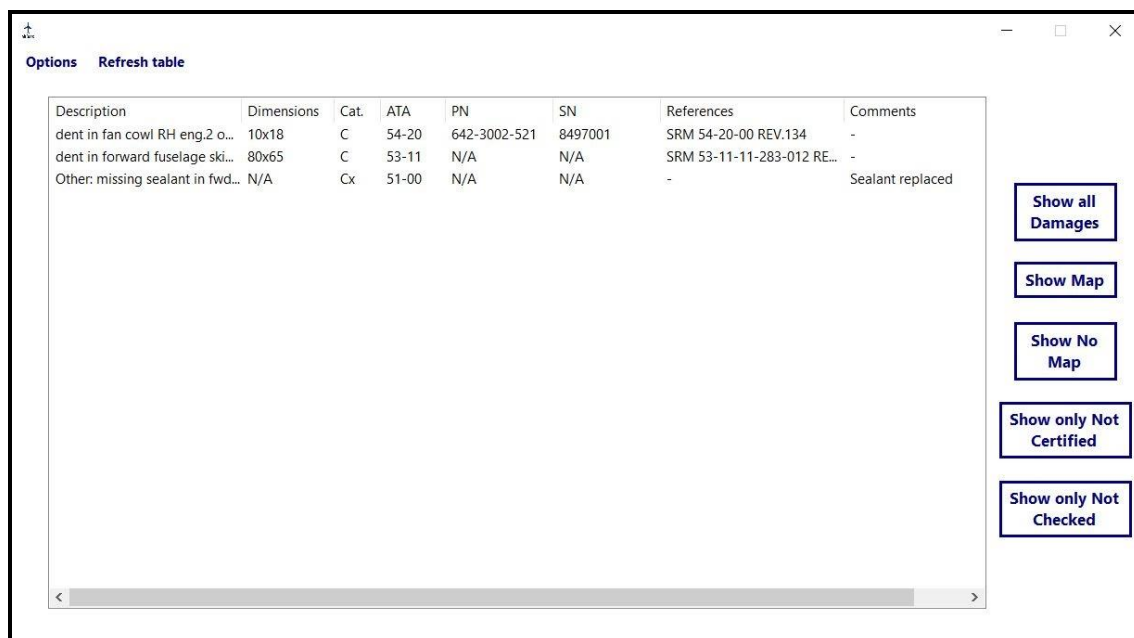


Fig. B.9 Show Damages para ingeniería

Todos los botones que hay en la parte derecha de la pantalla de la Figura B.9 son filtros para visualizar unos daños u otros según sea necesario. “Show Only Not Certified” muestra todos aquellos daños pendientes de certificar, mientras que “Show Only Not Checked” muestra los que todavía no han sido revisados por algún miembro de la CAMO. Por otro lado, los botones “Show Only Map” y “Show Only No Map” filtran los daños según si están incluidos en el mapa de daños o no.

En cuanto a las **acciones adicionales**, el equipo de ingeniería puede elegir entre las siguientes opciones disponibles en el desplegable de “Options”:

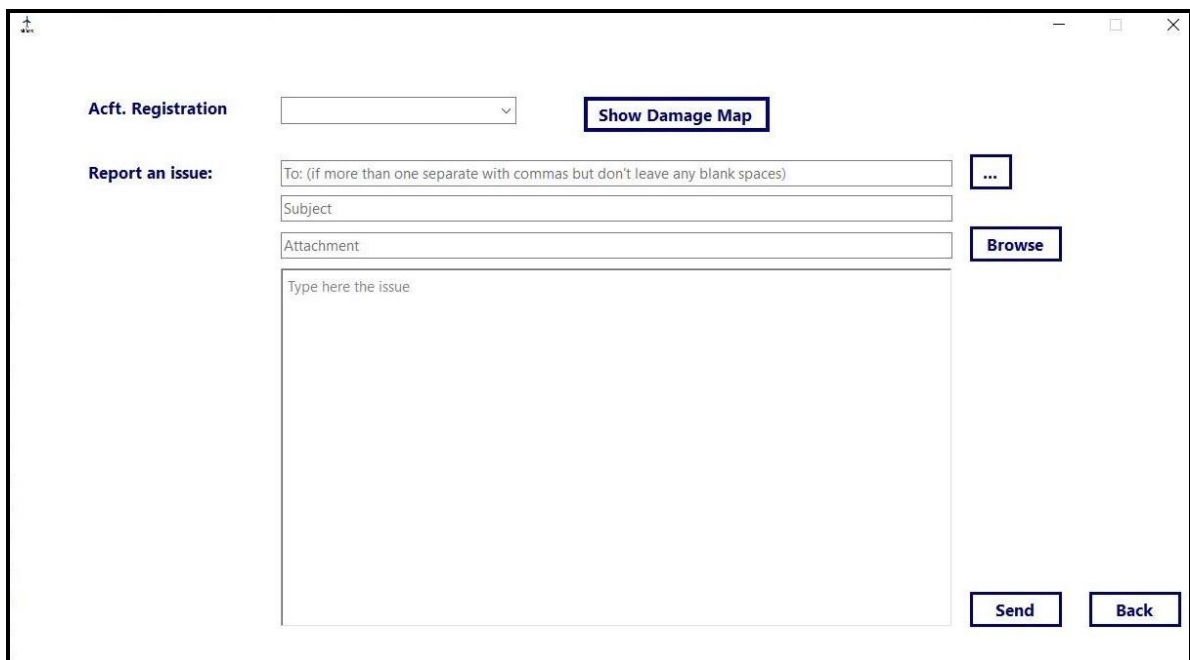
- **“Certify”**: Permite certificar un daño. Esta acción la pueden llevar a cabo todos los tipos de ingeniero excepto los de la CAMO. Para completar el proceso de certificación el usuario debe introducir su PIN, y el sistema guarda su nombre conforme ha sido él el que ha realizado la acción.
- **“Check”**: Permite indicar que un daño ha sido revisado. Esta acción la pueden llevar a cabo todos los tipos de ingeniero excepto los del MRO. Igual que para la certificación, el usuario debe introducir su PIN y el sistema guarda su nombre para así aportar trazabilidad al proceso.

- **“Further information”**: Para visualizar en una ventana emergente toda la información del daño seleccionado. Especialmente útil si hay datos que quedan cortados por el tamaño de la tabla.
- **“See Documentation”**: Permite visualizar toda la documentación almacenada en Dropbox relativa al daño.
- **“Set As Map / Set As No Map”**: Permite incluir o excluir un daño del mapa de daños.

Módulo de la tripulación

Para asegurar que el proceso es del todo digital, las tripulaciones disponen de su propio módulo (Figura B.10). Desde allí se puede acceder a los mapas de daños actualizados de toda la flota, con sólo seleccionar una matrícula del desplegable en “Acft. Registration”.

Además, también pueden enviar reportes al equipo de ingeniería y TMAs. Para obtener más detalles sobre cómo funciona el envío de mensajes a través de la versión de escritorio de airDocs, ver el apartado “Reportes y mensajes”.

