



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

UTILIZACIÓN DE BLOCKCHAIN E INTELIGENCIA ARTIFICIAL PARA EL MANTENIMIENTO DE AVIONES

Autor: Eva Manrique Sanz

Director: Atilano Fernández-Pacheco Sánchez-Migallón

Madrid – Junio 2023

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título Utilización de Blockchain e Inteligencia Artificial para el mantenimiento de aviones en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2022/23 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Eva Manrique Sanz

Fecha: 27/06/2023

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Atilano Ramiro Fernández-Pacheco Sánchez-Migallón

Fecha: 27/06/2023



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

UTILIZACIÓN DE BLOCKCHAIN E INTELIGENCIA ARTIFICIAL PARA EL MANTENIMIENTO DE AVIONES

Autor: Eva Manrique Sanz

Director: Atilano Fernández-Pacheco Sánchez-Migallón

Madrid – Junio 2023

Agradecimientos

A todos mis seres queridos, con especial mención a mi abuelo Alejandro.

UTILIZACIÓN DE BLOCKCHAIN E INTELIGENCIA ARTIFICIAL PARA EL MANTENIMIENTO DE AVIONES

Autor: Manrique Sanz, Eva

Director: Fernández-Pacheco Sánchez-Migallón, Atilano

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas)

RESUMEN DEL PROYECTO

Se ha diseñado un sistema formado por una web, un contrato inteligente, una base de datos y un algoritmo de regresión lineal múltiple. Con él, se realiza una mejora en la seguridad de la información del mantenimiento en aviación, almacenando los informes de revisiones realizadas en la cadena de bloques de Blockchain, de manera que no puedan ser alterados.

Así mismo, se busca evitar el mayor número de inconvenientes o accidentes posibles mediante el planteamiento de la fecha estimada de la próxima revisión de cada avión en función de su situación en la última revisión realizada.

Palabras clave: Blockchain, Hash, Machine Learning, Regresión Lineal Múltiple, Mantenimiento, Avión

1. Introducción

La tecnología Blockchain es un registro digital compartido y descentralizado que permite crear una base de datos inmutable y segura. Cada bloque de información está enlazado criptográficamente con el bloque anterior, garantizando que la información almacenada no pueda ser alterada sin cambiar todos los bloques anteriores. Se utiliza en áreas como contratos inteligentes, automatizando el proceso de negociación y aumentando la eficiencia y transparencia. [1]

Este uso de la tecnología Blockchain es el que se implementará en el mundo del mantenimiento en aviación, mejorando los sistemas actuales, haciéndolos más precisos y con un índice de fallo inferior al actual.

Por otro lado, el Machine Learning es una rama de la Inteligencia Artificial que permite a los ordenadores aprender y tomar decisiones sin ser programados explícitamente. Se basa en el análisis y aprendizaje de datos mediante modelos y algoritmos que reconocen patrones y realizan predicciones. [2]

En este proyecto, se utiliza un algoritmo de regresión lineal múltiple, un enfoque de aprendizaje supervisado. Se proporcionan ejemplos de entrada y salida esperada al modelo durante el entrenamiento. El objetivo es aprender una función lineal que relacione las variables de entrada con la variable de salida, ajustando un hiperplano a los datos de entrenamiento para reducir la diferencia entre las salidas predichas y las reales. Este algoritmo ayuda a estimar la fecha de revisión de las aeronaves y prevenir posibles fallos posteriores.

2. Definición del proyecto

En este Trabajo de Fin de Grado se ha diseñado un sistema tipo Intranet para mejorar en accesibilidad y seguridad, la creación y el almacenamiento de informes derivados del mantenimiento de aeronaves.

Para ello, se ha desarrollado un contrato inteligente, desplegado en Ethereum, que permite a los mecánicos guardar el hash de los informes que se realizan de manera automática, sin más que introducir en la web los resultados obtenidos de cada revisión. El hecho de almacenarlo en Blockchain servirá a la empresa para determinar si el archivo PDF ha sido modificado posteriormente, o se le ha añadido información extra.

Por otro lado, también se ha desarrollado un algoritmo de regresión lineal múltiple que realiza la estimación de la fecha de la próxima revisión de cada aeronave de manera automática, en función de diferentes valores para maximizar la seguridad y minimizar los riesgos de accidente de un avión y sus pasajeros.

Toda esta funcionalidad se encuentra envuelta en una página web accesible para todo tipo de usuarios, en la cual el usuario no tiene por qué tener conocimientos previos ni de tecnología Blockchain ni de Machine Learning, únicamente de aviación.

En conclusión, la funcionalidad del proyecto recoge que se organice y muestre el trabajo a realizar por un mecánico cada día sin necesidad de que alguien lo organice, puesto que recibe alertas en su perfil de la Intranet, que el mecánico (identificado) pueda registrar revisiones realizadas que se almacenen de manera que se asegure su veracidad, que se proponga de manera automática una nueva fecha de revisión que minimice problemas futuros, y por último, que se pueda comprobar la verificación de los informes, contrastándolos con la información almacenada en Blockchain.

3. Descripción del sistema

El Sistema desarrollado en este proyecto se compone de cuatro grandes bloques que interactúan entre sí, como se puede apreciar en la figura inferior: la red Blockchain (Ethereum), una base de datos PostgreSQL, un algoritmo de Machine Learning y la web con la que interactúa directamente cada usuario de la web.

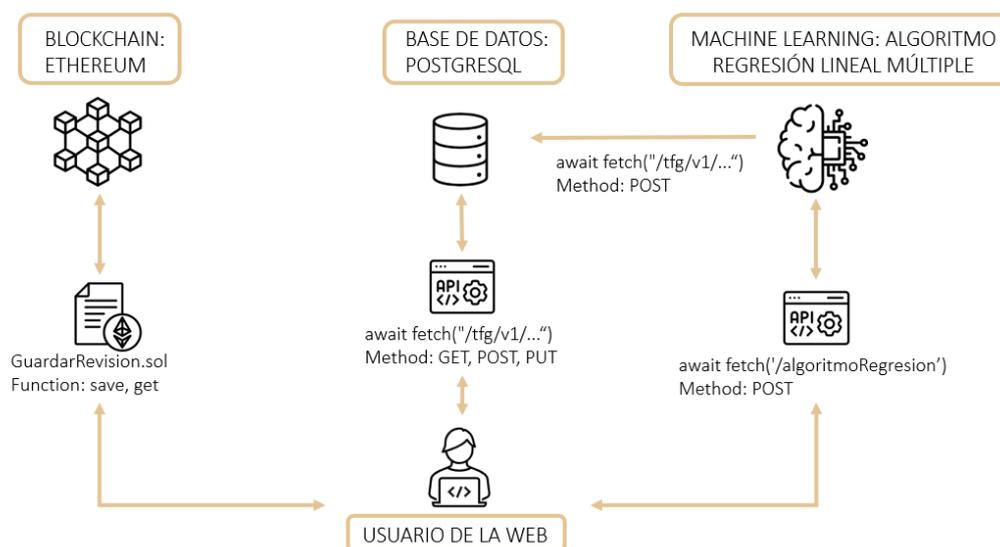


Figura 1. Diseño Arquitectura del Sistema

Para la red Blockchain se ha elegido Ethereum debido a su soporte para Contratos Inteligentes, y Solidity como lenguaje de programación debido a la gran cantidad de documentación disponible. En esta red se almacenará un hash del informe con toda la información susceptible de modificación en caso de accidente. Cada vez que se realice una revisión, el sistema llamará al contrato inteligente, que guardará la información en Blockchain. En cambio, si se realiza una modificación 'ilegal', no se guardará en Blockchain y a la hora de comprobar su veracidad, se revelará que ha sufrido cambios.

En segundo lugar, se ha elegido una base de datos PostgreSQL para almacenar la información de los usuarios, aviones y revisiones, por su compatibilidad con herramientas de administración como PgAdmin, que facilita las consultas y operaciones, principalmente durante el desarrollo de la aplicación.

El tercer bloque es el referente al Algoritmo de Machine Learning. Se ha elegido este algoritmo de aprendizaje supervisado debido a que se dispone de un amplio conjunto de datos de entrenamiento, gracias a los cuales, también se espera una predicción precisa. Así mismo, de manera más concreta, se ha elegido este Algoritmo de Regresión Lineal Múltiple porque se buscaba que la fecha fuera estimada en función de más de una variable.

Por último, se encuentra el bloque referente a la página web con la que el usuario interactúa. Se ha simulado la Intranet de una determinada empresa, en la cual se organiza el planning de los mecánicos mostrándoles alertas de las tareas que deben realizar en el día; donde pueden registrar las revisiones que realizar, ojear la situación del resto de aviones, y, por supuesto, comprobar la veracidad de un informe. Además, debido a que el mantenimiento en aviación es crítico, estos deben estar siempre correctamente identificados (se verifica el DNI). Cabe destacar que en la página web también hay un perfil de Administrador, que gestiona los usuarios, los aviones y los modelos disponibles; pero que no realiza las funciones de mecánico.

4. Resultados

El resultado de este TFG se considera satisfactorio. Se ha desarrollado una web en la cual se crean informes en PDF en base a los resultados de una revisión, se puede asegurar su veracidad mediante interacción con Blockchain, y, además, se realiza una estimación de valores, lo que suma valor al proyecto. Por otro lado, también se organiza la agenda de los empleados de una empresa.

Al tratar con un asunto de suma delicadeza, todas las transacciones quedan correctamente registradas e identificadas, lo que aporta transparencia en la trazabilidad de las revisiones.

En las siguientes imágenes se muestran ejemplos del funcionamiento final de la aplicación:

BLOCKCHAIN AIRWORKS 

ALERTAS LISTADO AVIONES LISTADO REVISIONES NUEVA REVISION COMPROBACIÓN INFORMES PERFIL

Introduce los datos de la revision que hayas realizado

Numero De Serie Del Avión:
35002

Nuevas Horas De Vuelo:
3

Modelo:

Fuselaje: Apto Revisar

Ala Izquierda: Apto Revisar

Ala Derecha: Apto Revisar

Cola: Apto Revisar

Motor: Apto Revisar

Tren De Aterrizaje: Apto Revisar

Cabina Pasajeros: Apto Revisar

Cabina Tripulación Técnica: Apto Revisar

Observaciones:
Limpiar ambas alas

Calcular fecha estimada próxima revisión

La fecha estimada es: 07/07/2023

Aceptar Fecha

Hash guardado en blockchain con transacción:
0a2eebd0c704e88045a108bca82067f1c58533ce705c33297eb77a47d805672
94

Figura 2. Pantalla de Introducir Revisión

Tal y como se ha explicado previamente, al final de esta pantalla aparece la fecha estimada y el hash de la transacción de Blockchain como prueba de que se ha almacenado correctamente.

BLOCKCHAIN AIRWORKS 

ALERTAS LISTADO AVIONES LISTADO REVISIONES NUEVA REVISION COMPROBACIÓN INFORMES PERFIL

Introduce el número de revisión y el PDF que deseas comprobar

ID Revisión:

PDF:
 Informe6.pdf

Figura 3. Pantalla de comprobar informes

En esta pantalla se puede comprobar si el informe PDF coincide con lo almacenado en Blockchain. El sistema mostrará un mensaje con el resultado de la comprobación.

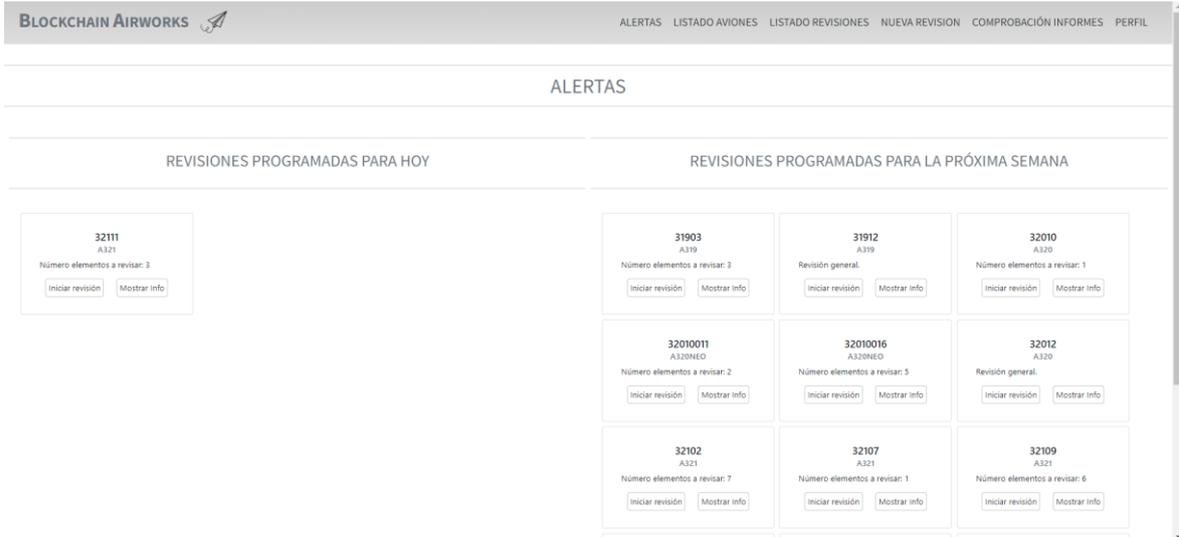


Figura 4. Sistema de alertas de revisiones

Con este sistema de alertas, se organiza la actividad de los mecánicos, pudiendo comprobar qué aviones necesitan revisión en el día y en la próxima semana.

Por otro lado, desde el perfil de administrador, el resultado más importante a destacar es la gestión de usuarios.

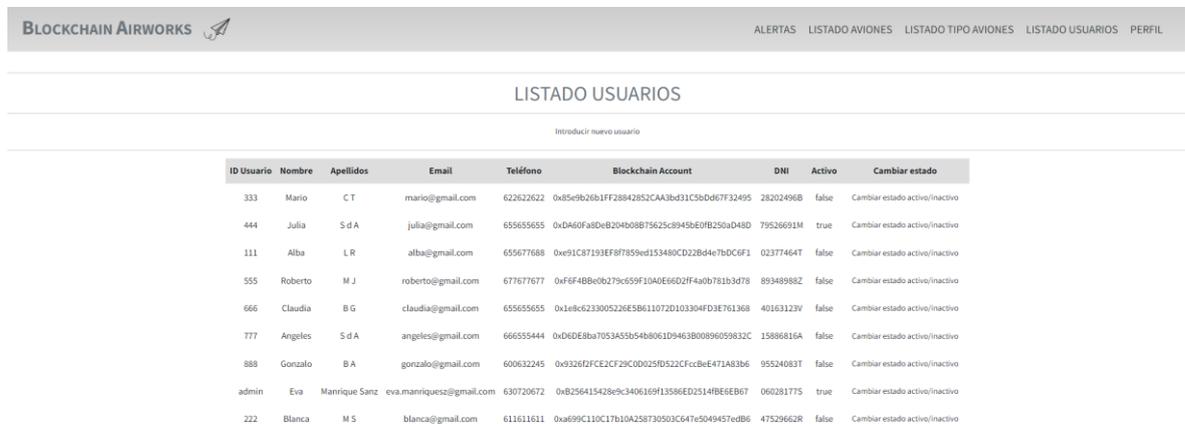


Figura 5. Listado de Usuarios - Perfil Administrador

Desde esta ventana se puede cambiar el estado de los trabajadores (activo-inactivo), para indicar si están trabajando en la actualidad; así como añadir usuarios nuevos a la Intranet.

5. Conclusiones

De la realización del proyecto y de los resultados obtenidos se concluye que se han cumplido los objetivos marcados al inicio, siendo ahora posible almacenar de forma segura los informes de las revisiones de mantenimiento aeronáutico, así como la estimación de fechas de próximas revisiones.

Así mismo, el proyecto, pese a ser completamente funcional, tiene vías para seguir desarrollándose, como la monitorización en tiempo real de los componentes de un avión, integrando IoT (Internet de las Cosas), haciendo uso de sensores de vuelo y otros sistemas de monitoreo, de manera que las predicciones se actualicen de forma dinámica, o estudiar el despliegue de la aplicación en otras redes Blockchain con precio inferior.

6. Referencias

- [1] *IBM - Smart Contracts*. (2022). Obtenido de <https://www.ibm.com/topics/smart-contracts>

- [2] *IBM - Machine Learning*. (2022). Obtenido de <https://www.ibm.com/topics/machine-learning>

USE OF BLOCKCHAIN AND ARTIFICIAL INTELLIGENCE IN AIRCRAFT MAINTENANCE

Author: Manrique Sanz, Eva

Supervisor: Sánchez-Pacheco Fernández-Migallón, Atilano

Collaborating Entity: ICAI – Universidad Pontificia Comillas)

ABSTRACT

The designed system consists of a website, a smart contract, a database and a multiple linear regression algorithm. With it, an improvement is made in the security of aviation maintenance information, storing the reports of revisions carried out in the Blockchain block chain, so that they cannot be altered.

It also seeks to avoid as many inconveniences or accidents as possible by setting the estimated date of the next overhaul of each aircraft based on its situation in the last overhaul performed.

Keywords: Blockchain, Hash, Machine Learning, Multiple Linear Regression, Maintenance, Aircraft

1. Introduction

Blockchain technology is a shared and decentralized digital registry that allows the creation of an immutable and secure database. Each block of information is cryptographically linked to the previous block, ensuring that the information stored cannot be altered without changing all the previous blocks. It is used in areas such as smart contracts, automating the trading process and increasing efficiency and transparency. [1]

It is this use of Blockchain technology that will be implemented in the world of aviation maintenance, improving current systems, making them more accurate and with a lower failure rate than at present.

On the other hand, machine learning is a branch of Artificial Intelligence that allows computers to learn and make decisions without being explicitly programmed. It is based on data analysis and learning through models and algorithms that recognize patterns and make predictions. [2]

In this project, a multiple linear regression algorithm, a supervised learning approach, is used. Examples of expected input and output are provided to the model during training. The goal is to learn a linear function that relates the input variables to the output variable, fitting a hyperplane to the training data to reduce the difference between the predicted and actual outputs. This algorithm helps to estimate the aircraft overhaul date and prevent possible subsequent failures.

2. Definition of the project

In this Final Project, an Intranet-type system has been designed to improve accessibility and security in the creation and storage of reports derived from aircraft maintenance.

To this end, a smart contract has been developed, deployed in Ethereum, which allows mechanics to save the hash of the reports that are automatically made, without having to enter the results obtained from each revision on the web. Storing it on the Blockchain will help the company to determine whether the PDF file has been subsequently modified or extra information has been added to it.

On the other hand, a multiple linear regression algorithm has also been developed that automatically estimates the date of the next revision of each aircraft, based on different values to maximize safety and minimize the risk of an accident for an aircraft and its passengers.

All this functionality is wrapped in a web page accessible to all types of users, in which the user does not need to have previous knowledge of Blockchain technology or Machine Learning, only aviation.

In conclusion, the functionality of the project includes the organization and display of the work to be carried out by a mechanic each day without the need for someone to organize it, as they receive alerts on their Intranet profile, the mechanic (identified) can record the checks carried out and these are stored in a way that ensures their veracity, a new check date is automatically proposed to minimize future problems, and finally, the verification of the reports can be checked, contrasting them with the information stored in Blockchain.

3. System description

The system developed in this project consists of four large blocks that interact with each other, as can be seen in the figure below: the Blockchain network (Ethereum), a PostgreSQL database, a Machine Learning algorithm, and the website with which each user of the website interacts directly.

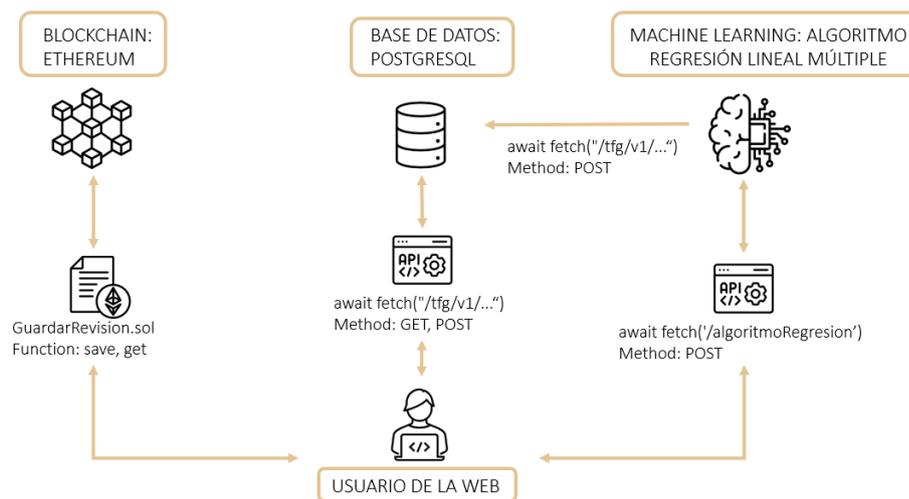


Figura 6. System Architecture Design

Ethereum has been chosen for the Blockchain network due to its support for Smart Contracts, and Solidity as a programming language due to the large amount of documentation available. In this network, a hash of the report will be stored with all the information that can be modified in case of an accident. Every time a revision is made, the system will call the smart contract, which will store the information in Blockchain.

On the other hand, if an 'illegal' modification is made, it will not be stored in Blockchain and when checking its veracity, it will reveal that it has undergone changes.

Secondly, a PostgreSQL database has been chosen to store the information on users, planes and revisions, due to its compatibility with administration tools such as PgAdmin, which facilitates queries and operations, mainly during the development of the application.

The third block refers to the Machine Learning Algorithm. This supervised learning algorithm has been chosen because it has a large set of training data, thanks to which an accurate prediction is also expected. More specifically, this Multiple Linear Regression Algorithm was chosen because the date was to be estimated as a function of more than one variable.

Finally, there is the block referring to the web page with which the user interacts. The Intranet of a certain company has been simulated, in which the mechanics' planning is organized, showing them alerts of the tasks to be carried out during the day; where they can register the revisions to be carried out, check the situation of the rest of the aircraft, and, of course, check the veracity of a report. In addition, because aviation maintenance is critical, they must always be correctly identified (the DNI is checked).

4. Results

The result of this TFG is considered satisfactory. A website has been developed in which PDF reports are created based on the results of a review, their veracity can be ensured through interaction with Blockchain, and, in addition, an estimation of values is carried out, which adds value to the project. On the other hand, the agenda of a company's employees is also organized.

Since we are dealing with a highly sensitive matter, all transactions are correctly recorded and identified, which provides transparency in the traceability of reviews.

The following images show examples of the final operation of the application:

BLOCKCHAIN AIRWORKS  ALERTAS LISTADO AVIONES LISTADO REVISIONES NUEVA REVISION COMPROBACIÓN INFORMES PERFIL

Introduce los datos de la revision que hayas realizado

Numero De Serie Del Avión:

Nuevas Horas De Vuelo:

Modelo:

Fuselaje: Apto Revisar

Ala Izquierda: Apto Revisar

Ala Derecha: Apto Revisar

Cola: Apto Revisar

Motor: Apto Revisar

Tren De Aterrizaje: Apto Revisar

Cabina Pasajeros: Apto Revisar

Cabina Tripulación Técnica: Apto Revisar

Observaciones:

Calcular fecha estimada próxima revisión

La fecha estimada es: 07/07/2023

Aceptar Fecha

Hash guardado en blockchain con transacción:
0a2eebd0c704e88045a108bca82067f1c5853ce705c33297eb77a47d805672
94

Figura 7. Enter Revision screen

As previously explained, at the end of this screen the estimated date and the hash of the Blockchain transaction is displayed as proof that it has been correctly stored.

BLOCKCHAIN AIRWORKS  ALERTAS LISTADO AVIONES LISTADO REVISIONES NUEVA REVISION COMPROBACIÓN INFORMES PERFIL

Introduce el número de revisión y el PDF que deseas comprobar

ID Revision:

PDF:

Figura 8. Check Reports screen

In this screen you can check if the PDF report matches what is stored in Blockchain. The system will display a message with the result of the check.

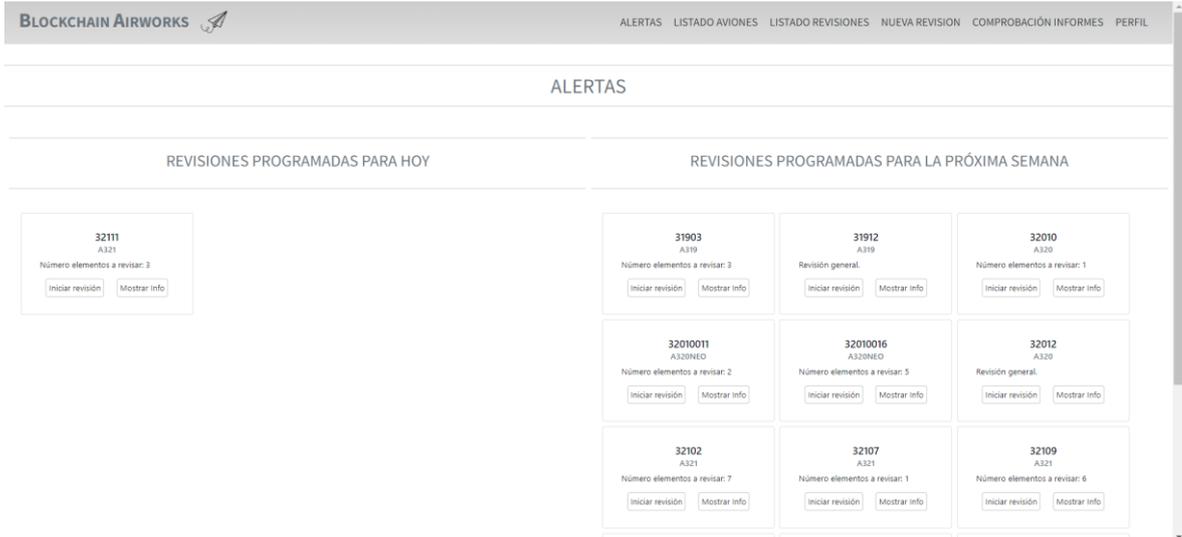


Figura 9. Revision Alert System

With this alert system, the activity of the mechanics is organized, being able to check which aircraft need to be checked on the day and in the next week.

On the other hand, from the administrator profile, the most important result to highlight is user management.



Figura 10. List of Users - Administrator Profile

From this window you can change the status of the workers (active-inactive), to indicate if they are currently working, as well as add new users to the Intranet.

5. Conclusions

From the completion of the project and the results obtained, it can be concluded that the objectives set at the beginning have been met, and it is now possible to securely store the reports of aeronautical maintenance checks, as well as the estimated dates of future checks.

Furthermore, the project, despite being fully functional, has avenues for further development, such as real-time monitoring of aircraft components, integrating IoT (Internet of Things), making use of flight sensors and other monitoring systems, so that

predictions are dynamically updated, or studying the deployment of the application in other Blockchain networks with a lower price tag.

6. References

- [1] *IBM - Smart Contracts*. (2022). Obtenido de <https://www.ibm.com/topics/smart-contracts>
- [2] *IBM - Machine Learning*. (2022). Obtenido de <https://www.ibm.com/topics/machine-learning>

Índice de la memoria

Capítulo 1. Introducción	9
Capítulo 2. Descripción de las Tecnologías.....	12
2.1 Blockchain.....	12
2.2 Ethereum	13
2.2.1 Smart Contracts.....	13
2.3 Aplicación	17
2.3.1 Back-End.....	17
2.3.2 Base De Datos	19
2.3.3 Front-End.....	21
2.4 Machine Learning.....	23
2.4.1 Python.....	23
2.5 Otras Herramientas.....	24
2.5.1 Visual Studio Code	24
2.5.2 Ethereum Remix.....	24
2.5.3 Ganache.....	25
Capítulo 3. Estado de la Cuestión	26
3.1 Blockchain.....	26
3.1.1 Libro Contable Distribuido	27
3.1.2 Red Peer-To-Peer (P2p).....	27
3.1.3 Algoritmo De Consenso.....	28
3.1.4 Criptografía.....	31
3.1.5 Máquina Virtual	35
3.2 Machine Learning.....	35
3.2.1 Proceso De Aprendizaje Automático.....	35
3.2.2 Categorías De Aprendizaje Automático.....	37
3.2.3 Algoritmos De Regresión Lineal	38
3.3 Mantenimiento De Aeronaves.....	39
3.3.1 Norma ISO 9001.....	40
3.3.2 Categorías De Mantenimiento De Aviones	44

3.3.3 Digitalización En Mantenimiento De Aviones	46
3.4 Blockchain En El Mundo De La Aviación	47
3.5 Accidentes De Aviación	47
Capítulo 4. Definición del Trabajo	49
4.1 Justificación.....	49
4.1.1 Seguridad En Verificación Y Certificación	49
4.1.2 Modificación De Informes.....	50
4.1.3 Elección Fecha Revisión	50
4.2 Objetivos	50
4.3 Metodología.....	52
4.4 Planificación.....	52
4.5 Estimación Económica	54
4.5.1 Software.....	54
4.5.2 Recursos Materiales	55
4.5.3 Capital Humano	56
4.5.4 Beneficios	57
Capítulo 5. Sistema/Modelo Desarrollado	61
5.1 Análisis Del Sistema	61
5.1.1 ANÁLISIS DE REQUISITOS.....	61
5.1.2 Características Destacadas.....	64
5.1.3 Casos De Uso	66
5.2 Diseño.....	67
5.2.1 Diseño De La Arquitectura	67
5.2.2 Estructura De Datos.....	71
5.2.3 Diagramas De Secuencia	76
5.3 Implementación	77
5.3.1 Contrato Inteligente.....	78
5.3.2 Base De Datos	80
5.3.3 Algoritmo De Machine Learning.....	82
5.3.4 Página Web	84
Capítulo 6. Análisis De Resultados Y Conclusiones	87
6.1 Primer Objetivo – Blockchain + Pdf	87

6.1.1	Introducir Revisión.....	87
6.1.2	Comprobar Informe.....	89
6.2	Segundo Objetivo – Machine Learning.....	90
6.3	Tercer Objetivo – Implementación Página Web	90
6.4	Cuarto Objetivo – Perfil Administrador	91
Capítulo 7. Trabajos Futuros Y Posibles Áreas De Mejora.....		94
Capítulo 8. Bibliografía.....		96
ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS		102
ANEXO II: Manual De Instalación		105
anexo ii.1	Ganache.....	105
anexo ii.2	Remix Ide.....	106
anexo ii.3	Metamask.....	109
ANEXO III: Manual De Uso		112
Anexo Iii.1	Usuario Sin Autenticar	112
Anexo Iii.1. 1	Inicio Sesión.....	112
Anexo Iii.1. 2	Registro.....	112
Anexo Iii.2	Usuario Mecánico.....	113
Anexo Iii.2.1	Alertas.....	113
Anexo Iii.2.2	Lista Aviones.....	114
Anexo Iii.2.3	Lista Revisiones	115
Anexo Iii.2.4	Introducir Revisión	115
Anexo Iii.2.5	Comprobar Informe	117
Anexo Iii.2.6	Perfil	118
Anexo Iii.3.	Usuario Administrador	118
Anexo Iii.3.1	Alertas.....	119
Anexo Iii.3.2	Listado Aviones.....	119
Anexo Iii.3.2	Listado Revisiones	119
Anexo Iii.3.3	Listado Tipo Aviones.....	119
Anexo Iii.3.4	Listado Usuarios.....	120
Anexo Iii.3.5	Introducir Nuevos Usuarios.....	120

Anexo Iii.3.6 Perfil 121

Índice de figuras

Figura 1. Diseño Arquitectura del Sistema.....	10
Figura 2. Pantalla de Introducir Revisión.....	12
Figura 3. Pantalla de comprobar informes.....	12
Figura 4. Sistema de alertas de revisiones.....	13
Figura 5. Listado de Usuarios - Perfil Administrador	13
Figura 6. System Architecture Design.....	16
Figura 7. Enter Revision screen.....	18
Figura 8. Check Reports screen.....	18
Figura 9. Revision Alert System	19
Figura 10. List of Users - Administrator Profile	19
Figura 11. Icono de Java.....	18
Figura 12. Icono de Python	18
Figura 13. Icono de PostgreSQL	19
Figura 14. Icono de PgAdmin	20
Figura 15. Icono de HTML	21
Figura 16. Icono de CSS.....	22
Figura 17. Icono de JavaScript	22
Figura 18. Icono de StatsModels	23
Figura 19. Icono de ScikitLearn	23
Figura 20. Icono de Visual Studio Code.....	24
Figura 21. Icono de Ethereum Remix.....	24
Figura 22. Icono de Ganache.....	25
Figura 23. Algoritmo de consenso: Proof of Work	29
Figura 24. Algoritmo de consenso: Proof of Stake	31
Figura 25. Criptografía Simétrica.....	32
Figura 26. Criptografía Asimétrica.....	33

Figura 27. Diferencia Hash modificando entrada.....	34
Figura 28. Precio del gas	54
Figura 29. Precio cambio USD - EUR	55
Figura 30. Diagrama de Casos de Uso	66
Figura 31. Diseño de la Arquitectura del Sistema.....	67
Figura 32. Diagrama Entidad Relación Base de Datos	73
Figura 33. Diagrama de Secuencia - Introducir Revisión	76
Figura 34. Diagrama de Secuencia - Comprobar Informe.....	77
Figura 35. Tabla Avión.....	80
Figura 36. Tabla Revisión	81
Figura 37. Tabla Tipo Avión	81
Figura 38. Tabla Usuario	81
Figura 39. Pantalla Introducir Revisión Vacía	88
Figura 40. Permisos para almacenar informe PDF.....	88
Figura 41. Pantalla Introducir Revisión Parte Final	89
Figura 42. Pantalla Comprobar Informe.....	89
Figura 43. Pantalla Final Introducir Revisión	90
Figura 44. Pantalla Mostrar Listado Tipo Aviones	91
Figura 45. Pantalla Mostrar Listado Usuarios	92
Figura 46. Pantalla de Introducir Nuevo Usuario.....	92
Figura 47. Pantalla Perfil de Administrador.....	93
Figura 48. Objetivos de Desarrollo Sostenible (ODS). Fuente: ONU.....	102
Figura 49. Paso 1 Configuración Ganache	105
Figura 50. Paso 2 Configuración Ganache	106
Figura 51. Abrir Remix IDE en VS Code	107
Figura 52. Cómo conectar Remix con Ethereum	107
Figura 53. Cómo compilar contrato en Remix	108
Figura 54. Despliegue del contrato en Remix	109
Figura 55. Lugar del ABI copiado.....	109
Figura 56. Lugar de la dirección del contrato desplegado.....	109

Figura 57. Elegir cuenta Blockchain e Importar Cuenta	110
Figura 58. Pantalla Importar Cuenta.....	111
Figura 59. Pantalla de Inicio de Sesión	112
Figura 60. Pantalla de Registro	113
Figura 61. Pantalla de Alertas	114
Figura 62. Pantalla Mostrar Lista Aviones	115
Figura 63. Pantalla Mostrar Lista Revisiones.....	115
Figura 64. Pantalla Introducir Revisión Vacía	116
Figura 65. Permisos para almacenar informe PDF.....	116
Figura 66. Pantalla Introducir Revisión Parte Final	117
Figura 67. Pantalla Comprobar Informe.....	118
Figura 68. Pantalla de Perfil de Usuario.....	118
Figura 69. Pantalla Mostrar Listado Tipo Aviones	119
Figura 70. Pantalla Mostrar Listado Usuarios	120
Figura 71. Pantalla de Introducir Nuevo Usuario.....	121
Figura 72. Pantalla Perfil de Administrador.....	121

Índice de tablas

Tabla 2. Diagrama de Gantt - Cronograma del proyecto	53
Tabla 3. Estimación Económica Proyecto	57
Tabla 4. Costes escenario Óptimo.	58
Tabla 5. Costes Escenario Realista.....	59
Tabla 6. Costes Escenario Pesimista	59

Capítulo 1. INTRODUCCIÓN

Desde el inicio de la era digital, la tecnología ha evolucionado de manera exponencial, dando lugar a múltiples nuevas tecnologías y herramientas. La creciente capacidad de cálculo de los ordenadores, que ha permitido a ingenieros y desarrolladores producir sistemas cada vez más complejos y sofisticados, es una de las principales causas de esta rápida expansión.

Uno de los cambios más importantes en la historia de la tecnología moderna ha sido el desarrollo de Internet. Ha afectado significativamente a prácticamente todos los aspectos de la vida contemporánea y ha cambiado la forma en que las personas obtienen información y se comunican entre sí. Ha hecho posible la creación de empresas completamente nuevas, la mejora de la eficiencia y la productividad en los negocios existentes, y la expansión del acceso a la educación y el conocimiento.

La evolución de la tecnología también ha tenido un impacto significativo en la aviación, permitiendo que la industria se adapte a las demandas del mercado y a las necesidades de los consumidores. Se han realizado innumerables mejoras en las redes de comunicación, la electrónica de aviación, los sistemas de navegación y los sistemas de control de vuelo. En particular, el desarrollo de sistemas de control electrónico ha hecho que el control de vuelo sea más preciso y eficaz, mejorando la comodidad y la seguridad de los pasajeros. [1]

Asimismo, este impacto se ha visto reflejado en la introducción de sistemas de gestión de reservas y de embarque en línea ha mejorado significativamente la experiencia del pasajero y ha permitido a las aerolíneas ser más eficientes y rentables. Además, la tecnología móvil ha aumentado la comunicación y la claridad entre las aerolíneas y sus clientes al permitir a los pasajeros obtener información en tiempo real sobre vuelos, retrasos y cancelaciones.

En el campo del mantenimiento, uno de los mayores avances ha sido la introducción de sistemas de diagnóstico en tiempo real, que permiten la detección precoz y averías. Estas

herramientas de diagnóstico en tiempo real pueden basarse en sensores que recopilan datos sobre el funcionamiento de los sistemas de la aeronave y los envían a un ordenador central para su evaluación. Esto permite a los mecánicos identificar inmediatamente cualquier problema y realizar las correcciones necesarias antes de que se produzcan fallos graves.

El uso de drones para la inspección y el mantenimiento de aeronaves es otro avance significativo. Los drones pueden volar alrededor de la aeronave y captar imágenes de alta resolución de su superficie, lo que permite detectar la corrosión o el deterioro estructural. También pueden utilizarse para inspeccionar turbinas y otras zonas de difícil acceso de la aeronave.

En cambio, todos los datos recogidos por sensores, drones y mecánicos se guardan de forma insegura, en bases de datos de código abierto que son fáciles de alterar. Por este motivo, el uso de tecnologías de registro distribuido como blockchain puede ser importante. La información puede compartirse y guardarse de forma descentralizada, segura y transparente gracias a la tecnología blockchain. Cada bloque de datos que se sube a la blockchain se cifra y sella, lo que hace imposible cambiarlo o alterarlo sin la aprobación de la red. [2]

De este modo, la integridad de los datos y la trazabilidad están garantizadas mediante el uso de la cadena de bloques para almacenar información sobre el mantenimiento de las aeronaves, ya que cualquier modificación o manipulación sería vista por la red y se reflejaría en los registros de la cadena de bloques. Además, esta tecnología permite la colaboración y potencia la eficacia del mantenimiento aeronáutico al poner de forma transparente toda la información a disposición y acceso de todas las partes implicadas.

Por otro lado, los sensores que se utilizan en los aviones son herramientas útiles para detectar posibles fallos, pero no siempre bastan para detenerlos. En consecuencia, es esencial contar con un algoritmo que tenga en cuenta no solo los datos proporcionados por los sensores, sino también otros factores como el número de horas de vuelo de un avión y su situación actual, el número de elementos que cuentan con deficiencias.

Para crear este algoritmo de prevención de fallos se pueden utilizar técnicas de aprendizaje automático, comúnmente conocidas como machine learning. Lo que se pretende entrenar es un modelo capaz de reconocer patrones y anomalías en los datos recogidos en revisiones previas sobre las horas de vuelo y el número de elementos que han resultado damnificados por el uso, o que son susceptibles de revisión.

Cada variable debe tener umbrales críticos establecidos por el algoritmo para que, cuando una o varias de ellas superen niveles predeterminados, se envíe una alarma al equipo encargado del mantenimiento de la aeronave. Además, debe ser capaz de adaptarse y mejorar con el tiempo a medida que se recopila más información. [3]

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

En este capítulo se describirán las tecnologías y herramientas utilizadas a lo largo del desarrollo del proyecto: Blockchain, Ethereum, Machine Learning.

2.1 BLOCKCHAIN

Blockchain es una tecnología de registro distribuido que permite el almacenamiento y la transmisión segura de información a través de una red descentralizada. Tiene su origen en el año 2008, con la publicación del paper “Bitcoin: un sistema de dinero electrónico peer-to-peer, por Satoshi Nakamoto. [4]

En términos simples, la tecnología Blockchain consiste en una cadena de bloques que contiene información. Cada bloque de la cadena contiene transacciones o información que se ha validado y confirmado por múltiples participantes en la red. Los bloques están enlazados entre sí mediante una técnica criptográfica, lo que garantiza la seguridad y la integridad de la información.

Cada uno de los participantes de dicha red dispone de una copia del registro completo de la cadena de bloques, lo que significa que no hay una única autoridad centralizada que controle la información que se almacena en la red. Esto dificulta a los hackers comprometer la seguridad de la red.

Esta tecnología tiene un gran potencial en múltiples industrias y se espera que siga creciendo en popularidad en los próximos años. En este proyecto se ha utilizado para almacenar cada informe de mantenimiento realizado a cada avión en forma de Hash (previamente calculado a partir del informe), con el objetivo de almacenar toda la información relativa a las revisiones en un lugar seguro, donde no se puedan realizar modificaciones. Esto se consigue y comprueba de dos maneras. En primer lugar, porque la información en Blockchain no se puede modificar, se debe añadir una nueva transacción,

pero siempre quedará constancia de la anterior. Por otro lado, si modifica el informe, a la hora de cotejarlo con el almacenado en Blockchain calculando el hash del informe almacenado en la base de datos y comparándolo, resultarán completamente diferentes, y, por tanto, será fácil advertir esta modificación.

2.2 ETHEREUM



Figura 1. Icono Ethereum

Ethereum es una plataforma Blockchain, nacida en el año 2014 con la publicación del paper de Vitalik Buterin “Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform”. [5] La plataforma utiliza su propia criptomoneda llamada Ether (ETH) para llevar a cabo las transacciones. Esta tecnología es altamente escalable y flexible, lo que permite la creación de tokens personalizados y soluciones Blockchain específicas para las necesidades de los usuarios. Así mismo, permite la creación y ejecución de aplicaciones descentralizadas (dApps) y contratos inteligentes (o Smart Contracts) en la cadena de bloques.

2.2.1 SMART CONTRACTS

Los contratos inteligentes (o Smart Contracts) son programas almacenados en una cadena de bloques que se ejecutan cuando se cumplen condiciones predeterminadas. Generalmente, se utilizan para automatizar la ejecución de un acuerdo para que todos los participantes puedan estar seguros de inmediato del resultado, sin la participación de intermediarios.

Estos contratos se ejecutan en un entorno de ejecución llamado Máquina Virtual Ethereum (EVM, por sus siglas en inglés). La EVM es una máquina virtual que ejecuta contratos inteligentes en un ambiente ‘sandboxed’, lo que significa que se ejecutan de manera segura y aislada del resto de la red. También garantiza que todos los nodos en la red Ethereum

ejecuten los contratos inteligentes de manera idéntica y consistente, lo que garantiza la integridad y la confiabilidad de la red. Por ello, se considera que la EVM es una parte esencial de Ethereum, sin la cual la red no funcionaría. [6]

Para llevar a cabo el desarrollo de un contrato inteligente en Ethereum, se puede realizar la programación en diferentes lenguajes, que van a ser analizados.

2.2.1.1 Solidity



Figura 2. Icono Solidity

Es el lenguaje de programación por excelencia en Ethereum. Está inspirado en lenguajes de programación como C++ y JavaScript. Solidity es un lenguaje de alto nivel, orientado a objetos y de tipo estático. [7]

El hecho de que sea un lenguaje orientado a objetos supone que los contratos escritos en este lenguaje se componen de objetos que pueden interactuar entre sí. Estos pueden tener funciones, variables, eventos y modificadores, lo que permite que los contratos sean personalizados.

Además, Solidity tiene un conjunto de funciones integradas que permiten a los desarrolladores interactuar con la Blockchain de Ethereum. Estas funciones incluyen el envío y recepción de ether, la creación y el control de tokens, y la interacción con otros contratos inteligentes.

Otra característica relevante de Solidity es su enfoque en la seguridad y la protección contra ataques malintencionados. Por ejemplo, requiere que los desarrolladores especifiquen las unidades de medida para todas las variables numéricas, lo que evita errores comunes de desbordamiento que pueden ser explotados por atacantes.

Solidity es compatible con varios entornos de desarrollo integrado (IDE) populares, haciendo posible la creación y prueba de los contratos en el mismo lugar.

Ejemplo: [8]

```
pragma solidity ^0.8.0;
contract ExampleHelloWorld {
    function helloWorld() public pure returns (string memory) {
        return "Hello, World!";
    }
}
```

2.2.1.2 Vyper

Este lenguaje de programación fue creado por Vitalik Buterin en 2017 como una alternativa a Solidity. Está inspirado en Python, tanto su sintaxis como su simplicidad y seguridad. [9]

Esta simplicidad limita la cantidad de funciones y características disponibles, haciendo menos probable que se cometan errores a la hora de escribir el contrato. Tampoco permite la programación orientada a objetos. En definitiva, es menos potente en términos de funcionalidad.

Ejemplo: [10]

```
Greeting: public(bytes[32]) # defining greeting state variable

@public
def __init__(): # initialization function
    self.greeting = "Hello, World."

@public # function can be called internally and externally
@constant # function will not change state
def printGreeting() -> bytes[32]:
    return self.greeting

@public
def setGreeting(_greeting: bytes[32]): # a state changing function
    self.greeting = _greeting
```

2.2.1.3 Rust

Rust es un lenguaje de programación desarrollado por Mozilla en 2010. Su popularidad se debe a su enfoque en la seguridad y su capacidad para manejar tareas de bajo nivel. Esto es, se requiere un alto nivel de conocimientos de programación y contratos inteligentes para utilizarlo. [11]

Es un lenguaje multiparadigma: permite tanto programación orientada a objetos, como programación estructurada y funcional. Una característica importante de este lenguaje es que los errores se detectan en tiempo de compilación, no de ejecución, lo que reduce el riesgo de ataques. Además, utiliza un modelo de concurrencia sin bloqueo, que significa que se pueden ejecutar simultáneamente varias tareas sin bloquearse entre sí.

Ejemplo: [12]

```
fn main() {
    println!("Hello, world!");
}

-- CMD --

> rustc main.rs
> .\main.exe
Hello, world!
```

2.2.1.4 Fe

Es un lenguaje de programación funcional de alto nivel, que basa su sintaxis en la de Python, y está diseñado para ser fácil de aprender. Es el más moderno de los cuatro. Se enfoca en la seguridad y simplicidad, buscando evitar errores comunes. Además, cuenta con verificaciones de seguridad estáticas y dinámicas para garantizar la integridad de los contratos inteligentes que se desarrollan. [13]

Cabe destacar que aún se encuentra en desarrollo activo y no ha sido lanzado oficialmente.

Para la realización del Smart Contract que se va a utilizar en este proyecto se ha decidido utilizar Solidity como lenguaje para programarlo, debido a su funcionalidad, las herramientas para su desarrollo y la abundante documentación existente para realizar el contrato.

2.3 APLICACIÓN

En este apartado se explican las tecnologías y lenguajes de programación utilizados para el desarrollo de la aplicación Intranet que contiene toda la funcionalidad del proyecto. Constará de tres partes: la parte del servidor (Back-end), la parte del almacenamiento de datos (Base de datos) y la parte de la página web y el diseño (Front-end).

2.3.1 BACK-END

El back-end de un proyecto se refiere a la parte del sistema que se encarga de la lógica en la capa de detrás de la interfaz de usuario, el procesamiento de datos y la interacción con la base de datos.

Esta parte se encarga de manejar las solicitudes del cliente, procesar la información que se recibe en el front-end y enviar las respuestas que correspondan. Esto conlleva acceder a la base de datos, efectuar cálculos, autenticar usuarios e implementar la lógica de la aplicación, entre otras.

El back-end se compone de diferentes tecnologías, véase servidores, frameworks y lenguajes de programación. Estas permiten también manejar las solicitudes HTTP. Entre las tecnologías utilizadas están Java, Python, Node.js o PHP, entre otros.