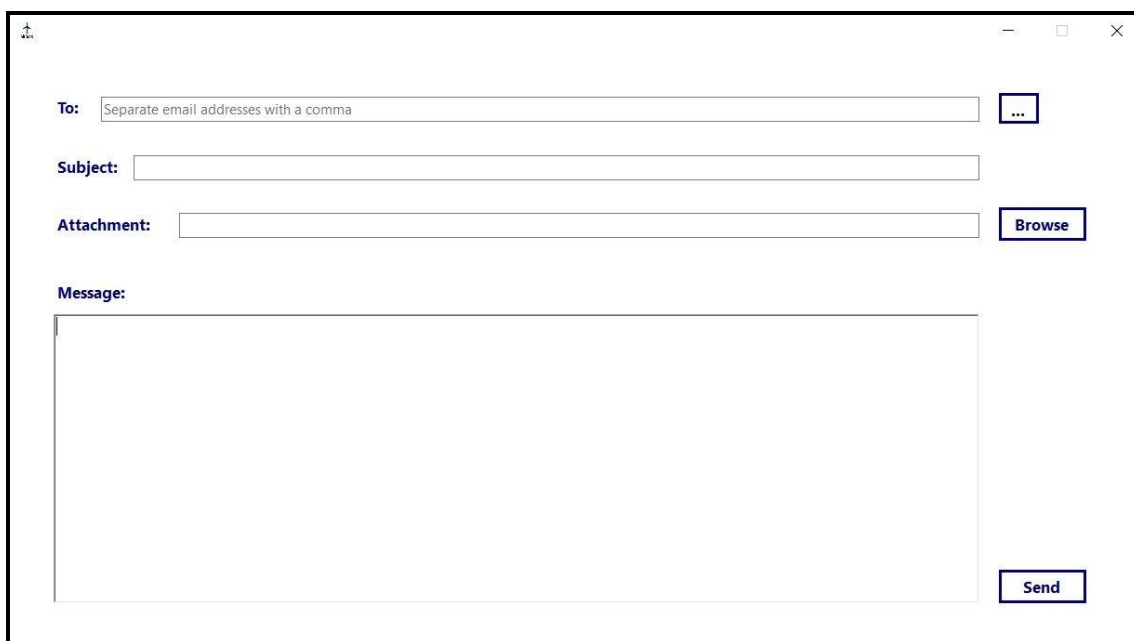


The screenshot shows a web application window titled "Módulo de la tripulación". It features a form for reporting an issue. At the top left, there is a dropdown menu labeled "Acft. Registration" and a button labeled "Show Damage Map". Below this, the "Report an issue:" section contains several input fields: "To: (if more than one separate with commas but don't leave any blank spaces)", "Subject", and "Attachment". To the right of the "To:" field is a button with three dots "...". To the right of the "Attachment" field is a button labeled "Browse". Below these fields is a large text area with the placeholder text "Type here the issue". At the bottom right of the form are two buttons: "Send" and "Back".

Fig. B.10 Módulo de la tripulación

Reportes y mensajes

En la versión de escritorio los mensajes no se envían a través de la aplicación de correo instalada en el dispositivo, sino que airDocs dispone de su propia página para ello (Figura B.11). En esta ventana, el usuario debe introducir las direcciones de los destinatarios separadas únicamente por una coma, el asunto del mensaje y el cuerpo del mismo. Además, es posible adjuntar un archivo mediante el botón “Browse”.

The image shows a web-based form for sending an email. It is contained within a window with standard OS window controls (minimize, maximize, close) in the top right corner. The form has the following fields and buttons:

- To:** A text input field with the placeholder text "Separate email addresses with a comma" and a small square button with three dots to its right.
- Subject:** A text input field.
- Attachment:** A text input field with a "Browse" button to its right.
- Message:** A large, empty text area for the message body.
- Send:** A button located at the bottom right of the message area.

Fig. B.11 Ventana para enviar mensajes desde airDocs

A la derecha de los destinatarios hay un botón que muestra una lista de contactos con los nombres de los usuarios del sistema, su posición y su correo electrónico.

Cuando el usuario haga clic en el botón “Send”, el sistema pedirá que introduzca la contraseña del correo electrónico asociado a su cuenta de airDocs, y procederá a enviarlo. Si la contraseña es incorrecta, el mensaje no se enviará y se mostrará un aviso.

Es posible que airDocs no sea capaz de enviar el mensaje, aunque la contraseña sea correcta. Esto puede ser debido a que la cuenta de correo electrónico no permite el acceso desde aplicaciones externas. Si, por ejemplo, se trata de una cuenta de Google es necesario acceder a “Gestionar tu cuenta de Google” y en el apartado “Seguridad” permitir el acceso.

Anexo C. Código de airDocs

En este anexo se recoge el código implementado que permite la comunicación entre la aplicación y la base de datos. Además, se incluyen las clases que estructuran los datos (ver apartado 4.2).

Aircraft

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AppTFG
{
    public class aircraft
    {
        public string flAcftReg { get; set; }
        public string manufDate { get; set; }
        public string acftType { get; set; }

        public aircraft(string reg, string manuf, string type)
        {
            flAcftReg = reg;
            manufDate = manuf;
            acftType = type;
        }
    }
}
```

Checks

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AppTFG
{
    public class checks
    {
        public string chAcftReg { get; set; }
        public string chType { get; set; }
        public string iniDate { get; set; }
        public string endDate { get; set; }
        public string chID { get; set; }
        public string CRSdate { get; set; }
        public string chStatus { get; set; }
        public string chComments { get; set; }
        public string assignedTo { get; set; }
        public string allProcessed { get; set; }

        public checks(string reg, string type, string start, string end, string
ID, string CRS, string status, string comments, string assigned, string
processed)
        {
            chAcftReg = reg;
            chType = type;
        }
    }
}
```

```

        iniDate = start;
        endDate = end;
        chID = ID;
        CRSdate = CRS;
        chStatus = status;
        chComments = comments;
        assignedTo = assigned;
        allProcessed = processed;
    }
}
}

```

Damages

```

using System;
using System.Collections.Generic;
using System.Text;

namespace AppTFG
{
    public class damages
    {
        public string damAcftReg { get; set; }
        public string IDcheck { get; set; }
        public string damCat { get; set; }
        public string ATA { get; set; }
        public string location { get; set; }
        public string certified { get; set; }
        public string damChecked { get; set; }
        public string TMAsigned { get; set; }
        public string SDTrb { get; set; }
        public string damComments { get; set; }
        public string damType { get; set; }
        public string damDesc { get; set; }
        public string damDimensions { get; set; }
        public string entryDate { get; set; }
        public string closingDate { get; set; }
        public string damReferences { get; set; }
        public string inspInterval { get; set; }
        public string intRepair { get; set; }
        public string PN { get; set; }
        public string SN { get; set; }
        public string damID { get; set; }
        public string signedBy { get; set; }
        public string certifiedBy { get; set; }
        public string checkedBy { get; set; }
        public string map { get; set; }
        public damages(string reg, string checkID, string cat, string ATArefer,
string loc, string cert, string ch, string sgn, string rb, string comm,
string type, string desc, string dim, string entry, string closing,
string refer, string insp, string rep, string pn, string sn, string IDdam,
string signed, string certBy, string check, string isMap)
        {
            damAcftReg = reg;
            IDcheck = checkID;
            damCat = cat;
            ATA = ATArefer;
            location = loc;
            certified = cert;
            damChecked = ch;

```

```

        TMAsigned = sgn;
        SDTrb = rb;
        damComments = comm;
        damType = type;
        damDesc = desc;
        damDimensions = dim;
        entryDate = entry;
        closingDate = closing;
        damReferences = refer;
        inspInterval = insp;
        intRepair = rep;
        PN = pn;
        SN = sn;
        damID = IDdam;
        signedBy = signed;
        certifiedBy = certBy;
        checkedBy = check;
        map = isMap;
    }
}
}

```

Users

```

using System;
using System.Collections.Generic;
using System.Text;

namespace AppTFG
{
    public class usuarios
    {
        public string fullName { get; set; }
        public string userName { get; set; }
        public string userPassword { get; set; }
        public string email { get; set; }
        public string position { get; set; }
        public string blocked { get; set; }
        public string PIN { get; set; }
        public usuarios(string fName, string uName, string pass, string mail,
string pos, string bl, string P)
        {
            fullName = fName;
            userName = uName;
            userPassword = pass;
            email = mail;
            position = pos;
            blocked = bl;
            PIN = P;
        }
    }
}

```


REST service

```

using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;

namespace AppTFG
{
    class RESTservice
    {
        HttpClient client;
        List<usuarios> listaPosition;
        List<checks> listaChecks;
        usuarios usuarioObtenido;
        checks checkObtenida;
        aircraft avionObtenido;

        public RESTservice()
        {
            client = new HttpClient();
            //the api waits 20 seconds for the server to respond
            //if it takes more than that it means that the server is not running
            (a normal response time would be of five seconds)
            client.Timeout = TimeSpan.FromSeconds(20);
        }

        //WORKING WITH USERS

        public async Task<List<usuarios>> getAllUsers()
        {
            //returns all users in database
            List<usuarios> listaUsuarios = new List<usuarios>();

            try
            {
                Uri uri = new
                Uri(string.Format("http://79.147.150.156:3000/api/usuarios", string.Empty)); //IP
                PC server

                HttpResponseMessage response = await client.GetAsync(uri);

                if (response.IsSuccessStatusCode)
                {
                    string content = await response.Content.ReadAsStringAsync();
                    listaUsuarios =
                    JsonConvert.DeserializeObject<List<usuarios>>(content);
                }

                return listaUsuarios;
            }
            catch
            {
                listaUsuarios = null;
                return listaUsuarios;
            }
        }

        public async Task<usuarios> getUser(string uName)
        {