2.3.1.1 Java

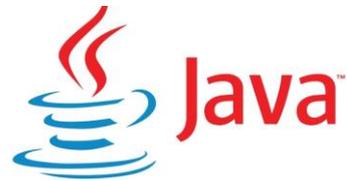


Figura 11. Icono de Java

Java es un lenguaje de programación de alto nivel que se utiliza ampliamente para el desarrollo de software empresarial y de aplicaciones de escritorio. Java es un lenguaje orientado a objetos que se ejecuta en una máquina virtual Java (JVM), lo que significa que el mismo código Java se puede ejecutar en diferentes sistemas operativos sin necesidad de volver a compilarlo. En este proyecto se ha utilizado principalmente para realizar la conexión entre el desarrollo JavaScript y la base de datos. Se ha elegido por encima de otras tecnologías por la portabilidad que ofrece, permitiendo que las aplicaciones se puedan ejecutar en diferentes sistemas operativos; y la amplia cantidad de soporte que ofrece, con una comunidad muy grande. [14]

2.3.1.2 Python



Figura 12. Icono de Python

Python es un lenguaje de programación conocido por su sintaxis sencilla y legible. Destaca por su amplia oferta en cuanto a bibliotecas y frameworks, que facilitan en desarrollo de aplicaciones. Así mismo, Python se integra fácilmente con otras tecnologías, convirtiéndolo así en una opción atractiva. En este proyecto se ha recurrido a la Programación Python por ese mismo motivo. [15]

2.3.1.3 Node.js

Node.js es un entorno de ejecución de JavaScript, que proporciona una plataforma construir aplicaciones de red escalables y de alto rendimiento. A diferencia de otros entornos de programación, proporciona un enfoque orientado a eventos, y sin bloqueo, permitiendo así que se manejen multitud de conexiones simultáneas sin bloquear la ejecución del programa. Además, cuenta con una gran cantidad de módulos y bibliotecas disponibles a través de npm (Node Package Manager). [16]

2.3.1.4 PHP

PHP se considera un lenguaje de scripting de código abierto, ampliamente utilizado para la creación de sitios web dinámicos. Es un lenguaje muy compatible con bases de datos y servidores web, facilitando el proceso de creación de aplicaciones interactivas. Es ampliamente utilizado por usuarios de C y JavaScript por la similitud del lenguaje. [17]

2.3.2 BASE DE DATOS

Como se ha explicado a lo largo de otros apartados previos, el hash de cada informe generado se almacenará en Blockchain para garantizar su inmutabilidad y trazabilidad. En cambio, cada escritura en la cadena de bloques de Blockchain tiene un coste. Es por eso por lo que sólo se almacenará ahí la información esencial, guardando el resto de la información en una base de datos convencional. Para ello, se utiliza PostgreSQL.

2.3.2.1 PostgreSQL



Figura 13. Icono de PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacionales de código abierto y gratuito. Es conocido por su fiabilidad, escalabilidad y capacidad para manejar grandes cantidades de datos. [18]

Entre sus características más destacadas incluye: una alta compatibilidad con el lenguaje SQL avanzado; escalabilidad, pues puede manejar grandes cantidades de datos; seguridad y alta disponibilidad. Además, tiene una amplia comunidad de usuarios y desarrolladores, que contribuyen a su continuo desarrollo; haciendo disponibles numerosas herramientas, bibliotecas y documentación.

Asimismo, con PgAdmin (posteriormente explicado) es posible visualizar de una forma atractiva la información que contiene.

Su uso se ha decidido debido a todas las ventajas previamente mencionadas.

2.3.2.2 PgAdmin



Figura 14. Icono de PgAdmin

PgAdmin es una herramienta de administración de bases de datos PostgreSQL gratuita y de código abierto que proporciona una interfaz gráfica de usuario para realizar tareas comunes de administración de bases de datos, como la creación y edición de tablas, la ejecución de consultas, la gestión de usuarios y permisos, la copia de seguridad y restauración de datos, entre otras. [19]

Proporciona una interfaz gráfica de usuario intuitiva que facilita la gestión de bases de datos PostgreSQL. Además, cuenta con un editor de consultas que permite escribir y ejecutar consultas SQL. Asimismo, permite administrar usuarios y permisos para acceder a la base de datos; y proporciona herramientas para realizar copias de seguridad y restauración de bases de datos.

2.3.3 FRONT-END

El front-end de una página web incluye la parte que el usuario utiliza para interactuar con el servidor. Para ello, se ha desarrollado una página web tipo Intranet. Se puede acceder a ella desde cualquier navegador, facilitando su uso.

Para que una página web sea considerada correcta y bien implementada, debe cumplir los estándares establecidos por W3C (World Wide Web Consortium) sobre el correcto uso de HTML, CSS y JavaScript; debe ser accesible, cargar rápidamente; tener un diseño responsivo y ser segura. [20]

La página web se compone de tres partes: contenido (HTML), estilo (CSS) y funcionalidad (JavaScript y Java).

2.3.3.1 HTML



Figura 15. Icono de HTML

HTML (HyperText Markup Language) es un lenguaje de marcado utilizado para crear páginas web y aplicaciones web. El HTML utiliza etiquetas para definir el contenido de una página web y su estructura. Permite la inclusión de enlaces a otros documentos, formularios interactivos para la entrada de datos y multimedia como video y audio. Además, puede interactuar con bases de datos y servidores. [21]

2.3.3.2 CSS



Figura 16. Icono de CSS

CSS (Cascading Style Sheets) es un lenguaje de diseño utilizado para definir el aspecto visual y el formato de un documento HTML o XML. Con CSS, se pueden definir diferentes estilos para diferentes elementos HTML, como el color, la fuente, el tamaño, la posición y el diseño de la página. Además, CSS permite separar la presentación del contenido, lo que significa que se pueden hacer cambios en el diseño sin afectar el contenido del documento. [22]

2.3.3.3 Javascript



Figura 17. Icono de JavaScript

JavaScript es un lenguaje de programación de alto nivel, orientado a objetos, y débilmente tipado. Se utiliza principalmente para crear aplicaciones web interactivas. Se ejecuta en el navegador del usuario y permite agregar interactividad a una página web, como validación de formularios, efectos visuales, animaciones y cambios dinámicos en el contenido. [23]

2.4 MACHINE LEARNING

El machine learning o aprendizaje automático es una tecnología que se enfoca en que los sistemas informáticos aprendan y mejoren automáticamente a partir de unos datos, sin ser específicamente programados para ello. Los algoritmos de machine learning son capaces de detectar patrones, y decidir en base a estos. En definitiva, permite automatizar tareas y mejorar la toma de decisiones basada en datos. [3]

2.4.1 PYTHON

Python es uno de los lenguajes de programación más populares y ampliamente utilizados en el campo del machine learning. Ofrece una amplia gama de bibliotecas y frameworks específicamente diseñados para el desarrollo de modelos de aprendizaje automático. Una de las principales bibliotecas de Python para machine learning es scikit-learn. [24]

2.4.1.1 Statsmodel



Figura 18. Icono de StatsModels

Es una biblioteca de Python utilizada para realizar análisis estadísticos y modelado de datos, incluyendo, entre otros, modelos como regresión lineal simple y múltiple. Asimismo, ofrece herramientas de visualización de resultados, generando gráficos, diagramas de dispersión, o gráficos de residuos. [25]

2.4.1.2 Scikit-Learn



Figura 19. Icono de ScikitLearn

También conocido como sklearn, es una librería de aprendizaje automático que proporciona una amplia gama de algoritmos para realizar diferentes tareas como regresión o clustering. Entre sus ventajas, se encuentra la consistencia en la interfaz, o la integración con otras librerías de Python como Numpy o Pandas. [26]

2.5 OTRAS HERRAMIENTAS

2.5.1 VISUAL STUDIO CODE



Figura 20. Icono de Visual Studio Code

Visual Studio Code es un editor de código fuente gratuito y de código abierto desarrollado por Microsoft. Visual Studio Code incluye características útiles para la edición de código, como el resaltado de sintaxis, el autocompletado de código, la depuración integrada, la integración con Git o la vista previa en vivo. Sin embargo, se caracteriza principalmente porque se pueden instalar todo tipo de extensiones para agregar funcionalidades adicionales y personalizar la experiencia de programación. Una de las más importantes que se utilizan en este proyecto es Ethereum Remix. [27]

2.5.2 ETHEREUM REMIX



Figura 21. Icono de Ethereum Remix

La extensión Ethereum Remix para VS Code es una herramienta de programación que permite escribir, probar y desplegar contratos inteligentes en la red Ethereum directamente desde el entorno de desarrollo integrado (IDE) de Visual Studio Code. La extensión

proporciona una interfaz gráfica de usuario (GUI) para interactuar con los contratos inteligentes, así como una amplia gama de características y herramientas para ayudar a los desarrolladores a codificar y depurar contratos de manera más eficiente. Entre las características incluyen la capacidad de conectarse a varias redes de prueba de Ethereum para probar contratos en diferentes entornos. [28]

2.5.3 GANACHE



Figura 22. Icono de Ganache

Ganache es una herramienta de línea de comandos que proporciona un entorno de prueba local para el desarrollo de aplicaciones descentralizadas (dApps) basadas en Ethereum. Ganache crea una red Ethereum privada que se ejecuta localmente en la máquina del desarrollador, lo que permite probar y depurar contratos inteligentes de manera rápida y eficiente sin tener que interactuar con la red principal de Ethereum. La herramienta también ofrece una interfaz gráfica de usuario (GUI) fácil de usar para ver detalles de la red, como direcciones de contrato, eventos y transacciones. Además, Ganache permite a los desarrolladores personalizar la configuración de la red, como el número de nodos y el equilibrio de las cuentas, para adaptarse a las necesidades de su proyecto. [29]

Capítulo 3. ESTADO DE LA CUESTIÓN

En este capítulo se explicarán los trabajos y soluciones existen en el ámbito de este proyecto. Se comenzará explicando el estado actual de la tecnología principal del proyecto, Blockchain, así como de la tecnología Machine Learning, también utilizada durante el proyecto, para, posteriormente, explicar en qué situación se encuentra actualmente el mantenimiento de aeronaves.

3.1 BLOCKCHAIN

Como se ha explicado en capítulos anteriores, Blockchain es una tecnología de registro distribuido que permite la creación de una base de datos segura e inmutable en la que se registran transacciones y eventos de forma descentralizada, sin necesidad de intermediarios. Cada registro se almacena en bloques y se enlaza criptográficamente al siguiente, formando una cadena de bloques o "blockchain".

Desde que se popularizó en el año 2008 con su aplicación a las criptomonedas (Bitcoin) [4], esta tecnología ha ido evolucionando y se le han dado diferentes usos fuera de las criptomonedas. Entre estos usos destaca el proyecto Food Trust [30], una plataforma creada por la empresa IBM para rastrear el origen y la calidad de los alimentos. Food Trust permite a los participantes de la cadena de suministro, como agricultores, procesadores, distribuidores y minoristas, compartir información sobre los alimentos en tiempo real, lo que ayuda a identificar rápidamente y a resolver los problemas de seguridad alimentaria, así como a mejorar la eficiencia y reducir los costos.

Otro ejemplo en el ámbito de salud es la plataforma Factom. [31] Su aplicación permite a los proveedores de atención médica registrar y compartir información de forma segura, sin necesidad de una entidad centralizada que gestione los datos. Factom también ayuda a garantizar la precisión y la integridad de los registros médicos, lo que puede mejorar la atención médica y reducir los costos asociados con errores de registro y fraudes.

En definitiva, Blockchain es una tecnología que ofrece una gran cantidad de usos y aplicaciones en distintos ámbitos. A continuación, se explicará en detalle cada uno de los componentes que hacen posible la implementación de esta tecnología. [32] [33]

3.1.1 LIBRO CONTABLE DISTRIBUIDO

Un libro contable distribuido es esencialmente una base de datos actualizada que contiene todas las transacciones realizadas. Se compone de varios bloques, cada uno de los cuales contiene al menos una transacción. Estos bloques están conectados por una cadena que emplea criptografía. En otras palabras, las identificaciones criptográficas del bloque precedente estarán contenidas en el bloque posterior. Por lo tanto, todos los bloques de la parte posterior de la cadena se verán afectados si algún bloque del pasado experimenta alguna modificación o dificultad.

Cada nodo de la red posee una copia completa de este libro contable, lo que garantiza que no haya una sola entidad que lo controle; y cada transacción se registra en todas las copias. Cabe destacar que la información sólo se guarda en el libro contable si se alcanza un consenso.

3.1.2 RED PEER-TO-PEER (P2P)

Una red peer-to-peer (P2P) es un paradigma descentralizado de comunicación entre varios usuarios, también denominados nodos, que es independiente de cualquier servidor central u otros nodos. Cada participante en una red P2P puede funcionar como cliente y como servidor. Esto indica que, una vez establecida la red, cada miembro posee una copia del libro contable. Sin la ayuda de un intermediario, puede utilizarse para distribuir y almacenar archivos.

A su vez, existen varios tipos de red peer-to-peer según su estructura y la forma en la que se gestionan las conexiones y el intercambio de recursos.

3.1.2.1 Redes P2P Centralizadas

En este tipo de red, existe un servidor central que actúa como intermediario entre los nodos. Los nodos se conectan al servidor central y se comunican entre sí a través de él. Ejemplos de este tipo de red son Napster y Kazaa.

3.1.2.2 Redes P2P Descentralizadas

En este tipo de red, los nodos se conectan directamente entre sí sin un servidor central. Los nodos se organizan en una red distribuida y se comunican directamente entre sí. Esta es la utilizada por Blockchain ya que asegura que no haya un punto único de control o fallo en la red, lo que aumenta la seguridad y la resistencia de la blockchain. Otros ejemplos de este tipo de red son Gnutella y eMule.

3.1.2.3 Redes P2P Híbridas

Este tipo de red combina características de las redes P2P centralizadas y descentralizadas. Los nodos se conectan a un servidor central para encontrar otros nodos y después se comunican directamente entre sí. Ejemplos de este tipo de red son BitTorrent y Skype.

3.1.3 ALGORITMO DE CONSENSO

El mecanismo de consenso establece un conjunto de directrices para que los nodos de la red peer-to-peer puedan coordinar sus esfuerzos y llegar a un acuerdo sobre qué transacciones son válidas y están cualificadas para añadirse a la cadena de bloques mediante contratos inteligentes. El verdadero estado de la cadena de bloques se determina mediante el mecanismo de consenso. Cada tipo de red Blockchain tendrá su propio tipo de mecanismo de consenso. [34]

3.1.3.1 Proof-Of-Work (PoW)

Este algoritmo es llevado a cabo por mineros. Para crear el bloque siguiente, los mineros (también conocidos como nodos) cooperan para resolver un rompecabezas criptográfico. La red Blockchain llegará a un consenso, permitirá al primer minero seleccionar el bloque que se incluirá y le concederá la recompensa adecuada.

La complejidad del problema matemático se ajusta automáticamente para que el tiempo promedio necesario para resolver un bloque sea constante. Esto significa que, si hay más nodos compitiendo por resolver el problema, la dificultad del problema aumenta para mantener constante el tiempo promedio necesario para resolver un bloque. Por ello, los mineros necesitan ordenadores potentes.

La idea detrás del algoritmo PoW es que requiere mucho trabajo y recursos para resolver un bloque, por lo que es poco probable que alguien intente falsificar una transacción o alterar la cadena de bloques después de que una transacción ha sido confirmada. Cuanto más tiempo ha pasado desde que se confirmó una transacción, más difícil sería alterarla, ya que se necesitaría una cantidad cada vez mayor de potencia de cómputo para resolver los bloques necesarios para revertir la transacción.

Sin embargo, el algoritmo PoW tiene algunos inconvenientes. En primer lugar, requiere mucha energía para resolver los problemas matemáticos necesarios para agregar un bloque a la cadena de bloques, lo que hace que la minería de criptomonedas basadas en PoW sea muy costosa y consume mucha energía. Además, el algoritmo PoW puede ser vulnerable a ataques del 51%, donde un grupo de nodos controla más del 50% de la potencia de cómputo de la red y puede manipular la cadena de bloques a su antojo.

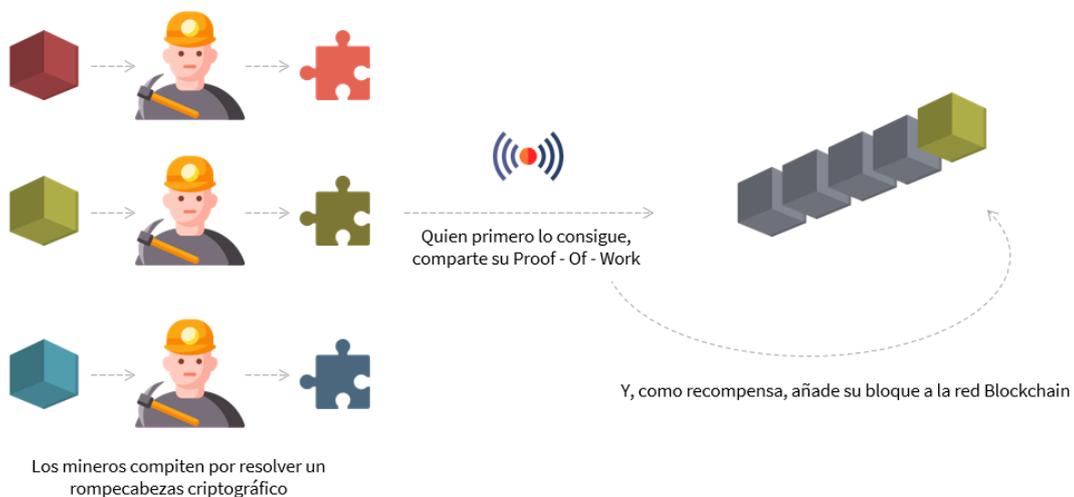


Figura 23. Algoritmo de consenso: Proof of Work

3.1.3.2 Proof-Of-Stake (PoS)

Cuando deben confirmarse varios tokens, se utiliza la expresión "PoS" para agilizar el proceso de minería. De acuerdo con la normativa PoS, los mineros deben demostrar que son propietarios de un porcentaje de acciones antes de emprender la correspondiente actividad de minería de dichas acciones. Como resultado, se ahorra más energía (electricidad) y gastos operativos.

Este algoritmo también busca solucionar el problema de la posibilidad de ataques del 51% de los nodos. En lugar de requerir que los nodos compitan para resolver problemas matemáticos complejos, como ocurre en el algoritmo PoW, el algoritmo PoS utiliza un sistema de selección aleatoria que otorga el derecho de validar transacciones y agregar bloques a la cadena de bloques a los nodos que poseen una cierta cantidad de la criptomoneda en cuestión (en este caso, Ethereum).

Esto significa que cuanta más criptomoneda posea un nodo, más probabilidades tendrá de ser seleccionado para validar transacciones y recibir recompensas. Además, el algoritmo PoS utiliza un sistema de castigo (llamado "slashing") para desincentivar a los nodos malintencionados que intentan manipular la cadena de bloques. Si un nodo intenta validar transacciones fraudulentas o actuar de manera maliciosa, puede perder parte de su criptomoneda como castigo.

Sin embargo, el algoritmo PoS también tiene algunas desventajas. En particular, algunos críticos argumentan que el algoritmo PoS favorece a los nodos más ricos y puede conducir a una mayor centralización en la red, ya que los nodos con más criptomoneda tendrán más poder de decisión.

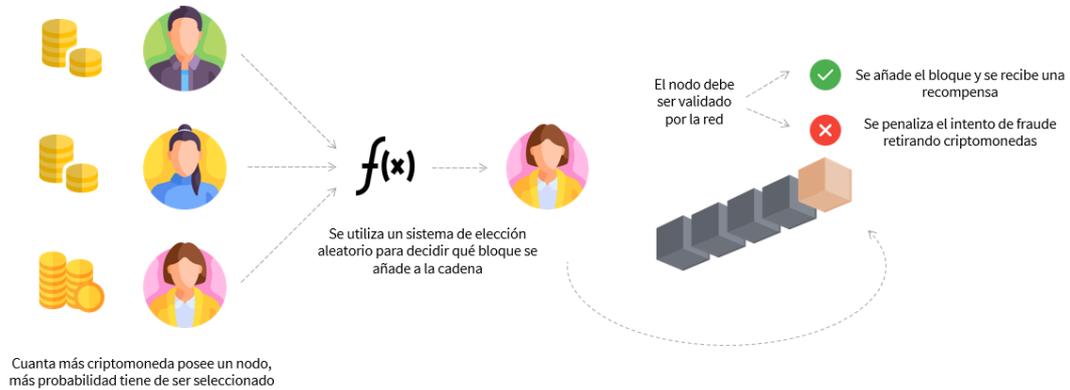


Figura 24. Algoritmo de consenso: Proof of Stake

3.1.4 CRIPTOGRAFÍA

Este elemento permite garantizar que los datos del libro mayor o los datos enviados entre nodos son seguros, auténticos y verificados. La criptografía ha desarrollado técnicas de cifrado indescifrables basándose en fundamentos matemáticos, en particular la teoría de la probabilidad, y en conocimientos de la teoría de juegos.

Existen diferentes tipos de algoritmos criptográficos, pero los principales son la criptografía simétrica y la criptografía asimétrica. [35]

3.1.4.1 Criptografía Simétrica

La criptografía simétrica es una forma de cifrado para proteger datos, en la que el cifrado y el descifrado de datos utilizan la misma clave. Dado que la clave se utiliza para descifrar los datos, debe mantenerse en secreto. Por lo tanto, cuando se utiliza una clave simétrica, el emisor y el receptor necesitan un mecanismo para intercambiar claves antes de intercambiar datos. [35]

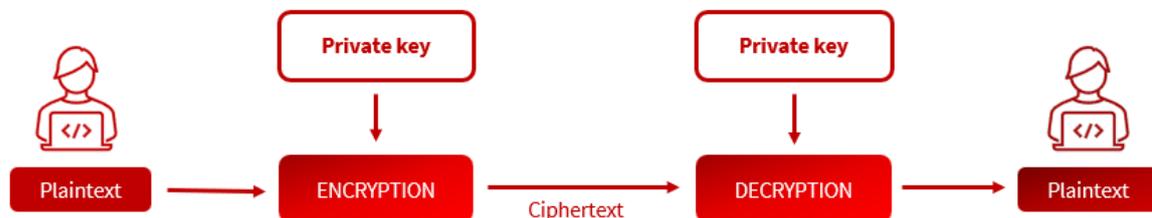


Figura 25. Criptografía Simétrica

Los algoritmos de cifrado simétricos más comunes incluyen DES (Data Encryption Standard), AES (Advanced Encryption Standard) y Blowfish. [35]

3.1.4.2 Criptografía Asimétrica

La criptografía asimétrica es una forma de cifrado para asegurar los datos, en la que el cifrado y descifrado de datos utiliza dos claves diferentes. La clave utilizada para cifrar los datos se denomina clave pública, que puede compartirse ampliamente y considerarse como la identidad de una persona (o como una dirección de Blockchain). La clave utilizada para descifrar los datos se denomina clave privada, necesaria por motivos de seguridad para proteger los derechos del receptor. [35]

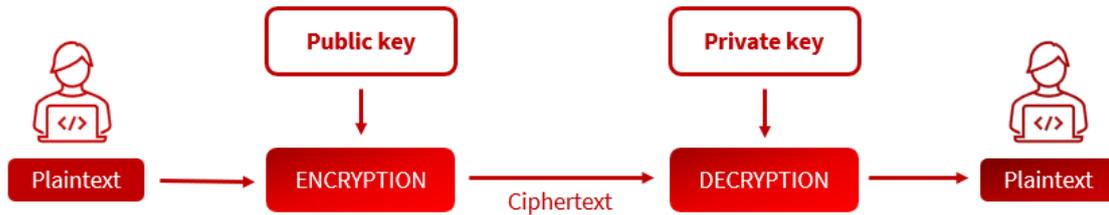


Figura 26. Criptografía Asimétrica

Los algoritmos de cifrado asimétricos más comunes incluyen RSA y ECC (Elliptic Curve Cryptography). [35]

3.1.4.3 Otras Técnicas Relevantes De Criptografía

3.1.4.3.1 Dirección Blockchain

Una dirección es una cadena única de caracteres que se utiliza para identificar una cuenta o billetera digital. Se comparte públicamente para que otros usuarios puedan enviar transacciones. Cada dirección de Blockchain se generará a partir de una clave pública, a través de un proceso de hashing que utiliza un algoritmo criptográfico para convertir la clave pública en una cadena de caracteres única y determinista. En el caso de Bitcoin, se utiliza el algoritmo SHA-256 para generar direcciones.