```

```
//returns the data of the user that has a certain username (uName)

try
{
    Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/usuarios/{0}", uName,
string.Empty)); //IP PC server

    HttpResponseMessage response = await client.GetAsync(uri);
    if (response.IsSuccessStatusCode)
    {
        string content = await response.Content.ReadAsStringAsync();
        usuarioObtenido =
JsonConvert.DeserializeObject<usuarios>(content);
    }

    return usuarioObtenido;
}
catch
{
    usuarioObtenido = null;
    return usuarioObtenido;
}
}

public async Task<bool> addUser(usuarios item)
{
    //inserts in the database the given user

    bool successful = false;

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/usuarios", string.Empty)); //IP
PC server

        var json = JsonConvert.SerializeObject(item);
        StringContent content = new StringContent(json, Encoding.UTF8,
"application/json");

        HttpResponseMessage response = null;

        //add user
        response = await client.PostAsync(uri, content);

        if (response.IsSuccessStatusCode)
        {
            Console.WriteLine("User successfully saved.");
            successful = true;
        }
        else
        {
            Console.WriteLine("Oops, something went wrong.");
        }

        return successful;
    }
    catch
    {
        return successful;
    }
}
```

```
    }  
  }  
  
  public async Task<bool> deleteUser(string eliminar)  
  {  
    //deletes from the database a given user  
    bool successful = false;  
  
    try  
    {  
      Uri uri = new  
Uri(string.Format("http://79.147.150.156:3000/api/usuarios/{0}", eliminar,  
string.Empty)); // IP PC server  
  
      var response = await client.DeleteAsync(uri);  
  
      if (response.IsSuccessStatusCode)  
      {  
        successful = true;  
      }  
  
      return successful;  
    }  
    catch  
    {  
      return successful;  
    }  
  }  
  
  public async Task<bool> updateUser(usuarios updated)  
  {  
    //updates in the database the given user  
    bool successful = false;  
  
    try  
    {  
      Uri uri = new  
Uri(string.Format("http://79.147.150.156:3000/api/usuarios/{0}",  
updated.userName, string.Empty)); // IP PC server  
  
      var json = JsonConvert.SerializeObject(updated);  
      StringContent content = new StringContent(json, Encoding.UTF8,  
"application/json");  
  
      var response = await client.PutAsync(uri, content);  
  
      if (response.IsSuccessStatusCode)  
      {  
        successful = true;  
      }  
  
      return successful;  
    }  
    catch  
    {  
      return successful;  
    }  
  }  
  
  public async Task<List<usuarios>> getAll_position(string position)  
  {  
    //returns all users with a specific position
```

```

        try
        {
            Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/usuarios?filter[where][position
][eq]={0}", position, string.Empty)); //IP PC server

            HttpResponseMessage response = await client.GetAsync(uri);

            if (response.IsSuccessStatusCode)
            {
                string content = await response.Content.ReadAsStringAsync();
                listaPosition =
JsonConvert.DeserializeObject<List<usuarios>>(content);
            }

            return listaPosition;
        }
        catch
        {
            listaPosition = null;
            return listaPosition;
        }
    }

    // WORKING WITH CHECKS

    public async Task<List<checks>> getAllChecks()
    {
        //returns all checks in the database
        try
        {
            Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/checks", string.Empty)); //IP
PC server

            HttpResponseMessage response = await client.GetAsync(uri);

            if (response.IsSuccessStatusCode)
            {
                string content = await response.Content.ReadAsStringAsync();
                listaChecks =
JsonConvert.DeserializeObject<List<checks>>(content);
            }

            return listaChecks;
        }
        catch
        {
            listaChecks = null;
            return listaChecks;
        }
    }

    public async Task<bool> addCheck(checks item)
    {
        //inserts in the database the given check
        bool successful = false;
        try
        {