Dicha clave pública se genera a partir de una clave privada que sirve como mecanismo para demostrar la propiedad de la clave pública (o lo que es lo mismo, la dirección Blockchain). Al realizar una transacción interactiva con la red Blockchain, el usuario utilizará la clave privada para firmar digitalmente, demostrando que el usuario es el propietario de la dirección Blockchain válida en la transacción.

Las direcciones de blockchain suelen ser largas y complicadas, lo que dificulta la falsificación o el acceso no autorizado a una cuenta. Sin embargo, también pueden ser difíciles de recordar y escribir, por lo que muchos usuarios utilizan aplicaciones o servicios para gestionar sus direcciones y transacciones.

3.1.4.3.2 Firma Digital

La firma digital es una cadena de caracteres cifrada que se envía con los datos originales de la transacción en la plataforma Blockchain. Para crear una firma digital, el usuario utilizará una clave privada para cifrar (lo que se denomina firma digital) los datos contenidos en la transacción enviada al destinatario. Recuerde que la clave secreta utilizada para este cifrado es la clave secreta que genera la dirección Blockchain del remitente. La firma digital se genera vinculada a un documento específico, por lo que cambiará si cambian los datos de la transacción utilizados para el cifrado, o en el caso de los mismos datos, pero utilizando la clave privada de un usuario diferente. [35] [36]

3.1.4.3.3 Función Hash

Es el proceso de convertir una cantidad ilimitada de datos de entrada y crear una longitud fija de datos de salida. Las funciones hash se utilizan a menudo para proteger la integridad de los datos. Los usuarios pueden verificar la validez de una transacción comparando el valor hash de la transacción en la aplicación con el valor hash de la transacción en el explorador de bloques.

La función hash es una operación unidireccional, es decir, no se puede obtener la entrada original a partir de la clave hash. Además, cualquier cambio en los datos de entrada, por pequeño que sea, dará como resultado una clave hash completamente diferente.

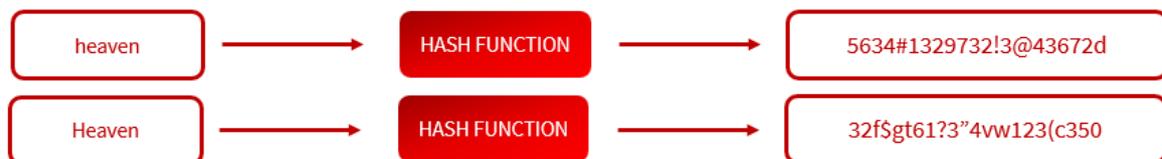


Figura 27. Diferencia Hash modificando entrada

En Blockchain se utilizan para garantizar la integridad y la seguridad de los bloques de transacciones. Cada bloque en una cadena de bloques contiene un valor hash que representa todas las transacciones del bloque. Si alguien intenta modificar una transacción dentro del bloque, el valor hash cambiará y se detectará que el bloque ha sido modificado. [35] [37]

3.1.5 MÁQUINA VIRTUAL

Un programa que simula un sistema informático se conoce como máquina virtual. Dispone de una CPU, memoria y almacenamiento virtual. Una máquina virtual funciona esencialmente como un ordenador físico; puede utilizarse para almacenar datos, ejecutar aplicaciones y colaborar con otras máquinas virtuales para ejecutar una red Blockchain.

3.1.5.1.1 Ethereum Virtual Machine (EVM)

La máquina virtual de Ethereum se utiliza para garantizar que las transacciones procesadas en entornos y configuraciones informáticas completamente diferentes generen siempre los mismos resultados en la plataforma Ethereum. Esencialmente, una EVM es una máquina que procesa contratos inteligentes que se ejecutan en Ethereum. [6]

Los nodos que participan en el sistema Ethereum procesan las transacciones recibidas a través de la EVM. Cualquier transacción que quiera cambiar el estado de la red debe pasar por el proceso de la EVM. La EVM no es más que una máquina virtual de la que se hacen muchas copias. Cada nodo que participa en la ejecución de las mismas transacciones posee una copia de la EVM para garantizar la coherencia del cómputo.

3.2 MACHINE LEARNING

Se conoce como Machine Learning a una disciplina del campo de la Inteligencia Artificial (IA) que, a través de algoritmos, dota a los ordenadores de la capacidad de identificar patrones en datos masivos y elaborar predicciones. Este aprendizaje permite a los computadores realizar tareas específicas de forma autónoma, es decir, sin necesidad de ser programados. [38]

3.2.1 PROCESO DE APRENDIZAJE AUTOMÁTICO

El proceso de aprendizaje automático que, generalmente, siguen los algoritmos de Machine Learning es el siguiente: [39]

3.2.1.1 Recopilación De Datos

Se recaban información y datos, posteriormente preparados para el problema al que se le busca solución. Para este tipo de algoritmos los datos pueden ser muy variados, desde texto, hasta audio o imágenes; o cualquier otra forma de información estructurada o no estructurada.

3.2.1.2 Preprocesamiento De Datos

Transformaciones realizadas a los datos para realizar limpieza y adaptarlos a nuestras necesidades. Entre estas transformaciones podemos encontrar: normalizaciones, tratamiento de valores faltantes, eliminación de ruido...

3.2.1.3 Selección De Modelo

Como se explicará más adelante, existen diferentes tipos de modelos, que se elegirán en función de lo adecuado para el problema en cuestión. Hay una amplia gama de modelos disponibles, desde modelos más clásicos como los de regresión lineal o árboles de decisión; y modelos con enfoques más novedosos, como redes neuronales, algoritmos de clasificación o algoritmos de clustering.

3.2.1.4 Entrenamiento Del Modelo

Una vez se han preprocesado los datos y se ha elegido un modelo, se separan un conjunto de datos, considerados de entrenamiento (o train). Durante el proceso de entrenamiento, el modelo “aprende” de los datos y ajusta sus características internas para mejorar su capacidad de realizar predicciones lo más precisas posibles. En otras palabras, identifica el patrón que siguen los datos para posteriormente aplicarlo con otros datos.

3.2.1.5 Evaluación Del Modelo

Tras entrenar el modelo, se evalúa su rendimiento utilizando otro conjunto de datos, el conjunto de train. Esto permitirá al desarrollador comprobar cómo se comporta el modelo ante datos no vistos previamente, y, si, por tanto, puede realizar predicciones con alto nivel de precisión.

3.2.1.6 Ajuste Y Optimización Del Modelo

A la vista de los resultados de la fase anterior, si el rendimiento del modelo no es satisfactorio, se pueden reajustar los parámetros del modelo o probar diferentes técnicas de preprocesamiento para tratar de mejorar los resultados obtenidos.

3.2.1.7 Despliegue Y Uso Del Modelo

Una vez el modelo ha sido entrenado y evaluado satisfactoriamente, se puede utilizar para hacer predicciones sobre nuevos datos.

3.2.2 CATEGORÍAS DE APRENDIZAJE AUTOMÁTICO

Por otro lado, como se menciona con anterioridad, los algoritmos de Machine Learning se dividen en varias categorías, siendo las dos primeras las más comunes. [40]

3.2.2.1 Aprendizaje Supervisado

En este tipo de algoritmos, se dispone de un conjunto de datos etiquetados; lo que quiere decir que se conocen las salidas correctas para cada variable/s de entrada. El algoritmo aprende a partir de esos datos para hacer predicciones o clasificaciones sobre nuevos datos. Algunos ejemplos son la regresión lineal simple o múltiple, árboles de decisión o redes neuronales.

3.2.2.2 APRENDIZAJE NO SUPERVISADO

En este caso, los algoritmos no cuentan con un conocimiento previo. Es decir, los datos no están etiquetados, y por tanto se enfrentan al caos de datos con el objetivo de encontrar patrones que permitan organizarlos de alguna manera. Entre los ejemplos, encontramos algoritmos de clustering como K-Means o DBScan.

3.2.2.3 Aprendizaje Por Refuerzo

Este tipo de algoritmo se basa en un proceso de toma de decisiones secuenciales, en el cual el algoritmo aprende a partir de la propia experiencia: en cada situación se debe tomar una decisión de acuerdo con un proceso de prueba y error en el que se recompensan las

decisiones correctas. Se utilizan en aplicaciones como juegos, robótica o control de sistemas; además de en reconocimiento facial.

En conclusión, estos son solo algunos ejemplos de los tipos de algoritmos utilizados en el aprendizaje automático. Cada uno tiene sus propias características, ventajas y desafíos, y la elección del algoritmo adecuado depende del problema específico que se esté abordando y de los datos disponibles.

3.2.3 ALGORITMOS DE REGRESIÓN LINEAL

A continuación, se explicarán en profundidad este tipo de algoritmos ya que el proyecto se utilizará un algoritmo de regresión lineal múltiple.

El algoritmo de regresión lineal se utiliza para modelar la relación entre una variable dependiente (la variable a predecir) y una o más variables independientes (las características o atributos). Es importante destacar que la regresión lineal asume una relación lineal entre las variables, por lo que no se considera adecuada para problemas con relaciones no lineales. [41]

Se clasifica en dos tipos en función del número de variables independientes.

3.2.3.1 Algoritmo De Regresión Lineal Simple

En su versión más sencilla, el algoritmo de regresión lineal simple consistirá en dibujar una recta que nos indicará la tendencia de un conjunto de datos continuos; se buscará minimizar la diferencia entre los valores observados y los valores predichos por el modelo. La recta se representa mediante la ecuación de la forma:

$$y = mx + b$$

Donde “y” es la variable que predecir, “x” es la variable independiente, “m” es la pendiente de la línea (representa la relación entre x e y) y “b” es el sesgo del algoritmo. El proceso de entrenamiento en este caso implica encontrar valores óptimos para los

coeficientes “m” y “b” que minimicen la una función de coste, que suele ser el error cuadrático medio (ECM). [42]

3.2.3.2 Algoritmo De Regresión Lineal Múltiple

Este segundo caso es una extensión del algoritmo de regresión lineal simple, que permitirá modelar la relación de una variable dependiente con múltiples variables independientes. En lugar de ajustar una recta, se ajustará un plano. La ecuación se representa de la siguiente manera:

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \dots + b_n * x_n$$

Donde “y” es la variable dependiente que se desea predecir, “x1, x2, ..., xn” son las variables independientes; y “b0, b1, ..., bn” son los coeficientes que representan la influencia de cada variable independiente en la variable dependiente.

Ahora, el proceso de entrenamiento del modelo de regresión lineal múltiple implica los valores más correctos para los coeficientes, de manera que se minimice de nuevo una función de costes, generalmente el error cuadrático medio (ECM). [42]

3.3 MANTENIMIENTO DE AERONAVES

El sistema de mantenimiento de aviones sigue un conjunto de estándares y regulaciones muy estrictos. Está regulado a nivel internacional por la Organización de Aviación Civil Internacional (OACI) [43] y a nivel nacional por las autoridades de aviación civil de cada país: en Europa la seguridad es responsabilidad de la Agencia Europea de Seguridad Aérea (EASA) [44], mientras que en Estados Unidos es de la Administración Federal de Aviación (FAA). [45]

Estas regulaciones establecen los requisitos y estándares mínimos que deben cumplir los operadores de aviones y los proveedores de mantenimiento para garantizar la seguridad de los vuelos. El objetivo de estos estándares es garantizar la seguridad de los pasajeros y la tripulación, así como maximizar la disponibilidad y confiabilidad de los aviones.

Las regulaciones de la OACI incluyen los Anexos al Convenio sobre Aviación Civil Internacional [46], que establecen los requisitos de seguridad para los aviones y los servicios de mantenimiento. Los Anexos relevantes para el mantenimiento de aviones son el Anexo 6 - Operación de Aeronaves y el Anexo 8 - Aeronavegabilidad. Estos Anexos establecen los requisitos para la certificación de las empresas de mantenimiento, la capacitación del personal de mantenimiento y los procedimientos de mantenimiento que deben seguirse.

En cuanto a la gestión de registros de mantenimiento, existe la norma ISO 9001 [47] que es una norma internacional que establece los requisitos para un sistema de gestión de calidad en una organización. En el contexto de la gestión de registros de mantenimiento, la norma ISO 9001 puede ayudar a garantizar la integridad y la exactitud de los registros de mantenimiento.

Asimismo, el mantenimiento de aviones se divide en diferentes categorías, dependiendo de la frecuencia y la complejidad de las tareas de mantenimiento requeridas. Estas categorías incluyen el mantenimiento programado, el mantenimiento no programado y el mantenimiento de líneas de base.

A continuación, se explicará en primer lugar la norma ISO 9001, pues es relevante conocerla para saber en qué consiste el registro actual de revisiones de mantenimiento. Y, posteriormente, se explicará la clasificación de las categorías en las que se divide el mantenimiento de aviones.

3.3.1 NORMA ISO 9001

La norma ISO 9001 es una norma internacional que establece los requisitos para un sistema de gestión de calidad en una organización. La norma se enfoca en la satisfacción del cliente y en la mejora continua de los procesos de la organización. Se divide en 10 secciones principales: [47]

- Objeto y campo de aplicación: Esta sección establece el propósito de la norma y a quién se aplica.

- Referencias normativas: Esta sección enumera las normas y regulaciones que deben ser cumplidas por la organización.
- Términos y definiciones: Esta sección establece los términos y definiciones que se utilizan en la norma.
- Contexto de la organización: Esta sección se enfoca en la comprensión de la organización y su contexto.
- Liderazgo: Esta sección establece los requisitos para el liderazgo y el compromiso de la alta dirección.
- Planificación: Esta sección establece los requisitos para la planificación del sistema de gestión de calidad.
- Apoyo: Esta sección establece los requisitos para los recursos necesarios para el sistema de gestión de calidad.
- Operación: Esta sección establece los requisitos para la ejecución del sistema de gestión de calidad.
- Evaluación del desempeño: Esta sección establece los requisitos para la medición y evaluación del desempeño del sistema de gestión de calidad.
- Mejora: Esta sección establece los requisitos para la mejora continua del sistema de gestión de calidad.

Su aplicación en el mantenimiento de aviones es crucial. A continuación, se describen algunas de las áreas específicas en las que se aplica esta norma.

3.3.1.1 Gestión De La Documentación

La norma ISO 9001 requiere que las organizaciones de mantenimiento de aviones tengan un sistema de gestión documental claro y eficaz. Esto incluye la definición de los documentos y registros necesarios para el mantenimiento de los aviones, así como su revisión, aprobación y distribución.

Dichos documentos y registros necesarios para el mantenimiento de aviones pueden variar según el tipo de avión, su fabricante, la regulación aplicable y otros factores específicos de

cada caso. Sin embargo, a continuación, se presentan algunos ejemplos de los documentos y registros comúnmente requeridos. [48]

3.3.1.1.1 Manual De Mantenimiento

Este es un documento que describe los procedimientos y estándares de mantenimiento recomendados por el fabricante del avión. El manual de mantenimiento debe ser revisado y actualizado regularmente para garantizar que refleje las últimas especificaciones del fabricante.

3.3.1.1.2 Hojas De Trabajo De Mantenimiento

Estas son documentos que describen las tareas de mantenimiento específicas que se deben realizar en cada avión, así como los procedimientos, herramientas y equipos necesarios para llevar a cabo estas tareas. Las hojas de trabajo de mantenimiento deben ser detalladas y estar actualizadas para garantizar que se sigan los procedimientos de mantenimiento adecuados.

3.3.1.1.3 Registros De Mantenimiento

Estos son documentos que registran las actividades de mantenimiento realizadas en cada avión, incluyendo las fechas de las revisiones, las reparaciones y las sustituciones de componentes o piezas. Los registros de mantenimiento deben ser precisos y completos para garantizar que se puedan rastrear todas las actividades de mantenimiento realizadas en el avión.

3.3.1.1.4 Certificados De Aeronavegabilidad

Estos son documentos oficiales emitidos por la autoridad reguladora de aviación del país que certifican que un avión cumple con los estándares de seguridad requeridos para su operación. Los certificados de aeronavegabilidad deben estar actualizados y disponibles en todo momento para garantizar que el avión pueda operar legalmente. [49] [50]

3.3.1.1.5 Informes De Inspección

Estos son documentos que registran los resultados de las inspecciones realizadas en cada avión, incluyendo cualquier defecto o problema identificado. Los informes de inspección deben ser detallados y precisos para garantizar que se puedan tomar medidas adecuadas para corregir cualquier problema identificado.

3.3.1.1.6 Planes De Mantenimiento

Estos son documentos que describen los planes de mantenimiento programado para cada avión, incluyendo los intervalos de tiempo y las tareas específicas que se deben realizar en cada revisión. Los planes de mantenimiento deben estar actualizados y disponibles en todo momento para garantizar que se sigan los procedimientos de mantenimiento adecuados.

3.3.1.2 Identificación Y Trazabilidad

La norma ISO 9001 exige que las organizaciones de mantenimiento de aviones identifiquen claramente todos los componentes y piezas utilizados en el mantenimiento de los aviones. Esto incluye la identificación de los fabricantes, los números de serie y cualquier otra información relevante. También se requiere que las organizaciones tengan un sistema de trazabilidad que permita rastrear el historial de mantenimiento de cada componente o pieza. [51]

3.3.1.3 Gestión De Riesgos

La norma ISO 9001 exige que las organizaciones de mantenimiento de aviones identifiquen y gestionen los riesgos asociados con sus procesos de mantenimiento. Esto incluye la identificación de los riesgos potenciales, la evaluación de su probabilidad y su impacto, y la implementación de medidas para minimizar o controlar estos riesgos.

3.3.1.4 Formación Y Competencia

La norma ISO 9001 exige que las organizaciones de mantenimiento de aviones tengan un plan de formación y desarrollo del personal para garantizar que estén debidamente capacitados y competentes para realizar sus tareas de mantenimiento. Esto incluye la

definición de los requisitos de formación, la evaluación de la competencia y la documentación de la formación y la competencia del personal.

3.3.1.5 Evaluación Del Desempeño

La norma ISO 9001 exige que las organizaciones de mantenimiento de aviones realicen una evaluación regular de su desempeño en relación con los objetivos establecidos. Esto incluye la realización de auditorías internas y la revisión de los resultados de las auditorías para identificar áreas de mejora y tomar medidas para mejorar continuamente el desempeño.

En resumen, la norma ISO 9001 se aplica al mantenimiento de aviones en varias áreas clave, incluyendo la gestión de la documentación, la identificación y trazabilidad de componentes y piezas, la gestión de riesgos, la formación y competencia del personal, y la evaluación del desempeño. Estas áreas son críticas para garantizar la seguridad, la calidad y la eficiencia en los servicios de mantenimiento aeronáutico. [52]

3.3.2 CATEGORIAS DE MANTENIMIENTO DE AVIONES

3.3.2.1 Mantenimiento Programado O Preventivo

Este tipo de mantenimiento se realiza según un calendario preestablecido y/o en función del número de horas de vuelo, ciclos de despegue/aterrizaje y/o tiempo transcurrido desde la última inspección. El objetivo del mantenimiento programado es detectar y corregir los problemas antes de que se conviertan en fallas críticas que puedan afectar la seguridad o la confiabilidad de la aeronave. [53]

El programa de mantenimiento programado se basa en las especificaciones y recomendaciones del fabricante de la aeronave, y también puede ser influenciado por los requisitos de las autoridades reguladoras de la aviación, como la Administración Federal de Aviación (FAA) en los Estados Unidos o la Agencia Europea de Seguridad Aérea (EASA) en Europa.

El programa de mantenimiento programado puede incluir tareas como:

- Inspecciones visuales y funcionales de los sistemas y componentes de la aeronave.
- Reemplazo de piezas y componentes según el tiempo de vida útil o la cantidad de horas de vuelo/ciclos de despegue-aterrizaje.
- Reparación de daños y desgaste en las estructuras y componentes de la aeronave.
- Lubricación y ajuste de los sistemas de la aeronave.
- Pruebas de sistemas y componentes, como los sistemas de frenos, los sistemas hidráulicos, los sistemas eléctricos y otros sistemas.

3.3.2.2 Mantenimiento No Programado O Correctivo

El mantenimiento no programado, también conocido como mantenimiento correctivo, se refiere a la reparación o reemplazo de componentes o sistemas de una aeronave que se han dañado o fallado inesperadamente, fuera del programa de mantenimiento regular. [53]

El mantenimiento no programado puede ser causado por una variedad de factores, como el desgaste normal de los componentes, la exposición a condiciones ambientales extremas, el mal uso o la falta de mantenimiento adecuado, o incluso daños causados por accidentes.

Las tareas de mantenimiento no programado son generalmente más urgentes y críticas que las tareas de mantenimiento programado, ya que se realizan para reparar o reemplazar un componente o sistema que ya ha fallado o está a punto de fallar, y que puede afectar la seguridad o la confiabilidad de la aeronave.

Las tareas de mantenimiento no programado pueden incluir:

- Reemplazo de componentes que han fallado, como un motor, una bomba, un generador, etc.
- Reparación de daños en la estructura de la aeronave causados por accidentes o incidentes, como golpes de pájaros o golpes de granizo.
- Resolución de problemas eléctricos o electrónicos que pueden afectar el funcionamiento de la aeronave.
- Solución de problemas mecánicos, como problemas en los sistemas de frenos o en los sistemas hidráulicos.

- Reparación de daños en el sistema de combustible, el sistema de aire acondicionado, el sistema de presurización, entre otros.

3.3.2.3 Mantenimiento Especial

Este tipo de mantenimiento se realiza en situaciones específicas, como, por ejemplo, después de un accidente o incidente, o cuando se requiere la instalación o modificación de un equipo o sistema nuevo. El mantenimiento especial puede requerir herramientas, habilidades y procedimientos especiales para garantizar que se realice de manera efectiva y segura.

El mantenimiento especial puede incluir tareas como:

- Inspección y reparación de componentes y sistemas que no están cubiertos en el programa de mantenimiento regular, como los sistemas de entretenimiento a bordo, los sistemas de iluminación, etc.
- Modificaciones o actualizaciones de la aeronave para cumplir con los nuevos requisitos regulatorios, mejorar la eficiencia, la seguridad o la comodidad de los pasajeros.
- Inspección y reparación de daños causados por accidentes o incidentes graves, como choques con aves, accidentes de tierra, etc.
- Inspección y reparación de daños en la estructura de la aeronave causados por la fatiga del material o por otros factores.
- Inspección y reparación de daños en componentes críticos que pueden afectar la seguridad o la confiabilidad de la aeronave, como el tren de aterrizaje, los sistemas de combustible, los sistemas hidráulicos, etc.

3.3.3 DIGITALIZACIÓN EN MANTENIMIENTO DE AVIONES

3.3.3.1 Iberia Mantenimiento

Desde el año 2020, la compañía Iberia, perteneciente al gigante de la aviación IAG apuesta por la digitalización de sus sistemas de mantenimiento. Eligió a la empresa Ramco,

proveedor de software especializado en la aviación para la gestión de este tipo de recursos.
[54]

El nuevo software de esta empresa permite a Iberia Mantenimiento unificar toda el área de revisiones, taller, componentes y cadena de suministros en un único sistema. Desde su implantación, ha mejorado la integración de toda la información, puesto que todo se almacena en un único software; así como ha mejorado la sostenibilidad de la empresa ya que no se tendrán que imprimir miles de documentos.

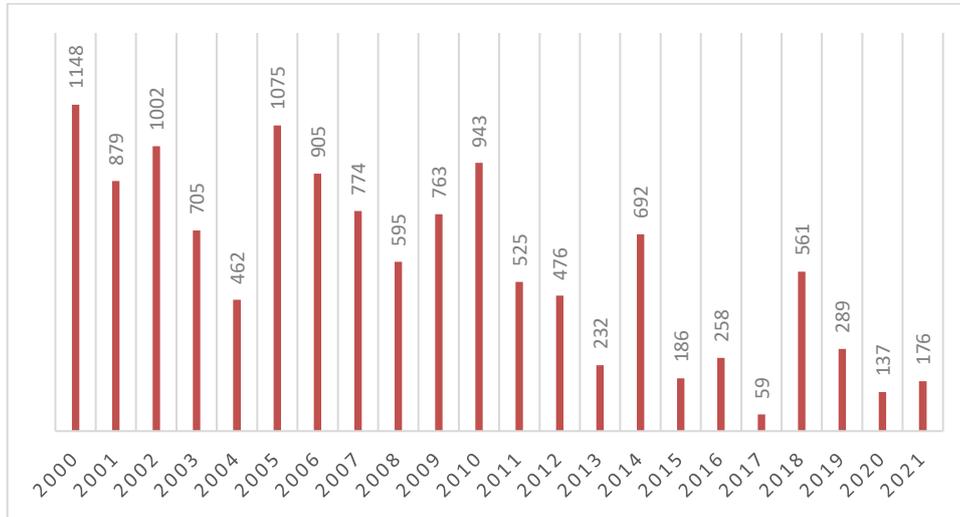
En cambio, quedan ámbitos de mejora en la implantación, como es la seguridad. El hecho de que toda la información se almacene en un único lugar, la base de datos del sistema hace que la información sea vulnerable y quede expuesta ante un único foco de ataque.

3.4 BLOCKCHAIN EN EL MUNDO DE LA AVIACIÓN

El uso de blockchain en el mundo de la aviación se encuentra en una etapa temprana de desarrollo y adopción, pero se espera que tenga un gran potencial para mejorar la eficiencia, la transparencia y la seguridad en varios aspectos de la industria.

3.5 ACCIDENTES DE AVIACIÓN

Como se ha mencionado en apartados previos, en el ámbito de la aviación, la seguridad y el mantenimiento son asuntos de vital importancia. En cambio, a lo largo de la historia se han descuidado en múltiples ocasiones, causando catástrofes mundiales que han supuesto la pérdida de múltiples vidas.



Desde el año 2000 hasta el año 2021, se han registrado más de 12800 fallecidos en accidentes de aviación. Afortunadamente, la tendencia es favorable, pues la cifra del año 2021 corresponde a 176 fallecidos, que supone una reducción del 84% respecto a la cifra del año 2000, 1148 fatalidades. En cambio, la situación debe cambiar y avanzar hasta llegar a un total de 0 víctimas mortales en accidentes. [55] [56]

Entre las causas principales de accidentes de aviación se encuentran los fallos mecánicos, defectos de mantenimiento, defectos de inspección, no reemplazar partes a tiempo, o fallos de diseño.

Los fallos mecánicos se postulan como la segunda causa de accidentes, suponiendo alrededor del 15% de los accidentes cada año. [57]

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

A la vista del Estado actual de la cuestión, se pueden extraer diferentes áreas de mejora en la gestión y verificación de revisiones, así como al almacenamiento de toda la información relativa. Entre ellos, se encuentra la seguridad a la hora de verificar y certificar un informe de revisión; de la posible modificación de los informes sin dejar rastro; y la pérdida en tiempo y dinero que supone elegir la fecha de la próxima revisión de manera manual.

4.1.1 SEGURIDAD EN VERIFICACIÓN Y CERTIFICACIÓN

En la actualidad, una vez que la revisión se completa confirme a los requisitos establecidos, se emite un certificado o informe de inspección que documenta los resultados obtenidos por parte de la persona que realiza la inspección. Dicho certificado se revisa y verifica por parte de personal autorizado, como inspectores de calidad o supervisores previo a almacenarla. Por tanto, la información viaja por diferentes manos sin asegurar su veracidad hasta el último momento.

Con la utilización de la tecnología Blockchain, el mecánico directamente guarda la información, quedando también un registro en la cadena de bloques, que proporciona, por un lado, la inmutabilidad en la cadena de bloques. Esto significa que una vez se registra una revisión realizada, no puede ser modificada ni alterada, lo que brinda una mayor confianza en la integridad de los datos obtenidos. Así mismo, proporciona transparencia y trazabilidad, ya que todos los integrantes de la red Blockchain pueden verificar los certificados o informes registrados; y reduce el número de intermediarios involucrados en tratar una información tan sensible.

4.1.2 MODIFICACIÓN DE INFORMES

La utilidad de la tecnología Blockchain radica en su capacidad para proporcionar un alto nivel de seguridad, con una estructura confiable donde la información introducida no puede ser modificada sin el consenso de la red. Esta cualidad resulta de vital importancia a la hora del registro de revisiones, puesto que proporciona credibilidad a los datos de la revisión. Hasta el momento, la información, certificada de manera insegura, como se ha explicado previamente, se almacena en bases de datos y carpetas inseguras, dando pie a un atacante o persona malintencionada a realizar cambios en la información, pudiendo generar una catástrofe.

Por tanto, el hecho de almacenar la información en Blockchain evitará una posible modificación del contenido de los informes, reduciendo así la superficie de ataque del sistema de revisiones de una determinada empresa.

4.1.3 ELECCIÓN FECHA REVISIÓN

Tradicionalmente, en todas las empresas de aviación se encontraba una persona encargada de fechar la próxima revisión de una aeronave, en base a unos criterios sobre el papel, fechados de años atrás, que no tenían en cuenta el estado de la flota de la empresa en cada momento.

Esto ocasiona una crisis de seguridad, puesto que según avanzan las aeronaves en horas de uso y en problemas anteriores, su confiabilidad va reduciendo por plazos, necesitando revisiones cada vez más constantes. El algoritmo desarrollado tiene en cuenta la información sobre el estado actual de la aeronave y su historial de vuelos para determinar cómo de próxima o lejana tiene que ser su siguiente revisión del sistema.

4.2 OBJETIVOS

Como se ha comentado a lo largo de la memoria, el objetivo de este proyecto ha sido la creación de una página web tipo Intranet, para mejorar el sistema digital del mantenimiento

de aviones portando la tecnología Blockchain y el Machine Learning consigo. Para conseguirlo, se han marcado los siguientes objetivos principales:

- Desarrollo, implementación y posterior despliegue de un contrato inteligente que permita a los usuarios crear un hash de un archivo PDF para que quede almacenado en la cadena de bloques, así como extraer dicho hash. En ambos casos se usan mappings y se registran los hashes junto con su número de informe.
- Desarrollo de un algoritmo de Machine Learning: Algoritmo de Regresión Lineal Múltiple que prediga en qué momento es necesario realizar una revisión de mantenimiento, basándose en el número de elementos a revisar, y las horas de vuelo que lleva un avión a sus espaldas.
- Implementación de una página web tipo Intranet, segura y accesible desde donde se puedan realizar todas las funcionalidades mencionadas a lo largo del informe.
- Implementación de un perfil de administrador para gestionar las cuentas del resto de usuarios (mecánicos).

Para lograr estos objetivos principales, se marcaron también ciertos objetivos intermedios:

- Creación de una base de datos con múltiples tablas para almacenar toda la información acerca de los aviones y las revisiones, de manera que sólo se guarde en Blockchain la mínima información posible, de manera que se consuman muchos menos recursos en la cadena de bloques, y, consecuentemente, tiempo y dinero.
- Creación de un programa que automatice la creación y guardado de informes, de manera que, posteriormente, esté disponible para su almacenamiento en Blockchain.
- Realización de la conexión entre el proyecto SpringBoot con Java para el manejo de la base de datos, y con Python para obtener resultados del Algoritmo de Regresión Lineal Múltiple.

4.3 METODOLOGÍA

El proyecto se ha realizado utilizando diferentes lenguajes de programación: para toda la parte de back-end se ha utilizado Java (conexiones con BBDD, etc.) y SQL (gestión de BBDD en sí); para la parte de front-end y todo lo relacionado con la apariencia hacia el usuario se ha utilizado JavaScript (con HTML y CSS); para el desarrollo del Smart Contract se ha elegido Solidity como lenguaje a utilizar; y por último, se ha utilizado Python para el desarrollo de todo lo referente a Machine Learning.

Con el objetivo de ir cumpliendo objetivos poco a poco, se siguió la metodología AGILE SCRUM.

La metodología AGILE es un enfoque para la gestión y el desarrollo de proyectos que permite adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez. En lugar de seguir un enfoque lineal y secuencial, Agile se centra en la entrega continua de valor al cliente y fomenta la participación de los interesados en todas las etapas del proyecto. [58]

Y más concretamente, el método SCRUM consiste en ejecutar el proyecto en ciclos temporales cortos (en inglés, sprints), para los cuales se asignan diferentes tareas de corta duración, que permiten ir creando entregas parciales y regulares del producto final. Se han obviado las reuniones diarias (en inglés, daily meetings), ya que el trabajo se ha realizado de forma individual con ayuda de un director, pero no se ha contado con un equipo amplio al que coordinar. [59]

4.4 PLANIFICACIÓN

Para cumplir con los objetivos establecidos, se ha seguido un planning. En primer lugar, se dedicó un tiempo al estudio de las tecnologías necesarias, puesto que algunas como Blockchain o Solidity no se han estudiado durante el grado. Posteriormente, se dividió el trabajo a realizar.

Todos los objetivos del proyecto se han dividido en tres fases principales. La primera tarea realizada fue la referente a la puesta en marcha de todas las tecnologías: conexión con la base de datos, con el servidor; y creación de las diferentes tablas de la base de datos.

A continuación, se elaboró el Smart Contract y todo lo referente a Blockchain, levantando así la conexión con la cadena de bloques y con MetaMask para la gestión de las cuentas.

En tercer lugar, se elaboró y entrenó el Algoritmo de Regresión Lineal Múltiple, efectuando también la conexión de Python con el proyecto.

Y, por último, se dio unión y estructura a todo el proyecto, creando la página web, mejorando todas las conexiones, así como la apariencia final del proyecto; y asignando las tareas a ambos perfiles: mecánico y administrador.

De manera simultánea se ha llevado a cabo también la redacción del trabajo, donde se incluye todo el trabajo desarrollado y los análisis pertinentes. La programación seguida se describe en la siguiente tabla:

Meses	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May
Aprendizaje tecnologías necesarias (Blockchain, Solidity...)								
Creación BBDD para información y conexiones del proyecto								
Creación Smart Contract – Blockchain								
Integración Blockchain y MetaMask al proyecto								
Aplicar Machine Learning para predecir cuándo será necesario mantenimiento								
Diseño de la página web + Revisión de errores								
Documentación								

Tabla 1. Diagrama de Gantt - Cronograma del proyecto

4.5 ESTIMACIÓN ECONÓMICA

A la hora de realizar un proyecto para comercializar, es vital realizar una estimación económica para estudiar su viabilidad. Para ello, se ha realizado un análisis de los costes asociados al proyecto referentes tanto al software, como al material utilizado y al capital humano.

4.5.1 SOFTWARE

Para llevar a cabo el proyecto se han utilizado programas gratuitos y de código abierto, como son el IDE Visual Studio Code para el desarrollo completo; Jupyter Notebooks para realizar pruebas de todo lo realizado con Python; PgAdmin para gestionar de una manera sencilla la base de datos; y Ganache y MetaMask para Blockchain.

En cambio, el despliegue del Smart Contract si supone un coste, que se paga una única vez, al iniciar el funcionamiento del programa. Depende de la velocidad que se desee en la ejecución del contrato inteligente en cuestión, se pagará una cantidad diferente, tal y como se ve en la figura:



Figura 28. Precio del gas

Por comodidad, se utilizará el precio a velocidad estándar, el mostrado en la parte superior de la imagen. En el momento de escritura de este informe, se necesitan 27 gwei para

realizar la transacción a una velocidad estándar, y el precio de 1 ETH es de \$1877'72, que en euros equivale a 1714,66€. Por tanto, se calcularán cuantos ETH son necesarios para realizar el despliegue del contrato.

Dado que el despliegue del contrato requiere 473342 unidades, y 1 Ether equivale a 1.000.000.000 Gwei, la tarifa o valor aproximado del gas será:

$$ETH \text{ del despliegue} = \text{Número de gwei} * \text{Unidades de gas}$$

$$ETH \text{ del despliegue} = 27 * 473342 = 12780234 \text{ Gwei} = 0,012780234 \text{ ETH}$$

Lo que, en dinero, se traduce a:

$$\text{Precio final} = \text{Número de ETH} * \text{Precio 1 ETH}$$

$$\text{Precio final} = 0,012780234 \text{ ETH} * \frac{1877'72€}{\text{ETH}} = 23,997 €$$



Figura 29. Precio cambio USD - EUR

Por otro lado, el Sistema Operativo también tiene un costo. Sin embargo, viene incluido con la compra del ordenador, por lo que su precio también está incluido en el del ordenador, que se verá a continuación.

4.5.2 RECURSOS MATERIALES

En lo que a recursos materiales se refiere, se han utilizado un ordenador portátil, un monitor y un iPad para actividades puntuales. Sus costes se detallan en los próximos apartados.

4.5.2.1 Ordenador Portatil

El ordenador utilizado es el HP Laptop 14s-dq1021ns; un equipo de 14", con una CPU Intel Core de décima generación, modelo i7; y con memoria RAM de 8GB. Su precio aproximado fue de 800€. Dado que la vida útil de un ordenador de dichas características es de 4 años, y el uso de este para la realización del TFG ha sido de octubre a Junio (9 meses), el coste total, realizando los cálculos pertinentes es de 150€. [60]

4.5.2.2 Monitor

El monitor utilizado es el HP 27f Display; una pantalla auxiliar de 27" con IPS con retroiluminación LED, antirreflejo para facilitar el trabajo de programación. Su precio fue de 229€, y, dado que su vida útil, como la del ordenador, es de 4 años y el uso también ha sido de 9 meses, el coste total ha sido de 43€. [61]

4.5.2.3 IPad

El iPad utilizado es el iPad Air 5ª generación 256GB, junto con el Apple Pencil 2ª generación. El precio del conjunto fue de 860€. Dado que su vida útil es de 6 años, y también se ha utilizado durante 9 meses, el coste total es de 107.5€. [62]

4.5.3 CAPITAL HUMANO

Para la realización de este proyecto únicamente se ha necesitado un desarrollador software y un director que supervise el proyecto.

El salario medio de un programador junior en España es de 14.62€/hora. [63] Teniendo en cuenta que el trabajo ha supuesto alrededor de 400 horas de trabajo, el precio de este ha sido de un total de 5850€.

El salario del director, un programador más experimentado, es de 25.05€/hora. [63] Su trabajo de revisión, supervisión y ayuda con dudas ha supuesto alrededor de 50 horas, por lo que el precio asciende a 1250€.

Por tanto, el capital humano del proyecto ha supuesto un precio de 7100€ en total.

Tras desglosar el coste de las diferentes partes del proyecto, en la siguiente tabla se incluye un resumen de todo lo explicado anteriormente:

CONCEPTO	PRECIO
Despliegue Smart Contract	23,997€ ≈ 24€
Ordenador Portátil	150 €
Monitor	43 €
IPad + Apple Pencil	107.5€ ≈ 108€
Desarrollador Junior (400 horas)	5,850 €
Desarrollador Senior – Director (50 horas)	1,250 €
TOTAL	7,425 €

Tabla 2. Estimación Económica Proyecto

4.5.4 BENEFICIOS

Como todo proyecto, para estudiar su viabilidad es necesario analizar los resultados obtenidos en un corto, medio y largo plazo desde tres perspectivas o escenarios diferentes (pesimista, realista y óptimo). Se analizarán los beneficios que se obtendrán por realizar las revisiones correctamente y a tiempo. [64]

4.5.4.1 Escenario Óptimo

Habitualmente, las revisiones que se llevan a cabo por elementos que han fallado suelen acarrear un coste económico mayor, puesto que provocan una parada del avión no prevista, modifican los esquemas de los mecánicos que no planeaban tener que dedicar horas a esa avería; y suelen suponer un gasto mayor porque a menudo precisan de comprar recambios. En el escenario óptimo se contempla que no existan este tipo de fallos imprevistos hasta pasados 5 años del uso del avión, y, que, por tanto, los costes de mantenimiento sean relativamente bajos. En la tabla se muestran los precios de mantenimiento de los diferentes elementos del avión, y cuáles se aplican en este caso.

Elemento mantenimiento	Precio	1 año - app	1 año - trad	5 años - app	5 años - trad
Mantenimiento rutinario	100000	6	4	8	12
Fuselaje	20000	0	2	1	4

Ala izquierda	50000	0	1	1	3
Ala derecha	50000	0	1	1	3
Cola	20000	0	1	0	1
Motor	50000	0	2	1	4
Tren aterrizaje	50000	0	1	1	4
Cabina pasajeros	2000	0	1	0	1
Cabina trip técnica	2000	0	1	0	1
TOTAL		600000	714000	1020000	2004000

Tabla 3. Costes escenario Óptimo.

Gracias al uso de la aplicación, el ahorro económico, y, por tanto, beneficios para la empresa serían de 114000€ a corto plazo, únicamente durante el primer año. Posteriormente, estos beneficios irán creciendo de manera exponencial, puesto que cada vez se evitará un mayor número de revisiones imprevistas, con el único coste de aumentar el número de revisiones programadas; llegando a obtener cerca de 1 millón de euros de beneficio por avión cada año.

Por tanto, realizando la operación en el momento actual de la empresa con una flota de 83 aviones, los beneficios en el año 1 serán de 9462000€, mientras que, pasados 5 años, serán de 81672000€.

4.5.4.2 Escenario Realista

El escenario expuesto en el subapartado previo, dado el historial de las aeronaves a lo largo de los años, se puede considerar irreal, puesto que es altamente probable que existan fallos de manera imprevista. Un escenario realista sería el mostrado a continuación:

Elemento mantenimiento	Precio	1 año - app	1 año - trad	5 años - app	5 años - trad
Mantenimiento rutinario	100000	6	4	8	12
Fuselaje	20000	1	3	3	5
Ala izquierda	50000	1	2	3	4
Ala derecha	50000	1	2	3	4
Cola	20000	0	2	2	3
Motor	50000	1	2	3	4
Tren aterrizaje	50000	1	2	2	4

Cabina pasajeros	2000	0	2	1	2
Cabina trip técnica	2000	1	2	2	3
TOTAL		822000	908000	1456000	2170000

Tabla 4. Costes Escenario Realista

En este caso, ya en el primer año ocurren fallos, pese a las revisiones de mantenimiento preventivo realizadas. Aun así, es evidente que se produce un ahorro económico, y, por tanto, beneficio para la empresa. En el primer año, el beneficio será de 86000 euros. Según avanza el tiempo, el beneficio, será inferior que en el caso óptimo, pero igualmente crecerá de manera exponencial, llegando a más de 700000 euros por avión.

Por tanto, realizando la operación en el momento actual de la empresa con una flota de 83 aviones, los beneficios en el año 1 serán de 7138000€, mientras que, pasados 5 años, serán de 59262000€.

4.5.4.3 Escenario Pesimista

Por último, es importante analizar los resultados en el peor caso posible puesto que, siendo improbable que suceda, siempre estará la posibilidad de este y el sistema no se debe poner en uso si en este caso se producen pérdidas.

Elemento mantenimiento	Precio	1 año - app	1 año - trad	5 años - app	5 años - trad
Mantenimiento rutinario	100000	8	6	12	14
Fuselaje	20000	3	4	5	7
Ala izquierda	50000	2	3	5	6
Ala derecha	50000	2	3	5	6
Cola	20000	2	3	4	5
Motor	50000	3	4	5	5
Tren aterrizaje	50000	3	4	4	6
Cabina pasajeros	2000	2	4	2	4
Cabina trip técnica	2000	2	4	4	5
TOTAL		1408000	1456000	2342000	2808000

Tabla 5. Costes Escenario Pesimista

Por inspección visual de la tabla, se infiere que siguen produciéndose beneficios por el uso de la aplicación, aunque menores que en los anteriores escenarios. En el primer año, el

beneficio por cada avión es de 48000€, mientras que, pasados los 5 años, el beneficio se incrementa, ascendiendo hasta los 466000€.

Por tanto, realizando la operación en el momento actual de la empresa con una flota de 83 aviones, los beneficios en el año 1 serán de 3984000€, mientras que, pasados 5 años, serán de 38678000€.

A la vista de todos estos resultados, se puede confirmar que la puesta en marcha del proyecto es viable económicamente y que se obtendrá un alto beneficio económico gracias a él.

Capítulo 5. SISTEMA/MODELO DESARROLLADO

A lo largo de este capítulo se explicará el sistema desarrollado en el proyecto, dando a conocer su funcionamiento. El sistema consiste en una web tipo Intranet, dentro de la cual se han desarrollado funcionalidades que permitirán al departamento de mantenimiento de una empresa aeronáutica funcionar con normalidad. Para ello, se ha recurrido a la tecnología Blockchain; a la novedosa Inteligencia Artificial, haciendo hincapié en el campo de Machine Learning; y a la gestión de bases de datos.

Para explicar de manera adecuada el sistema que se ha creado a lo largo del Trabajo de Fin de Grado, se dividirá el capítulo en tres partes: una primera referente al análisis del sistema, donde se analizarán los requisitos del sistema y la funcionalidad a desarrollar; una segunda parte donde se explicará el diseño de la arquitectura que se ha llevado a cabo, estructura de la información, etc.; y un tercer punto que hará referencia a la implementación de los cuatro bloques fundamentales que se han desarrollado en el proyecto: Blockchain, Base de Datos, Machine Learning y Página Web.

5.1 ANÁLISIS DEL SISTEMA

5.1.1 ANÁLISIS DE REQUISITOS

Como se ha explicado en apartados anteriores, el objetivo fundamental del proyecto es el desarrollo de una página web tipo Intranet que permita a los mecánicos encargados del mantenimiento de aviones registrar las revisiones que realizan a los aviones y almacenarlas de manera segura utilizando la tecnología Blockchain.

Para ser más exhaustivos, se va a proceder a la realización de un análisis de los requisitos que se utilizan en el desarrollo software para describir qué debe hacer un sistema y cómo debe hacerlo. Se dividen en dos categorías: funcionales y no funcionales. Los requisitos funcionales hacen referencia a las tareas, operaciones o servicios que el sistema creado

debe ser capaz de ejecutar; centrándose en acciones y comportamientos del sistema. Por otro lado, los requisitos no funcionales hacen referencia a atributos o características que el sistema debe poseer, además de sus funcionalidades; principalmente en términos de seguridad, rendimiento y usabilidad; centrándose en las cualidades y restricciones del sistema, más que en las acciones. [65]

5.1.1.1 Requisitos Funcionales

En este apartado se enumeran los requisitos funcionales que el sistema desarrollado debe cumplir:

- El mecánico responsable de realizar una revisión debe poder registrar una revisión realizada, accediendo al apartado de “Nueva Revisión” de la web.
- Como segunda opción, el mecánico también debe poder registrar dicha revisión realizada desde el apartado de “Alertas”, que le debe redirigir a la pestaña de “Nueva Revisión”, mencionada en el punto anterior.
- El sistema debe ser capaz de generar el informe PDF de cada revisión introducida por el mecánico de manera automática y guardarla con el nombre informeXX.pdf (siendo XX el número de revisión realizada).
- El sistema debe guardar también de manera automática un mapping en Blockchain con el número de revisión y el hash del informe PDF previamente generado como map.
- En línea con el requisito anterior, también debe ser posible extraer la información guardada en Blockchain en dichos mapping aportando únicamente el número de revisión.
- El sistema debe contar con una pestaña “Comprobar Informes”, dedicada a comprobar si los informes almacenados en PDF son fieles a la realidad, comprobando que su hash coincida con el hash almacenado en Blockchain del informe correspondiente a esa revisión.
- La aplicación debe permitir el acceso a la Intranet únicamente a aquellos usuarios correctamente identificados: con DNI y cuenta de Blockchain válidos.

- La aplicación debe permitir registrar nuevos usuarios (debidamente autenticados), en caso de que la empresa crezca.
- El sistema de Inicio de Sesión y registro debe encriptar utilizando la técnica de “Salt” la contraseña del usuario.
- La aplicación debe contar con una pestaña consistente en “Alertas”, que organice la agenda de los mecánicos para ese día y para la próxima semana.
- El sistema debe acceder correctamente a los datos almacenados en la base de datos para todas las funcionalidades que lo necesiten desde JavaScript, así como desde Python para llevar a cabo el entrenamiento del algoritmo de regresión lineal múltiple.
- Siguiendo el punto anterior, el programa debe realizar correctamente una conexión con Python para la ejecución del Algoritmo de Regresión Lineal Múltiple cada vez que un mecánico realice y registre una revisión a una aeronave.
- La aplicación debe disponer de una pestaña en la que se pueda ver de un vistazo la situación general de los aviones en una tabla. Coloreados en verde estarán los aviones que son aptos para volar, y que, por tanto, solo necesitarán recibir una revisión rutinaria; mientras que en rojo se colorearán aquellas aeronaves que tengan algún fallo, y que, por tanto, precisen de un mantenimiento más exhaustivo. De igual forma, debe ser posible filtrar la búsqueda, tanto por número de elementos a revisar como por modelo de avión.
- Así mismo, la aplicación debe disponer de una pestaña para poder visualizar las revisiones que se han realizado hasta el momento. En esta misma pestaña se podrá consultar dónde se encuentran almacenados los informes en PDF de cada revisión.
- Por otro lado, el sistema debe contar con un perfil de administrador, que tenga funciones diferentes al mecánico.
- El administrador debe poder ver una lista de los empleados (mecánicos) que se encuentran en activo o no, modificar su estado o añadir empleados nuevos.
- El administrador debe poder modificar la lista de modelos de avión que estén en la flota de la compañía.