```

```
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/checks", string.Empty)); //IP
PC server

        var json = JsonConvert.SerializeObject(item);
        StringContent content = new StringContent(json, Encoding.UTF8,
"application/json");

        HttpResponseMessage response = null;

        //add user
        response = await client.PostAsync(uri, content);

        if (response.IsSuccessStatusCode)
        {
            successful = true;
        }

        return successful;
    }
    catch
    {
        return successful;
    }
}

public async Task<bool> deleteCheck(string eliminar)
{
    //deletes from database the given check
    bool successful = false;

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/checks/{0}", eliminar,
string.Empty)); // IP PC server

        var response = await client.DeleteAsync(uri);

        if (response.IsSuccessStatusCode)
        {
            successful = true;
        }

        return successful;
    }
    catch
    {
        return successful;
    }
}

public async Task<bool> updateCheck(checks updated)
{
    //updates in the database the given check
    bool successful = false;

    try
    {
```

```

        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/checks/{0}", updated.chID,
string.Empty)); // IP PC server

        var json = JsonConvert.SerializeObject(updated);
        StringContent content = new StringContent(json, Encoding.UTF8,
"application/json");

        var response = await client.PutAsync(uri, content);

        if (response.IsSuccessStatusCode)
        {
            successful = true;
        }

        return successful;
    }
    catch
    {
        return successful;
    }
}

public async Task<checks> getCheck(string checkID)
{
    //returns only the check with the given ID

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/checks/{0}", checkID,
string.Empty)); //IP PC server

        HttpResponseMessage response = await client.GetAsync(uri);
        if (response.IsSuccessStatusCode)
        {
            string content = await response.Content.ReadAsStringAsync();
            checkObtenida =
JsonConvert.DeserializeObject<checks>(content);
        }

        return checkObtenida;
    }
    catch
    {
        checkObtenida = null;
        return checkObtenida;
    }
}

public async Task<List<checks>> getCheckByReg(string registration)
{
    // returns all checks that belong to a certain registration
    List<checks> listaCheckReg = new List<checks>();

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/checks?filter[where][chAcftReg]
[eq]={0}", registration, string.Empty)); //IP PC server

        HttpResponseMessage response = await client.GetAsync(uri);

```

```
        if (response.IsSuccessStatusCode)
        {
            string content = await response.Content.ReadAsStringAsync();
            listaCheckReg =
JsonConvert.DeserializeObject<List<checks>>(content);
        }

        return listaCheckReg;
    }
    catch
    {
        listaCheckReg = null;
        return listaCheckReg;
    }
}

//WORKING WITH THE FLEET

public async Task<List<aircraft>> getAllAircraft()
{
    //returns all aircraft in the database
    List<aircraft> listAircraft = new List<aircraft>();

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/aircraft", string.Empty)); //IP
PC server

        HttpResponseMessage response = await client.GetAsync(uri);

        if (response.IsSuccessStatusCode)
        {
            string content = await response.Content.ReadAsStringAsync();
            listAircraft =
JsonConvert.DeserializeObject<List<aircraft>>(content);
        }

        return listAircraft;
    }
    catch
    {
        listAircraft = null;
        return listAircraft;
    }
}

public async Task<bool> deleteAircraft(string reg)
{
    //deletes a certain aircraft from the database
    bool successful = false;

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/aircraft/{0}", reg,
string.Empty)); // IP PC server

        var response = await client.DeleteAsync(uri);

        if (response.IsSuccessStatusCode)
```

```

        {
            successful = true;
        }

        return successful;
    }
    catch
    {
        return successful;
    }
}

public async Task<bool> addAircraft(aircraft newAcft)
{
    //inserts a given aircraft into the database
    bool successful = false;

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/aircraft", string.Empty)); //IP
PC server

        var json = JsonConvert.SerializeObject(newAcft);
        StringContent content = new StringContent(json, Encoding.UTF8,
"application/json");

        HttpResponseMessage response = null;

        //add aircraft
        response = await client.PostAsync(uri, content);

        if (response.IsSuccessStatusCode)
        {
            Console.WriteLine("Aircraft successfully saved.");
            successful = true;
        }
        else
        {
            Console.WriteLine("Oops, something went wrong.");
        }

        return successful;
    }
    catch
    {
        return successful;
    }
}

public async Task<bool> updateAircraft(aircraft updatedAcft)
{
    //updates the information of the given aircraft
    try
    {
        bool successful = false;
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/aircraft/{0}",
updatedAcft.f1AcftReg, string.Empty)); // IP PC server

        var json = JsonConvert.SerializeObject(updatedAcft);