- El administrador debe poder cambiar su perfil a un perfil de mecánico para poder probar las funcionalidades del perfil de estos.

5.1.1.2 Requisitos No Funcionales

A continuación, se enumerarán diferentes requisitos no funcionales asociados a la calidad que cumplen:

- Seguridad y privacidad: El sistema debe garantizar principalmente la seguridad de los datos de la revisión almacenados en la cadena de bloques, pero también la información de los usuarios, puesto que la aplicación tiene en su poder información personal, además del DNI y la cuenta de Blockchain.
- Rendimiento: La aplicación desarrollada debe ser capaz de soportar grandes volúmenes de datos y responder a las peticiones de manera rápida con cual sea la cantidad de datos.
- Escalabilidad: El sistema debe ser capaz de manejar un incremento en número de aviones o en el registro constante de revisiones, puesto que el funcionamiento de las flotas de cada empresa es independiente.
- Disponibilidad: El sistema debe estar al alcance de la mano de los usuarios de manera confiable y continua, sin interrupciones.
- Interoperabilidad: El sistema debe ser compatible con otros sistemas, por ejemplo, mediante REST API, para poder integrarse con otros sistemas de la compañía.
- Modularidad y Mantenibilidad: La aplicación debe estar diseñada de forma modular, facilitando el mantenimiento o actualización en el futuro de diferentes bloques. Este requisito va en línea con la disponibilidad: la modularidad hace que las revisiones en un bloque de la aplicación no bloqueen el funcionamiento del resto de bloques.

5.1.2 CARACTERÍSTICAS DESTACADAS

El sistema desarrollado posee algunas características relevantes que resultan novedosas para el sector del mantenimiento en aviación.

5.1.2.1 Identificación Con Cuenta De Blockchain

Hasta los últimos años, el Blockchain ha estado únicamente asociado a las criptomonedas, sin ser investigados posibles usos más allá. En cambio, la idea que propone este proyecto involucra dicha tecnología, situándola en un foco principal. En el mundo del mantenimiento de aviación, así como en la industria en general, resulta innovador que un usuario de una Intranet de una empresa deba poseer una cuenta en Blockchain (proporcionada por la empresa) asociada a su usuario, de manera que quede un registro exacto de toda la información introducida, por quién, etc.

5.1.2.2 Sustitución Certificados En Papel Por Blockchain

Como se ha explicado previamente, el sistema de revisiones actual involucra al mecánico que la realiza y a un perito que certifica lo que el mecánico ha hecho o dicho, firmando un sinnúmero de papeles, que resultan contaminantes para el medio ambiente, así como ineficientes a la hora de realizar comprobaciones. La aplicación ayuda a reducir el número de personas que acceden a la información no verificada, reduciendo las posibilidades de cambios malintencionados, y sustituye las altas pilas de papeles, automatizando y digitalizando el proceso de certificación, de una manera mucho más segura: recurriendo a la cadena de bloques.

5.1.2.3 Uso De Machine Learning Para Fechas

Es habitual que los mecánicos estimen de manera automática la fecha de revisión de las aeronaves, principalmente las revisiones rutinarias: por ejemplo, todas cada 2 semanas; de manera general, para todas las aeronaves de la compañía por defecto. En cambio, la implementación de un algoritmo de Machine Learning como el de Regresión Lineal Múltiple supone un alto nivel de personalización a la hora de plantear las revisiones, ofreciendo más frecuencia a aquellas aeronaves que lo necesiten, en función de las características y la condición de cada una de ellas.

5.1.3 CASOS DE USO

En la siguiente figura se muestra el diagrama de casos de uso, en el cual intervienen tres actores: el usuario sin cuenta, el mecánico ya autenticado y el usuario con perfil de administrador.

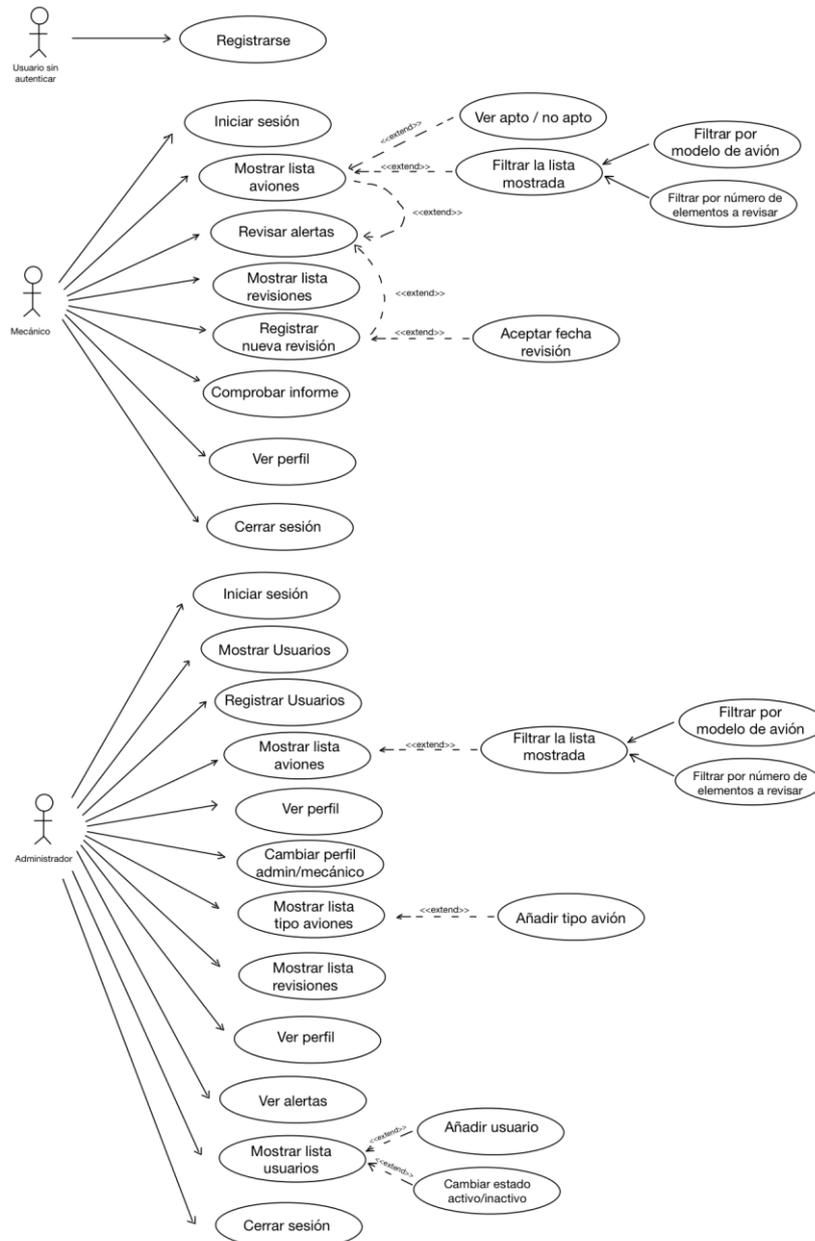


Figura 30. Diagrama de Casos de Uso

5.2 DISEÑO

A lo largo de este subapartado se explicará paso a paso el diseño del sistema desarrollado, explicando todos sus componentes y haciendo uso de diferentes diagramas UML, entre otras herramientas, para mostrar su funcionamiento.

5.2.1 DISEÑO DE LA ARQUITECTURA

En aras de cumplir los objetivos y requisitos previamente marcados para el proyecto, se ha realizado un diseño de arquitectura del sistema basado en cuatro bloques: la red Blockchain (Ethereum), una base de datos PostgreSQL, un algoritmo de Machine Learning y la web con la que interactúa directamente cada usuario de la web; todos ellos relacionados entre sí como se muestra en la figura:

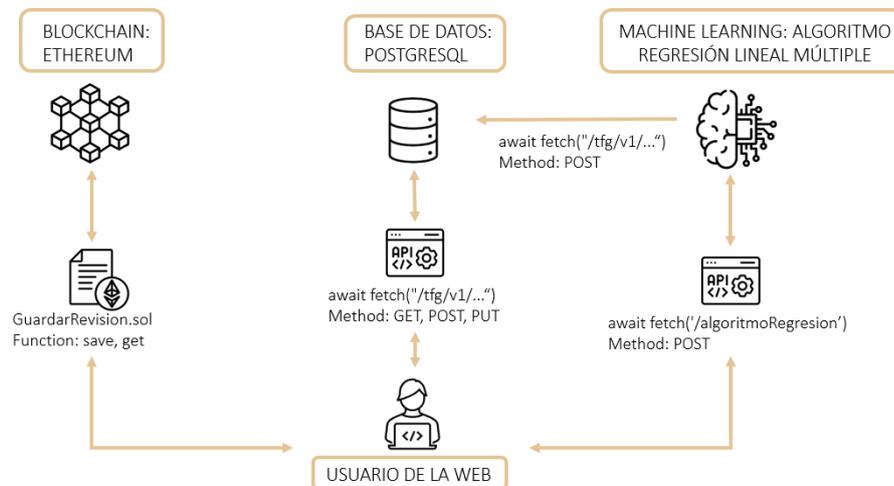


Figura 31. Diseño de la Arquitectura del Sistema

5.2.1.1 Red Blockchain

Para la red Blockchain, se ha optado por utilizar Ethereum debido a su soporte para Contratos Inteligentes, lo que permite que se ejecuten automáticamente acuerdos previamente definidos cuando se cumplan ciertas condiciones.

En el caso del lenguaje de programación, se ha elegido Solidity debido a la abundante documentación disponible: Solidity es ampliamente adoptado y cuenta con una comunidad activa, lo que facilita el aprendizaje y el desarrollo de Contratos Inteligentes en Ethereum.

En esta red, se almacenará un mapping que contendrá el número de revisión y el hash del PDF del informe previamente generado (más adelante se explica cómo se genera este informe), que contendrá toda la información de la revisión que se ha realizado y que se considere susceptible de modificación en caso de accidente. Cabe destacar que será este hash el que determine si un informe ha sufrido o no una modificación.

Cada vez que se lleve a cabo una revisión, el sistema llamará al contrato inteligente para guardar dicha información relevante en Blockchain. En cambio, si se realiza una modificación 'ilegal' o no autorizada, no se guardará en Blockchain y a la hora de comprobar su veracidad, la discrepancia entre el registro en la cadena de bloques y los datos del informe a comprobar será evidente, revelando que sea producido un cambio no autorizado en los informes PDF almacenados. Esto se debe a la naturaleza descentralizada y segura de Blockchain, en la cual es necesario el consenso de toda la red Blockchain para realizar cualquier transacción.

En resumen, la elección de Blockchain Ethereum y Solidity como lenguaje ofrece una base sólida para garantizar la integridad y transparencia de los informes de las diferentes revisiones a aviones.

5.2.1.2 Base De Datos PostgreSQL

Para el almacenamiento de la información de los aviones, revisiones y usuarios se ha optado por utilizar una base de datos PostgreSQL. Entre los motivos principales que han llevado a la toma de esta decisión se encuentra su compatibilidad con la herramienta de administración PgAdmin. PgAdmin ofrece una interfaz gráfica intuitiva que facilita las consultas y operaciones, y, en definitiva, la gestión de la base de datos. La compatibilidad con herramientas de este estilo simplifica las tareas de desarrollo de la aplicación y agiliza las tareas de gestión de los datos y las tablas.

Así mismo, en línea con el objetivo de proteger la información, se ha elegido PostgreSQL porque cumple con los estándares y regulaciones de seguridad de datos.

5.2.1.3 Algoritmo De Machine Learning

El tercer bloque en el que se divide el proyecto es el referente al Algoritmo de Machine Learning. Se ha optado por elegir un algoritmo regresión lineal múltiple. Dicho algoritmo es un ejemplo de aprendizaje supervisado, consistente en que se dispone de un conjunto de datos de entrenamiento – conjunto de train – donde las variables de entrada (características) y las variables de salida (etiquetas) están disponibles, y se utilizan para entrenar el algoritmo, y que sepa cómo reaccionar (calcular una salida) ante diferentes datos de entrada – conjunto de test. Esta elección se ha realizado, en primer lugar, debido a que se buscaba que la fecha de próxima revisión fuera estimada en función de más de una variable. Por otro lado, un motivo de peso también es que el conjunto de datos almacenados en la base de datos es considerablemente diverso y proporciona la información necesaria para poder realizar predicciones correctas. Cabe destacar que cuanto más amplio y variopinto sea el conjunto de entrenamiento del algoritmo, es más probable que el algoritmo aprenda los patrones y realice predicciones de manera más precisa.

5.2.1.4 Página Web

Por último, se encuentra el bloque referente a la página web, la cual desempeña un papel fundamental en la interacción del usuario y en la gestión de diversas actividades. Para dar funcionalidad al sistema desarrollado, se ha diseñado la web simulando la Intranet de una determinada empresa que decida comprar este software.

Una Intranet es una red privada que permite el intercambio de información y la colaboración entre los miembros de una misma organización. En este caso, entre las muchas funcionalidades, la Intranet sirve para organizar y mostrar el planning del día y la semana a los mecánicos, permitiéndoles así las tareas que deben llevar a cabo. Estas alertas ayudan a mantener un seguimiento y realizar una gestión efectiva del tiempo de trabajo de los empleados.

En segundo lugar, desde la Intranet de la empresa se pueden registrar las revisiones realizadas, cuya información posteriormente se guardará de manera segura en Blockchain. De igual forma, se puede mostrar un listado del estado actual de los aviones, así como un listado de revisiones realizadas. Esta funcionalidad permite mantener un registro actualizado de las revisiones realizadas, facilitando la supervisión y el seguimiento de la flota de aviones, y ayuda también a priorizar las tareas de mantenimiento según las necesidades.

Respecto a la veracidad de los informes, desde la página web se puede comprobar si son fieles a lo almacenado en Blockchain, y, por tanto, a lo introducido. Los usuarios pueden ingresar un informe específico, junto al número de revisión al que se refieren, y el sistema comprobará el hash de este informe con el hash almacenado en Blockchain. Esta característica refuerza la fiabilidad y la integridad de los informes realizados.

Por último, debido a que el mantenimiento en aviación es un asunto crítico, los mecánicos que realicen una revisión, comprobación, etc. deben estar siempre correctamente identificados. La web incluye un proceso de verificación de identidad utilizando Documento Nacional de Identidad (DNI) y añadiendo la cuenta de Blockchain al registrarse; garantizando que solo el personal autorizado tenga acceso a las funcionalidades y datos sensibles de la web.

En paralelo con todo lo referente a las actividades realizadas por parte de los mecánicos de la empresa, la Intranet también cuenta con un perfil de administrador, que lleva a cabo labores diferentes a los mecánicos.

En primer lugar, el administrador tiene la potestad de ver el listado de empleados que se encuentran en activo en la empresa, cambiar su estado, o añadir nuevos empleados, introduciendo toda su información y creándoles el perfil en la Intranet.

Por otro lado, el administrador también puede gestionar qué modelos de aviones se trabajan en la empresa, pudiendo añadir más si se añaden más aviones a la flota.

Así mismo, el administrador puede cambiar su perfil a un perfil de mecánico para ver cómo ven ellos la página web, y realizar las posibles modificaciones necesarias.

5.2.1.5 Relación Entre Los Cuatro Bloques

Los cuatro bloques previamente explicados se relacionan entre sí para aportar funcionalidad al proyecto, siendo la página web (y sus usuarios) el epicentro principal.

El usuario se relaciona con Blockchain mediante el Contrato Inteligente diseñado. Este contrato cuenta con 3 métodos, el constructor, un método save y un método get. El método save es el que el usuario utiliza para introducir el hash del informe previamente creado; mientras que el método get es el utilizado para leer dicho hash almacenado en Blockchain por cada revisión.

En segundo lugar, el usuario de la página web interactúa con la base de datos mediante endpoints REST API. Las tablas que se almacenan en la base de datos son: avión, revisión, usuario y tipo_avión. Para acceder a ellas se utilizan los métodos GET, POST y PUT.

En tercer lugar, el usuario utiliza el algoritmo de Machine Learning implementado en Python mediante un endpoint REST API, utilizando el método POST. Con él, introduce parámetros que luego se tendrán en cuenta en el algoritmo.

Y, por, último, también existe una relación mediante endpoints REST API entre la base de datos y el algoritmo de Machine Learning puesto que el resultado que se obtiene de la ejecución del algoritmo también se guarda en la base de datos, junto al resto de información de la revisión.

5.2.2 ESTRUCTURA DE DATOS

Como se puede ver en la arquitectura del sistema, en el diseño del proyecto se incluyen dos formas diferentes de almacenar datos: en una base de datos y en blockchain, ambas cada una útil para un tipo de información.

5.2.2.1 Base De Datos

En la base de datos se almacena toda la información referente a los usuarios, pero también a los aviones y sus revisiones debido a que almacenar absolutamente todo en Blockchain dispararía el precio del proyecto, ya que las transacciones de Blockchain cuestan dinero, como se ha explicado en el capítulo 4 de este proyecto. Para ello, en este proyecto, se ha hecho uso de tres tablas, que se definirán a continuación, y, posteriormente, se mostrarán las relaciones entre ellas:

- usuario: esta tabla se almacena la información personal de cada mecánico. Entre las columnas se encuentra: ID de usuario, salt, hash de la contraseña, su email y teléfono, y, por último, los principales elementos identificativos de cada usuario: su DNI y su cuenta de Blockchain.
- avión: en esta tabla se guarda toda la información referente al estado actual de la flota de aviones de la empresa que utilice la Intranet. Cada vez que se realiza una revisión se actualiza de manera que, mostrando el contenido de esta tabla, se pueda conocer por inspección visual que aeronaves se encuentran disponibles para volar en el momento.
- revisión: en esta tercera tabla se lleva un registro de todas las revisiones que se han realizado, incluyendo información de los elementos revisados, y de las próximas revisiones programadas.
- tipo avión: en esta última tabla, se almacenan los modelos de avión de los que dispone la compañía. Esta tabla está relacionada con la tabla principal, la de avión mediante una clave foránea a la columna modelo.

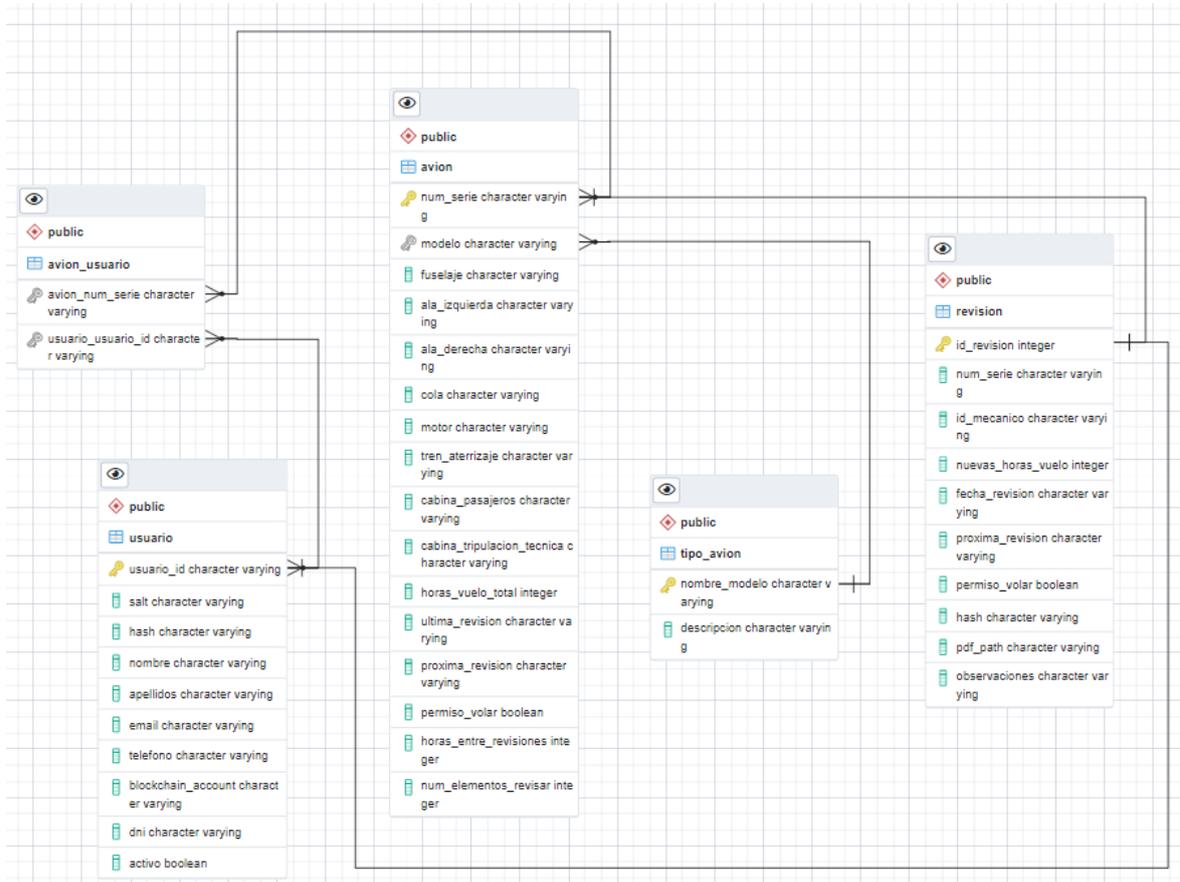


Figura 32. Diagrama Entidad Relación Base de Datos

Tal y como se muestra en la Figura 32. Diagrama Entidad Relación Base de Datos', todos los usuarios pueden realizar todas las modificaciones que considere pertinentes en la tabla avión, actualizando todos los campos que cambien en cada revisión realizada. De la misma forma, la tabla avión puede ser modificada por todos los usuarios.

Respecto a la relación entre la tabla de usuario con la de revisión, cada usuario puede realizar varias revisiones; pero cada revisión puede ser realizada únicamente por un usuario.

Las tablas de avión y tipo avión están relacionadas por una clave foránea, desde la columna modelo de la tabla avión.

Por último, entre las tablas avión y revisión, existe una relación 1-N. Cada revisión corresponde únicamente a un avión, pero un avión puede tener múltiples revisiones.

5.2.2.2 Blockchain

Para llevar a cabo la funcionalidad deseada de Blockchain, ha sido necesario desarrollar una única clase. Esta clase cuenta con tres atributos y tres métodos que se explicarán a continuación.

Los atributos son los siguientes:

```
uint256 revisionId;  
  
string hashCod;  
  
mapping (uint256 => string) public revisionHashes;
```

- **revisionId:** Un entero sin signo de 256 bits que representa el ID de revisión. El rango de valores posibles para un uint256 va desde 0 hasta $2^{256}-1$. Al ser un entero sin signo, el tipo uint256 no permite representar valores negativos. También es importante destacar que el uso de uint256 consume más espacio de almacenamiento y puede requerir más gas en las transacciones de Ethereum en comparación con tipos de datos más pequeños, como uint8 o uint16.
- **hashCod:** Una cadena de texto que almacena el código hash. En este caso, el funcionamiento del string es como en otros lenguajes de programación: se utiliza para almacenar y manipular cadenas de texto. Se puede componer de cualquier combinación de caracteres individuales (letras, números, símbolos o espacios en blanco). En este caso, al contener únicamente hashes, serán de la forma '0xAAAA', siendo AAAA una combinación larga de números y letras.
- **revisionHashes:** Un mapeo que asocia un ID de revisión (uint256) con un código hash (string). Es público, lo que significa que se puede acceder a él desde fuera del contrato. En programación, un mapping (o mapeo) es una estructura de datos que permite relacionar una clave (key) con un valor (value). Cada clave está asociada a un único valor; y se puede asemejar a los diccionarios en otros lenguajes de

programación. En Blockchain, se utiliza la clave para acceder a la información contenida en su valor correspondiente.

Y los métodos son los siguientes:

```
constructor() {}

function saveRevisionHash(uint256 _revisionId, string memory _hashCod) public
{
    revisionHashes[_revisionId] = _hashCod;
}

function getRevisionHash(uint256 _revisionId) public view returns (string memory)
{
    return revisionHashes[_revisionId];
}
```

- **constructor():** El constructor del contrato. Se ejecuta una vez durante la creación del contrato, al inicio, con el objetivo de inicializar los atributos o propiedades del objeto recién creado. Es importante destacar que en el resto de los lenguajes de programación, el método constructor se define utilizando el mismo nombre que la clase, pero en Solidity se define utilizando la palabra ‘constructor’. A diferencia de otros métodos, es un método sin retorno y se ejecuta automáticamente.
- **saveRevisionHash(uint256 _revisionId, string memory _hashCod) public:** Es una función pública definida en el contrato, cuyo propósito es permitir guardar un código hash en el mapeo ‘revisionHashes’. Dentro de la función, el código asigna el valor del código hash _hashCod al mapeo revisionHashes utilizando el ID de revisión _revisionId como clave. Esto se realiza mediante la siguiente instrucción: `revisionHashes[_revisionId] = _hashCod;`
- **getRevisionHash(uint256 _revisionId) public view returns (string memory):** Se trata de una función pública y de solo lectura (es decir, no realiza modificaciones en el contrato) que devuelve el código hash correspondiente a un ID de revisión específico. Dentro de la función, el código accede al mapeo revisionHashes y devuelve el valor asociado con la clave revisionId. Esto se logra mediante la instrucción: `return revisionHashes[_revisionId];`

5.2.3 DIAGRAMAS DE SECUENCIA

Finalmente, para completar el diseño del sistema a desarrollar en el Proyecto, se han creado dos diagramas de secuencia en los que se podrá ver los pasos que seguirán y el flujo que los usuarios realizarán al llevar a cabo una acción.

Se va a realizar el diagrama de secuencia del procedimiento que se sigue cuando el usuario introduce una revisión en Blockchain, y el procedimiento que se sigue cuando un usuario comprueba la veracidad de un informe. Se han elegido estos dos casos puesto que suponen la funcionalidad principal del proyecto realizado.

El primer escenario contempla el proceso que se sigue para introducir una nueva revisión en Blockchain.

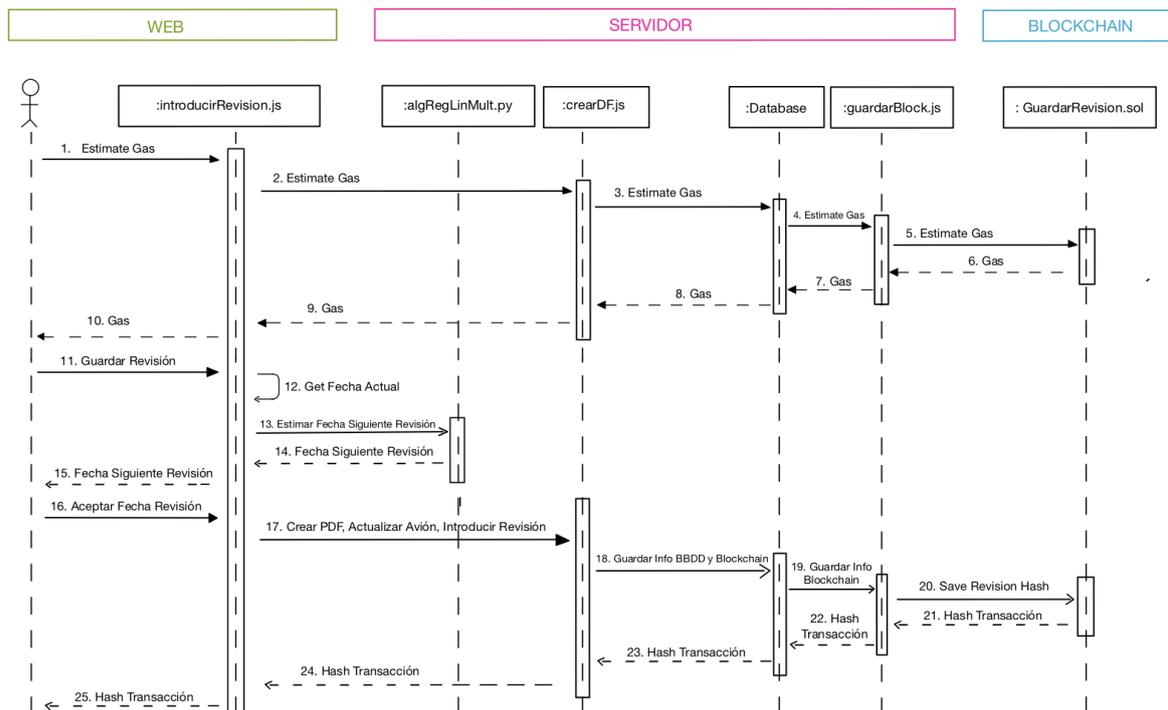


Figura 33. Diagrama de Secuencia - Introducir Revisión

Puesto que vamos a guardar información el Blockchain, la transacción tendrá un precio (de gas). Antes de comenzar, se deberá realizar una estimación del gas necesario para calcular

las tasas. A partir de ahí, se ordena que se inicie el proceso. Es importante destacar que en el JS de introducirRevisión se realiza una verificación de la corrección de los datos, por lo que no hay posibilidad de error en el proceso.

El segundo escenario que se va a explicar es la comprobación de la validez de un informe.

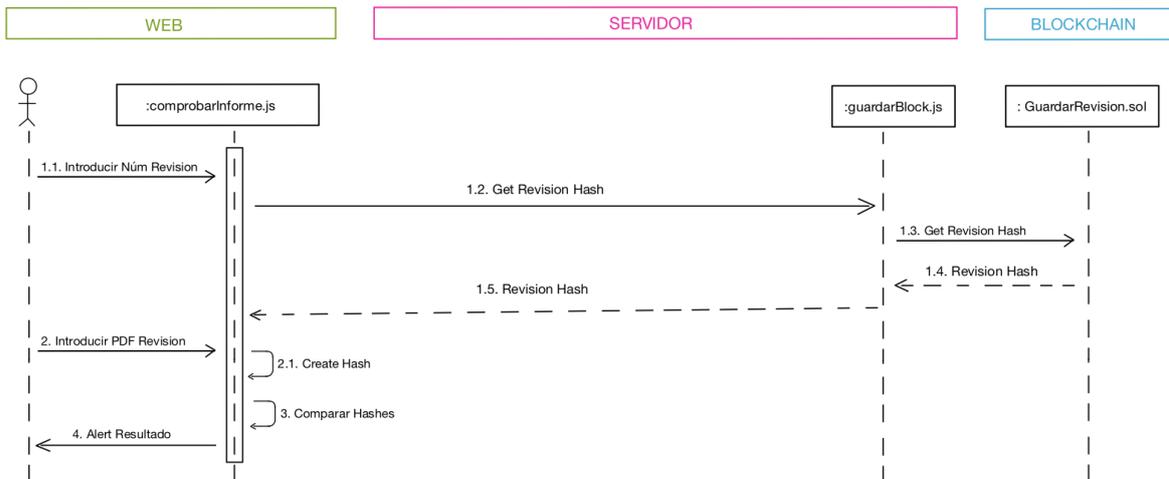


Figura 34. Diagrama de Secuencia - Comprobar Informe

En este caso, es importante destacar que se realizará una operación GET, por lo que no tendrá coste de gas, y, por tanto, no es necesario estimarlo. El archivo PDF se introducirá desde los archivos del equipo del mecánico.

5.3 IMPLEMENTACIÓN

A continuación, se explicará el desarrollo de la implementación del proyecto, dividiendo la explicación en los cuatro bloques principales, que son el Contrato Inteligente para Blockchain, la base de datos para almacenar la información, el algoritmo de regresión lineal múltiple de Machine Learning, y, por último, la página web. Asimismo, se incluirá parte del código del Contrato Inteligente y del algoritmo de regresión lineal múltiple. El resto del código, debido a su extensión, se publicará en un repositorio GitHub para facilitar el acceso al mismo.

5.3.1 CONTRATO INTELIGENTE

Como se ha explicado previamente y se detalla a continuación, el contrato inteligente utilizado en este proyecto cuenta con tres atributos y tres métodos. Los atributos son los siguientes:

```
uint256 revsisionId;  
  
string hashCod;  
  
mapping (uint256 => string) public revisionHashes;
```

El uint256 hace referencia al número de revisión que se almacenará; el string será el hash del PDF que se querrá almacenar en Blockchain; y, por último; el mapping agrupará el número de revisión con el hash de su PDF correspondiente para ser guardados en Blockchain.

Respecto a los métodos, se enumeran por separado a continuación:

```
constructor() {}
```

En primer lugar, está el método constructor, al igual que en todos los lenguajes de programación, con la diferencia de que en Solidity se debe llamar constructor siempre. Este método se ejecuta automáticamente cuando se inicia el Contrato Inteligente, y su principal función es inicializar atributos y demás métodos.

```
function saveRevisionHash(uint256 _revisionId, string memory _hashCod) public  
{  
    revisionHashes[_revisionId] = _hashCod;  
}
```

Esta segunda función es la utilizada para introducir cada par de valores en el mapping revisionHashes. Sus componentes son:

- **saveRevisionHash:** Es el nombre de la función.
- **uint256 _revisionId:** Es el primer argumento de la función y representa el ID de la revisión que se va a guardar. Es de tipo uint256, que es un entero sin signo de 256 bits.

- **string memory _hashCod**: Es el segundo argumento de la función y representa el código hash que se va a guardar. Es de tipo string, que es una cadena de caracteres.
- **public**: Es un modificador de acceso que indica que la función es pública, lo que significa que puede ser llamada desde fuera del contrato.
- **mapping (uint256 => string) public revisionHashes**: Es el mapeo llamado revisionHashes que se utiliza para almacenar los códigos hash. Es un mapeo que asocia valores de tipo uint256 (IDs de revisión) con valores de tipo string (códigos hash). Al declararlo como public, se genera automáticamente una función getter para acceder a los valores almacenados en el mapeo.
- **function saveRevisionHash(uint256 _revisionId, string memory _hashCod) public**: Es la declaración completa de la función, que indica el nombre, los argumentos y los modificadores de acceso.

```
function getRevisionHash(uint256 _revisionId) public view returns (string memory)
{
    return revisionHashes[_revisionId];
}
```

La tercera función es la utilizada para leer de Blockchain en la función de comprobar los Informes. Devuelve el valor asociado a la clave que se introduzca como argumento. Sus componentes son:

- **getRevisionHash**: Es el nombre de la función.
- **uint256 _revisionId**: Es el único argumento de la función y representa el ID de revisión para el cual se desea obtener el código hash. Es de tipo uint256, que es un entero sin signo de 256 bits.
- **public**: Es un modificador de acceso que indica que la función es pública y puede ser llamada desde fuera del contrato.
- **view**: Es un modificador que indica que la función no realiza modificaciones en el contrato y solo lee datos. Esto es útil cuando se desea obtener información sin incurrir en costos de gas en la cadena de bloques.

- returns (string memory): Especifica el tipo de dato que devuelve la función, en este caso, un valor de tipo string que representa el código hash correspondiente al ID de revisión proporcionado.

5.3.2 BASE DE DATOS

La base de datos que se ha utilizado a lo largo del proyecto es una base de datos de PostgreSQL, principalmente por su integración con la herramienta PgAdmin para la administración de esta. Como se ha mencionado previamente, se han creado cuatro tablas que interaccionan entre sí. De igual manera, para interactuar con las tablas desde la aplicación, se utilizan diferentes endpoints. A continuación, se muestra un pequeño fragmento de cada tabla, donde se pueden ver las columnas que las conforman, junto con los endpoints utilizados para interacciona con ella.

num_serie	modelo	fuelaje	ala_izquierda	ala_derecha	cola	motor	tren_averiaje	cabina_pasajeros	cabina_tripulacion_tecnica	horas_vuelo_total	ultima_revision	proxima_revision	permiso_volar	horas_entre_revisiones	num_alimentacion_revisar
1 31901	A319	apto	apto	apto	apto	apto	revisar	apto	apto	22 2023-05-12	2023-06-03	false		543	1
2 31902	A319	revisar	revisar	revisar	revisar	revisar	revisar	revisar	revisar	239 2023-06-02	2023-06-25	false		80	8
3 31903	A319	apto	apto	apto	apto	apto	apto	apto	revisar	171 2023-06-02	2023-06-17	false		320	3
4 31904	A319	apto	apto	apto	apto	apto	apto	apto	apto	2 2023-06-04	2023-05-25	true		521	0
5 31905	A319	revisar	apto	apto	apto	revisar	revisar	apto	apto	119 2023-05-23	2023-06-09	false		278	3
6 31906	A319	revisar	revisar	apto	apto	revisar	revisar	apto	apto	122 2023-04-12	2023-04-17	false		121	4
7 31907	A319	apto	apto	apto	apto	revisar	revisar	apto	apto	81 2023-06-01	2023-06-14	false		320	2
8 31908	A319	revisar	revisar	revisar	revisar	revisar	revisar	apto	apto	160 2023-05-28	2023-06-08	false		280	5
9 31909	A319	apto	apto	apto	apto	apto	revisar	apto	apto	41 2023-06-13	2023-06-30	false		429	1
10 31910	A319	apto	revisar	revisar	revisar	revisar	revisar	revisar	revisar	200 2023-06-18	2023-06-23	false		179	7

Figura 35. Tabla Avión

- GET: /tfg/v1/aviones → devuelve todos los aviones de la BBDD
- GET: /tfg/v1/aviones/ordered → devuelve todos los aviones de la BBDD ordenados por su número de serie
- GET: /tfg/v1/aviones/{numSerie} → devuelve el avión cuyo número de serie sea el especificado
- PUT: /tfg/v1/aviones → actualiza la información del avión que se le pasa como argumento
- POST: /tfg/v1/aviones → añadir un nuevo avión a la BBDD
- DELETE: /tfg/v1/aviones/{numSerie} → eliminar de la BBDD el avión especificado
- GET: /tfg/v1/aviones/modelo/{modelo} → devuelve todos los aviones del modelo especificado

- PUT: /tfg/v1/usuarios → actualiza la información del usuario que se le pasa como argumento
- POST: /tfg/v1/usuarios → añadir un nuevo usuario a la BBDD
- DELETE: /tfg/v1/usuarios/{usuarioId} → eliminar de la BBDD el usuario especificado

5.3.3 ALGORITMO DE MACHINE LEARNING

El tercer bloque en el que se divide el proyecto es un algoritmo de Machine Learning, y, como se ha mencionado previamente, se ha elegido utilizar un algoritmo de regresión lineal múltiple, puesto que se disponía de información de varias entradas y de la salida. Estas variables son:

- De entrada:
 - **Número de horas de vuelo totales del avión:** Se tienen en cuenta el número de horas de vuelo que se han realizado desde el inicio de la vida útil de la aeronave. Cuantas más horas de vuelo lleve a sus espaldas la aeronave, más frecuentes serán las revisiones.
 - **Número de elementos a revisar desde la última revisión:** En cada revisión realizada, se hace un recuento del número de elementos a revisar. Esto es porque se estima que a mayor número de elementos a revisar o modificar, mayor será el tiempo que hay que dejar para llevarlo a cabo.
- De salida:
 - **Número de horas entre revisiones:** Con los datos de entrada, es preciso estimar una cifra. Es por eso, que el algoritmo estima el número de horas entre la revisión actual y la próxima. Para la obtención de la fecha, a posteriori, se suman estas horas y se muestra por pantalla el resultado del día calculado.

El algoritmo se realiza en Python, utilizando las librerías Psycopg2, Statsmodel y ScikitLearn.

Para disponer de los datos, se debe conectar el archivo de Python con la base de datos usando la función connect de la librería psycopg2, realizar la query, y, por último, cerrar la conexión. El código para el proceso es el siguiente:

```
#Conectar con la base de datos

conn = psycopg2.connect(host="localhost", port="5432", user="postgres",
password="postgres", database="tfg")
cursor = conn.cursor()

#Realizar la query

query = "SELECT horas_vuelo_total, num_elementos_revisar, horas_entre_revisiones
FROM avion"
cursor.execute(query)
data = cursor.fetchall()

# Cerrar la conexión a la base de datos

cursor.close()
conn.close()
```

A continuación, se separan los datos para entrenar el algoritmo:

```
X_train = [[row[0], row[1]] for row in data]
y_train = [row[2] for row in data]
```

El siguiente paso será crear una instancia del modelo de regresión lineal usando LinearRegression() de ScikitLearn, que proporciona un conjunto de métodos y atributos para entrenar y utilizar el modelo de regresión lineal; como fit(), que en este caso se utilizará para ajustar el algoritmo.

```
modelo = LinearRegression()

# Ajustar el modelo a los datos de entrenamiento
modelo.fit(X_train, y_train)
```

Como en todo algoritmo de Machine Learning, se deberán fijar los datos de test, y, por último, realizar la predicción con el método predict() de ScikitLearn – LinearRegression().

```
X_test = [[horas_vuelo_total_new, num_elementos_revisar_new]]
y_test = modelo.predict(X_test)
```

Es importante destacar que, para hacer uso de este algoritmo desarrollado en Python desde JavaScript e incorporarlo a la página de introducirRevision, ha sido necesario crear un endpoint de tipo REST API y ejecutar un método POST. La llamada ha sido la siguiente:

```
var parametros = {
  "horasVueloTotal": horasVueloTotal,
  "numElementosRevisar": numElementosRevisar
};

let res = await fetch('/algoritmoRegresion', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(parametros)
})
const horasASumar = await res.json();
```

5.3.4 PÁGINA WEB

El último gran bloque de este proyecto es la página web, tipo intranet que une todos los bloques explicados anteriormente. La página web se ha dividido en dos según el tipo de usuario que acceda a ella: usuarios mecánicos y usuarios administrador. La diferencia fundamental de estos dos tipos de perfiles recae en las funciones que puede realizar cada uno de ellos.

Al igual que los accesos a la base de datos, los accesos a las diferentes páginas de la web se hacen mediante endpoints a REST APIs. Todas ellas comienzan por 'http://localhost:8080/'. Se mencionarán y explicarán las más relevantes.

- introducirRevision.html → esta es la ventana para que los mecánicos introduzcan las revisiones que han realizado, y donde aparece la fecha estimada por el algoritmo de Machine Learning. Desde aquí se ejecuta la llamada al js que realiza el informe PDF y el guardado del hash de dicho PDF en Blockchain. Antes de que todas estas funciones se realicen, el usuario deberá haber introducido de forma manual los datos obtenidos de la revisión, así como las observaciones que pueda aportar.
- comprobarInforme.html → desde esta ventana los usuarios tipo mecánico pueden comprobar la veracidad de los informes que se encuentran guardados en sus

archivos. Estos deberán introducir en un campo el número de revisión de la que se trata, y en otro campo el archivo PDF como tal. Desde esta clase se llamará a un método que creará el hash del PDF que hemos introducido y lo comparará con el que se ha leído de Blockchain. En caso de que coincidan, se avisará de que todo es correcto, y en caso de que no, se notificará el error.

- `mostrarListaAviones.html` → junto con la clase `mostrarListaRevisiones.html`, estas clases son las únicas que ambos perfiles tienen a su alcance. En la primera, se muestra un listado con toda la flota de aviones de la compañía, con su estado actual, y coloreada en verde o rojo según si la aeronave es apta o no para realizar un vuelo.
- `mostrarListaRevisiones.html` → en este caso, se muestra un listado con las revisiones realizadas, mostrando también la ruta para el archivo PDF.
- `mostrarListaUsuarios.html` → esta clase es exclusiva del perfil administrador. En ella, puede ver qué empleados tiene la empresa, cuáles se encuentran en activo, así como modificar su estado de actividad y añadir empleados nuevos, registrándoles en la Intranet.
- `mostrarListaTipoAviones.html` → esta clase también es exclusiva del perfil administrador, ya que en ella se muestran los tipos de aviones (modelos) que tiene la empresa, y se pueden añadir nuevos modelos en la Intranet si la flota se modifica.
- `alertas.html` → en esta pantalla se muestran las revisiones pendientes de realizar en el día de hoy y en la próxima semana, de manera que la agenda de los mecánicos queda organizada. Es importante destacar que ambos perfiles pueden ver esta pantalla, pero con una pequeña diferencia: los mecánicos pueden iniciar una revisión desde esta pantalla y mostrar la información de los aviones; mientras que el administrador únicamente puede mostrar la información, ya que el administrador no realiza revisiones.
- `inicioSesion.html` → pantalla para iniciar sesión y entrar a la Intranet. Si eres un mecánico sin usuario, desde esta pantalla puedes acceder a la pantalla de registro.

En esta clase se comprobará si el usuario y contraseña son correctos, pero también si la cuenta de Blockchain asociada a la cuenta está activada con MetaMask.

- registro.html → Todos los mecánicos sin cuenta podrán desde esta pestaña crearse una. Para ello, necesitarán sus datos personales, su DNI y su cuenta de Blockchain conectada a MetaMask. En caso de no cumplir estos requisitos, no se podrá crear el perfil. Respecto a la contraseña, se almacena en una base de datos salteada; es decir, a la contraseña se le añade un ‘salt’ (calculado con un programa de Python, llamado mediante un endpoint de una REST API) y luego todo el conjunto se hashlea; y en la base de datos sólo se almacena el salt y el hash, de forma que la contraseña queda asegurada.
- introducirUsuarioAdmin.html → es una clase muy similar a la clase registro, con la salvedad de que no creas la cuenta para ti, sino que es el administrador el que lo lleva a cabo.
- perfilUsuario.html y perfilAdmin.html → estas dos clases, cada una correspondiente a un tipo de perfil, muestran la información del usuario que ha iniciado sesión. El detalle que marca la diferencia en ellas es el hecho de que la clase perfilAdmin.html dispone de un botón para cambiar el modo: de Admin a Mecánico para simular cómo ven los mecánicos la web desde sus perfiles.

Es importante destacar que todas las páginas que no son inicioSesión.html o registro.html requieren estar autenticado, por lo que, si se intenta acceder a ellas sin haber iniciado sesión previamente, la página web se dirigirá automáticamente a la página de inicioSesion.html.

Respecto al aspecto de la web, se han implementado pantallas sencillas, con un diseño minimalista y con una barra de navegación entre las diferentes páginas en la parte superior para simplificar su uso a personas sin conocimientos de programación.

Por otro lado, la página web es responsiva, por lo que se adapta al tamaño de cualquier pantalla de ordenador, facilitando así su uso en diferentes equipos informáticos, desde ordenadores portátiles pequeños, a monitores de TV.

Capítulo 6. ANÁLISIS DE RESULTADOS Y

CONCLUSIONES

Tras haber realizado la implementación del sistema tal y como se describe en el capítulo 5 de esta memoria, se concluye en que los resultados del proyecto han sido satisfactorios, y se han logrado los objetivos establecidos.

A lo largo de este capítulo se explicarán los resultados obtenidos, y se analizará el producto final del proyecto y el que los usuarios experimentarán: la página web.

6.1 PRIMER OBJETIVO – BLOCKCHAIN + PDF

El primer objetivo implica la implementación y el posterior despliegue de un contrato inteligente que permita a los usuarios almacenar el hash de un informe PDF previamente creado en la cadena de bloques, así como extraer dicho hash. Este objetivo se ha visto realizado en el apartado 5.3.1, Implementación de Blockchain, en las pestañas de introducirRevision y comprobarInforme, respectivamente. El resultado se muestra en las figuras: **¡Error! No se encuentra el origen de la referencia., ¡Error! No se encuentra el origen de la referencia., ¡Error! No se encuentra el origen de la referencia. y ¡Error! No se encuentra el origen de la referencia..**

6.1.1 INTRODUCIR REVISIÓN

En esta pantalla se desarrolla una de las funcionalidades más importantes. El usuario debe introducir la información de la revisión que ha realizado, para que, a continuación, se cree el PDF, y su correspondiente hash; el cual se almacenará en Blockchain.