```



```

        StringContent content = new StringContent(json, Encoding.UTF8,
"application/json");

        var response = await client.PutAsync(uri, content);

        if (response.IsSuccessStatusCode)
        {
            successful = true;
        }

        return successful;
    }
    catch
    {
        //the system gets here when there is no connection to server
        bool successful = false;
        return successful;
    }
}

public async Task<aircraft> GetAircraft(string reg)
{
    //returns the aircraft that has the given registration

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/aircraft/{0}", reg,
string.Empty)); //IP PC server

        HttpResponseMessage response = await client.GetAsync(uri);
        if (response.IsSuccessStatusCode)
        {
            string content = await response.Content.ReadAsStringAsync();
            avionObtenido =
JsonConvert.DeserializeObject<aircraft>(content);
        }

        return avionObtenido;
    }
    catch
    {
        avionObtenido = null;
        return avionObtenido;
    }
}

//WORKING WITH DAMAGES

public async Task<List<damages>> getDamagesFromCheck(string checkID)
{
    // returns all damages that where found during a certain checkç
    List<damages> listaDamCheck = new List<damages>();

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/damages?filter[where][IDcheck][
eq]={0}", checkID, string.Empty)); //IP PC server

        HttpResponseMessage response = await client.GetAsync(uri);
        if (response.IsSuccessStatusCode)

```

```

        {
            string content = await response.Content.ReadAsStringAsync();
            listaDamCheck =
JsonConvert.DeserializeObject<List<damages>>(content);
        }
        return listaDamCheck;
    }
    catch
    {
        listaDamCheck = null;
        return listaDamCheck;
    }
}

public async Task<bool> updateDamage(damages updatedDamage)
{
    //updates the given damage
    bool successful = false;

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/damages/{0}",
updatedDamage.damID, string.Empty)); // IP PC server

        var json = JsonConvert.SerializeObject(updatedDamage);
        StringContent content = new StringContent(json, Encoding.UTF8,
"application/json");

        var response = await client.PutAsync(uri, content);

        if (response.IsSuccessStatusCode)
        {
            successful = true;
        }

        return successful;
    }
    catch
    {
        return successful;
    }
}

public async Task<bool> deleteDamage(string ID)
{
    //deletes the given damage
    bool successful = false;

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/damages/{0}", ID,
string.Empty)); // IP PC server

        var response = await client.DeleteAsync(uri);

        if (response.IsSuccessStatusCode)
        {
            successful = true;
        }
    }
}

```

```
        return successful;
    }
    catch
    {
        return successful;
    }
}

public async Task<bool> addDamage(damages newD)
{
    //inserts into the database the given damage
    bool successful = false;

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/damages", string.Empty)); //IP
PC server

        var json = JsonConvert.SerializeObject(newD);
        StringContent content = new StringContent(json, Encoding.UTF8,
"application/json");

        HttpResponseMessage response = null;

        //add aircraft
        response = await client.PostAsync(uri, content);

        if (response.IsSuccessStatusCode)
        {
            successful = true;
        }

        return successful;
    }
    catch
    {
        return successful;
    }
}

public async Task<List<damages>> getAllDamages()
{
    //returns all damages in database
    List<damages> listDamages = new List<damages>();

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/damages", string.Empty)); //IP
PC server

        HttpResponseMessage response = await client.GetAsync(uri);

        if (response.IsSuccessStatusCode)
        {
            string content = await response.Content.ReadAsStringAsync();
            listDamages =
JsonConvert.DeserializeObject<List<damages>>(content);
        }

        return listDamages;
    }
}
```

```
    }
    catch
    {
        listDamages = null;
        return listDamages;
    }
}

public async Task<List<damages>> getDamagesFromAcft(string acftReg)
{
    // returns all damages in the aircraft with registration acftReg
    List<damages> listaDamAcft = new List<damages>();

    try
    {
        Uri uri = new
Uri(string.Format("http://79.147.150.156:3000/api/damages?filter[where][damAcftRe
g][eq]={0}", acftReg, string.Empty)); //IP PC server

        HttpResponseMessage response = await client.GetAsync(uri);
        if (response.IsSuccessStatusCode)
        {
            string content = await response.Content.ReadAsStringAsync();
            listaDamAcft =
JsonConvert.DeserializeObject<List<damages>>(content);
        }
        return listaDamAcft;
    }

    catch
    {
        listaDamAcft = null;
        return listaDamAcft;
    }
}
}
```