Introduce los datos de la revision que hayas realizado

Numero De Serie Del Avión:

Nuevas Horas De Vuelo:

Modelo:

Fuselaje: Apto Revisar

Ala Izquierda: Apto Revisar

Ala Derecha: Apto Revisar

Cola: Apto Revisar

Motor: Apto Revisar

Tren De Aterrizaje: Apto Revisar

Cabina Pasajeros: Apto Revisar

Cabina Tripulación Técnica: Apto Revisar

Observaciones:

Calcular fecha estimada próxima revisión

Figura 39. Pantalla Introducir Revisión Vacía

Tras introducir toda la información, generar la fecha estimada y aceptarla, se creará el informe PDF, que solicitará permiso para acceder a los archivos, para encontrar la ubicación donde almacenarlo:

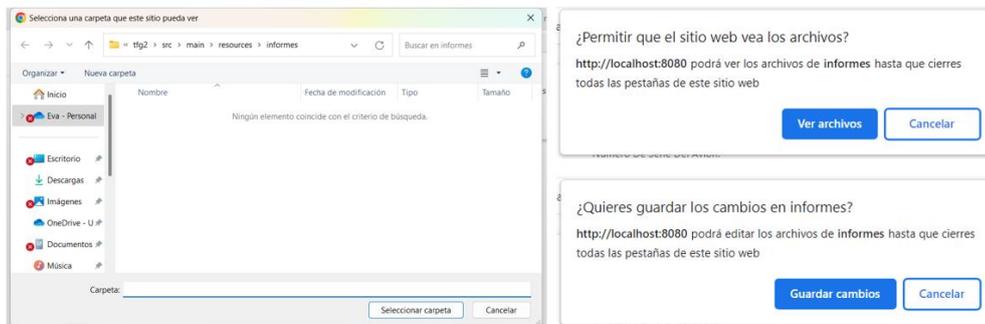
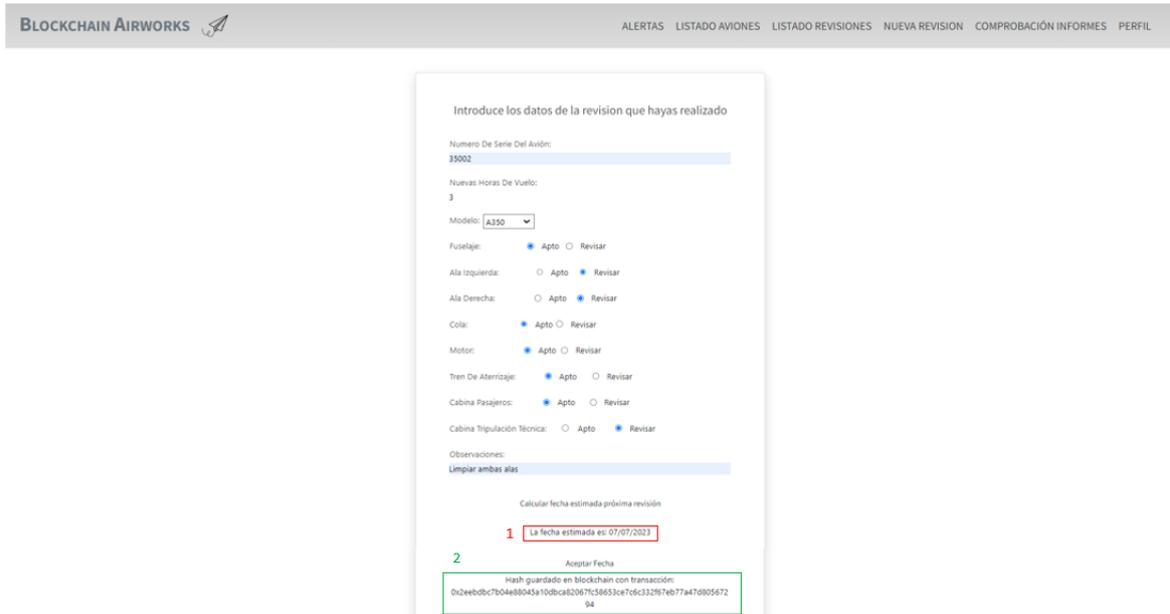


Figura 40. Permisos para almacenar informe PDF

Finalmente, tras aceptar todo lo que se solicita, se mostrará una ventana final con toda la información introducida, la nueva fecha (1) y el hash de la transacción en Blockchain (2), probando que esta transacción efectivamente ha sido realizada.



Introduce los datos de la revision que hayas realizado

Numero De Serie Del Avión:
35002

Nuevas Horas De Vuelo:
3

Modelo: A350

Fuselaje: Apto Revisar

Ala Izquierda: Apto Revisar

Ala Derecha: Apto Revisar

Cola: Apto Revisar

Motor: Apto Revisar

Tren De Atarizaje: Apto Revisar

Cabina Pasajeros: Apto Revisar

Cabina Tripulación Técnica: Apto Revisar

Observaciones:
Limpiar ambas alas

Calcular fecha estimada próxima revisión

1 La fecha estimada es: 07/07/2023

2 Aceptar Fecha

Hash guardado en blockchain con transacción:
0x2eebdc7b04e88045a10dbca82097fc58653ce76dc3329f7b77a47d80567294

Figura 41. Pantalla Introducir Revisión Parte Final

6.1.2 COMPROBAR INFORME

Un segundo pilar de vital importancia para el proyecto es la comprobación de la veracidad de los informes, llevada a cabo introduciendo el número de la revisión que deseas comprobar (1), junto al archivo PDF de esa revisión (2). Tras la comprobación, se informará de si todo es correcto o si han ocurrido fallos.



Introduce el número de revisión y el PDF que deseas comprobar

1 ID Revisión:
16

2 PDF:
Seleccionar archivo informe16.pdf

Comprobar

Figura 42. Pantalla Comprobar Informe

6.2 SEGUNDO OBJETIVO – MACHINE LEARNING

El segundo objetivo consistía en la implementación de un algoritmo de Machine Learning, más concretamente, Algoritmo de Regresión Lineal Múltiple para predecir en qué momento es necesario realizar una revisión de mantenimiento, basándose en el número de elementos a revisar y las horas de vuelo realizadas por un avión. Se puede apreciar la consecución de este objetivo en el apartado 5.3.3, Implementación del Algoritmo de Machine Learning, y los resultados en la página web se mostrarán en la figura siguiente:

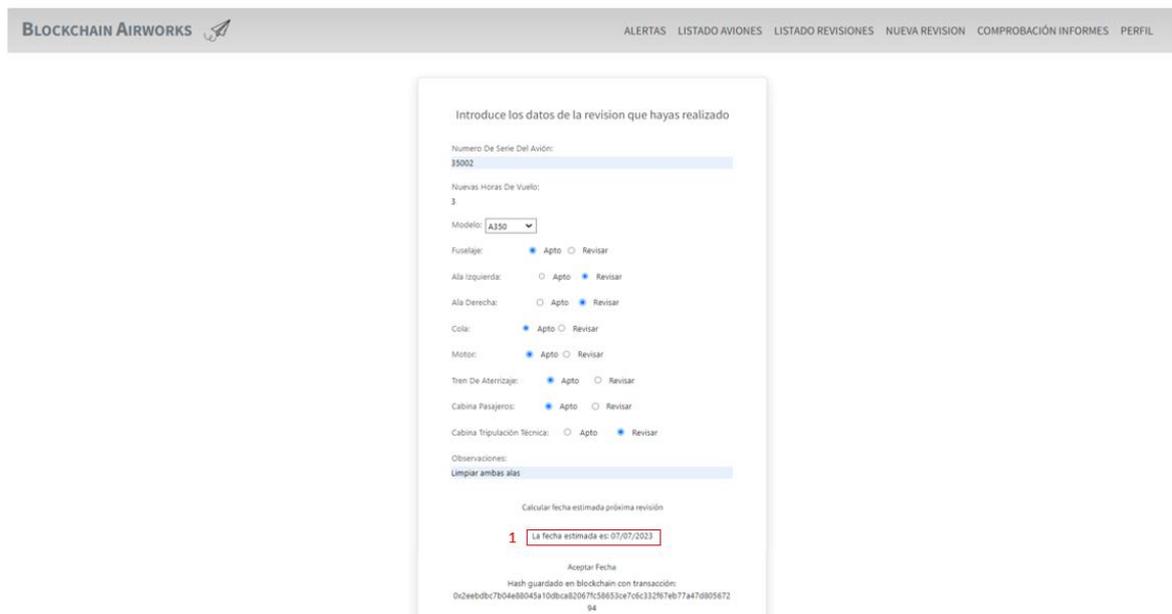


Figura 43. Pantalla Final Introducir Revisión

Al final de la página, se muestra que se ha realizado una predicción de la fecha (1) en función de la información introducida y de la información almacenada en la base de datos. Esta será la fecha que se guardará en el informe, y, por tanto, también en Blockchain.

6.3 TERCER OBJETIVO – IMPLEMENTACIÓN PÁGINA WEB

En tercer lugar, se marcó como objetivo implementar una página web tipo Intranet, segura y accesible donde se pudieran realizar todas las funcionalidades mencionadas a lo largo de

la memoria. Se puede ver el procedimiento seguido para la implementación de este objetivo el punto 5.3.4. de esta memoria; e imágenes de las diversas pantallas en el Anexo III Manual de Uso.

Cabe destacar que al principio se había marcado como objetivo una página austera, pero finalmente se han añadido mejoras de diseño y funcionalidad; como el registro de usuarios, el poder ver el perfil, o el hecho de poder mostrar los listados de información incluidos en la base de datos.

6.4 CUARTO OBJETIVO – PERFIL ADMINISTRADOR

Por último, durante la realización del proyecto se ha añadido un objetivo extra para cumplir funcionalidades que eran necesarias: la implementación de un perfil de administrador que gestione las cuentas del resto de usuarios (mecánicos) así como otros asuntos referentes a la flota de aviones. Se puede ver cómo se ha realizado la implementación en el apartado 5.3.4 de la memoria, y sus resultados en las siguientes figuras.



BLOCKCHAIN AIRWORKS 

ALERTAS LISTADO AVIONES LISTADO TIPO AVIONES LISTADO USUARIOS PERFIL

LISTADO TIPO AVIONES

1

Nombre de modelo	Descripción
A220	El Airbus A220 es un avión de pasajeros de tamaño medio diseñado para vuelos de corto y medio alcance. Ofrece una cabina espaciosa y cómoda, capacidad para entre 100 y 150 pasajeros y tecnologías avanzadas de eficiencia de combustible y reducción de ruido. Es una opción popular para aerolíneas que buscan mejorar su rentabilidad y reducir su impacto ambiental.
A330-300	El A330-300 es un avión de pasajeros de fuselaje ancho fabricado por Airbus. Es conocido por su versatilidad y capacidad de largo alcance. Con capacidad para transportar entre 277 y 440 pasajeros, el A330-300 ofrece un equilibrio entre eficiencia y comodidad. Cuenta con una configuración de cabina espaciosa y moderna, con opciones de asientos flexibles y sistemas de entretenimiento a bordo. Su diseño aerodinámico y eficiente motores le permiten volar largas distancias de manera eficiente y reducir el consumo de combustible.
A330-200	El A330-200 es un avión de pasajeros de fuselaje ancho fabricado por Airbus. Es reconocido por su capacidad de largo alcance y su versatilidad en diversas rutas. Con capacidad para transportar entre 246 y 406 pasajeros, el A330-200 ofrece un equilibrio entre eficiencia y comodidad. Cuenta con una cabina espaciosa y moderna, con opciones de asientos configurables y sistemas de entretenimiento a bordo. Su diseño aerodinámico y eficiente motorización le permite realizar vuelos de larga distancia de manera eficiente y reducir el consumo de combustible.
A321	El A321 es un avión de pasajeros de fuselaje estrecho fabricado por Airbus. Es conocido por su eficiencia y versatilidad en rutas de corto y medio alcance. Con capacidad para transportar entre 185 y 240 pasajeros, el A321 ofrece una opción ideal para aerolíneas que buscan una mayor capacidad en comparación con otros modelos de la familia A320. Cuenta con una cabina moderna y cómoda, con opciones de configuración flexibles y sistemas de entretenimiento a bordo. Gracias a su diseño aerodinámico y motores eficientes, el A321 logra un menor consumo de combustible y emisiones reducidas.
A320	El A320 es un avión de pasajeros de fuselaje estrecho fabricado por Airbus. Es ampliamente reconocido por su eficiencia y versatilidad en rutas de corto y medio alcance. Con capacidad para transportar entre 150 y 180 pasajeros, el A320 ofrece una opción popular para aerolíneas y pasajeros. Cuenta con una cabina moderna y confortable, con sistemas de entretenimiento a bordo y opciones de configuración flexibles. Gracias a su diseño aerodinámico y motores eficientes, el A320 logra un menor consumo de combustible y emisiones reducidas. Es un avión confiable y utilizado por aerolíneas de todo el mundo para vuelos cortos y medios.
A320neo	El A320neo (New Engine Option) es una versión actualizada del avión de pasajeros A320, fabricado por Airbus. Esta variante se caracteriza por contar con motores más eficientes y mejoras aerodinámicas. El A320neo ofrece una mayor eficiencia de combustible y una reducción significativa en las emisiones de CO2 y en la huella de ruido. Además, proporciona una mayor autonomía de vuelo y capacidad de carga. La cabina del A320neo mantiene la comodidad y configuración flexible del modelo original, incluyendo sistemas de entretenimiento actualizados.
A319	El A319 es un avión de pasajeros de fuselaje estrecho fabricado por Airbus. Es una versión más pequeña de la familia A320, diseñada para rutas de corto y medio alcance. Con capacidad para transportar entre 124 y 156 pasajeros, el A319 ofrece una opción compacta y eficiente para aerolíneas y operadores regionales. A pesar de su tamaño reducido, conserva las características de comodidad y versatilidad de la familia A320, con una cabina moderna y sistemas de entretenimiento a bordo. Gracias a su diseño aerodinámico y motores eficientes, el A319 logra un menor consumo de combustible y emisiones reducidas.

Figura 44. Pantalla Mostrar Listado Tipo Aviones

El administrador de la Intranet dispone del permiso para revisar qué tipos de avión están disponibles en la flota de la compañía, pudiendo añadir nuevos a la intranet (1), si estos se añaden a la flota de la compañía.



ID Usuario	Nombre	Apellidos	Email	Teléfono	Blockchain Account	DNI	Activo	Cambiar estado
333	Mario	C T	mario@gmail.com	622622622	0x85e9b26b1f28842852CAA3bd31C5bD467F32495	28202496B	false	Cambiar estado activo/inactivo
111	Alba	L R	alba@gmail.com	655677688	0xe91C87193EF87859ed153480CD22B4e7bDC6F1	02377464T	true	Cambiar estado activo/inactivo
777	Angelés	S d A	angelés@gmail.com	666555444	0xDdE8B8a7053A55b54b8061D9463B00896059832C	15886816A	true	Cambiar estado activo/inactivo
888	Gonzalo	B A	gonzalo@gmail.com	600632245	0x9326f2FCE2CF29C0D025FD522CfccBeE471A83b6	95524083T	true	Cambiar estado activo/inactivo
444	Julia	S d A	julia@gmail.com	655655655	0xD460Fa8DcB204b08875625c8945bE0B250aD48D	79526691M	true	Cambiar estado activo/inactivo
555	Roberto	M J	roberto@gmail.com	677677677	0xF6F4BBe0b279c659F10A0E66D2fF4a0b781b3d78	89348988Z	false	Cambiar estado activo/inactivo
666	Claudia	B G	claudia@gmail.com	655655655	0x1e8c6233005226E5B611072D103304FD3E761368	40163123V	false	Cambiar estado activo/inactivo
admin	Eva	Manrique Sanz	eva.manriquesz@gmail.com	630720672	0xB256415428e9c3406169f13586ED2514fB6E8E67	060281775	true	Cambiar estado activo/inactivo
222	Blanca	M S	blanca@gmail.com	611611611	0xa699C110C17b10A258730503C647e5049457edB6	47529662R	false	Cambiar estado activo/inactivo

Figura 45. Pantalla Mostrar Listado Usuarios

Así como el usuario puede añadir nuevos tipos de aviones, también puede ver y gestionar la plantilla de empleados.

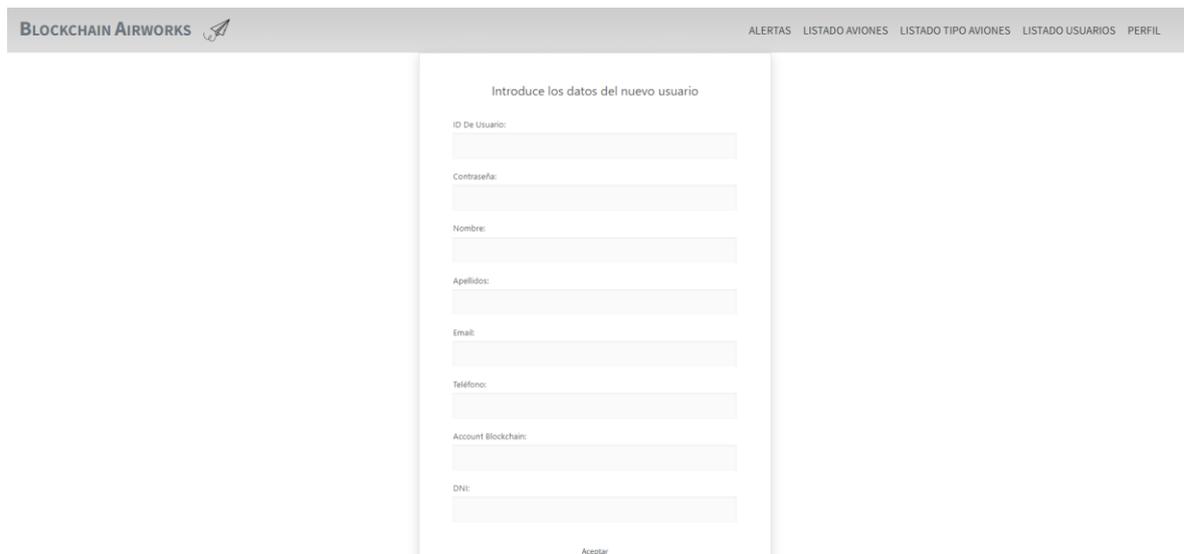


Figura 46. Pantalla de Introducir Nuevo Usuario

Como se ha mencionado en el subpartado anterior, el administrador puede añadir nuevos usuarios a la página web, y deberá incluir la siguiente información, como si el usuario se auto registrase.

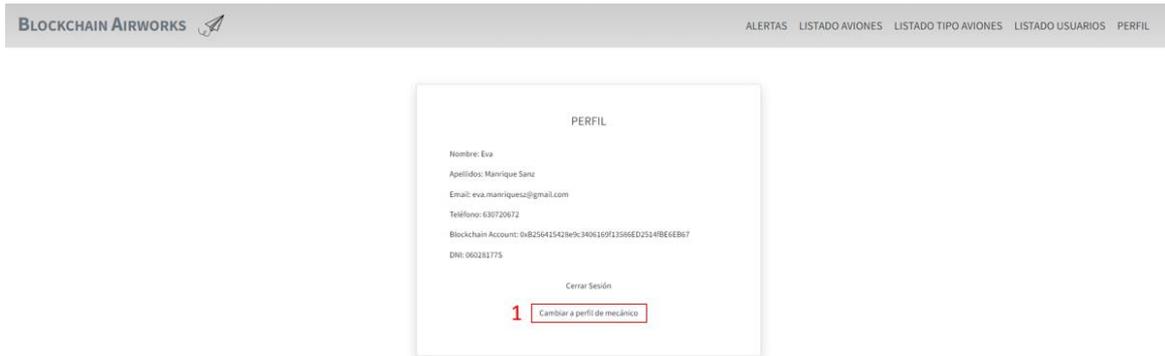


Figura 47. Pantalla Perfil de Administrador

El perfil del administrador es similar al del mecánico, con la salvedad de que el administrador puede cambiar de perfil para simular que es un mecánico (1).

En conclusión, se han logrado todos los objetivos marcados al inicio del proyecto, ofreciendo un producto completamente funcional.

Capítulo 7. TRABAJOS FUTUROS Y POSIBLES

ÁREAS DE MEJORA

Como se refleja en el proyecto, la tecnología Blockchain abre un gran abanico de posibilidades respecto a la forma de actuar y a las posibles nuevas funcionalidades. Cada año que pasa, esta tecnología crece más y más, ofreciendo un sinfín de posibilidades.

En el ámbito de la aviación, el uso de Blockchain no es popular en la actualidad. De hecho, como se explica en el estado de la cuestión, la digitalización de las aerolíneas es lenta: se han realizado multitud de cambios en los últimos años, pero aún queda mucho camino por recorrer.

Por otro lado, para avanzar en la implementación de esta tecnología es necesario formar a los responsables de manera que tengan una ligera idea de lo que ocurre con información, lo que supone una inversión económica mucho mayor, que muchas empresas son reticentes a realizar. En la misma línea, la tecnología Blockchain evoluciona a pasos agigantados, por lo que también sería necesario mantener al día a los administradores en este caso, para que puedan mantener su sistema al día.

Respecto al sistema desarrollado, se podría seguir mejorando y añadiéndole funcionalidad para introducirlo al mercado con una mayor calidad. Entre estas mejoras o adiciones se encuentran:

- Mejora de los flujos de inicio de sesión de manera que se alineen con los estándares de OAuth 2.0. [66]
- Añadir la funcionalidad de comprobar la revisión directamente desde el listado, autocompletando los campos en la pantalla comprobarInforme.

- Estudiar el despliegue y utilización de otras redes Blockchain, puesto que Ethereum tiene un coste mayor que otras, así como tiempos de respuesta más largos.
- Monitorizar los componentes en tiempo real, integrando IoT, haciendo uso de sensores de vuelo y otros sistemas de monitoreo, de manera que las predicciones se actualicen de forma dinámica.

A modo de resumen, Blockchain ofrece un sinfín de posibilidades para la industria, pero, sin embargo, no están excesivamente estudiadas. En el momento que esto ocurra, supondrá una revolución para todos los sectores, incluido el de la aviación.

Capítulo 8. BIBLIOGRAFÍA

- [1] *Apty - Aviation Digital Transformation*. (7 de Enero de 2023). Obtenido de <https://www.apty.io/blog/aviation-digital-transformation/>
- [2] *IBM - Blockchain Technology*. (2022). Obtenido de <https://www.ibm.com/topics/blockchain>
- [3] *IBM - Machine Learning*. (2022). Obtenido de <https://www.ibm.com/topics/machine-learning>
- [4] Nakamoto, S. (2008). *Bitcoin: A peer-to-peer cash system*. Obtenido de <https://bitcoin.org/bitcoin.pdf>
- [5] Buterin, V. (2013). *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*. Obtenido de <https://ethereum.org/en/whitepaper/>
- [6] *Ethereum - Ethereum Virtual Machine (EVM)*. (18 de Marzo de 2023). Obtenido de <https://ethereum.org/en/developers/docs/evm/>
- [7] *Solidity Examples*. (2023). Obtenido de <https://docs.soliditylang.org/en/v0.8.7/>
- [8] *Solidity*. (10 de Mayo de 2023). Obtenido de <https://soliditylang.org/>
- [9] *Vyper Lang*. (2023). Obtenido de <https://docs.vyperlang.org/en/stable/>
- [10] *GitHib - Vyper*. (Marzo de 2023). Obtenido de <https://github.com/vyperlang/vyper>
- [11] *Rust Lang*. (2023). Obtenido de <https://www.rust-lang.org/>
- [12] *Doc Rust Lang - Hello World*. (2022). Obtenido de <https://doc.rust-lang.org/book/ch01-02-hello-world.html>
- [13] *Fe Lang*. (2023). Obtenido de <https://fe-lang.org/>

- [14] *Java*. (2023). Obtenido de <https://www.java.com/es/>
- [15] *Python*. (2023). Obtenido de <https://www.python.org/>
- [16] *Node.js*. (2023). Obtenido de <https://nodejs.org/en>
- [17] *PHP*. (2023). Obtenido de <https://www.php.net/>
- [18] *PostgreSQL*. (2023). Obtenido de <https://www.postgresql.org/>
- [19] *PgAdmin*. (2023). Obtenido de <https://www.pgadmin.org/>
- [20] *W3 Standards*. (2023). Obtenido de <https://www.w3.org/standards/>
- [21] *Mozilla Developer - HTML*. (2023). Obtenido de <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [22] *W3Schools - CSS*. (2022). Obtenido de <https://www.w3schools.com/css/>
- [23] *W3Schools - JavaScript*. (2022). Obtenido de <https://www.w3schools.com/js/>
- [24] Sebastian Raschka, V. M. (2019). *Machine Learning in Python: Essential Techniques*. Marcombo.
- [25] Seabold, S. a. (2010). *Statsmodels: Econometric and statistical modeling with python*. Obtenido de <https://www.statsmodels.org/stable/index.html>
- [26] *Scikit Learn*. (2022). Obtenido de <https://scikit-learn.org/stable/>
- [27] *VS Code Learn*. (2022). Obtenido de <https://code.visualstudio.com/learn>
- [28] *Remix Ethereum*. (2022). Obtenido de <https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=nul1&version=soljson-v0.8.18+commit.87f61d96.js>
- [29] *TruffleSuite - Ganache*. (2023). Obtenido de <https://trufflesuite.com/ganache/>
-

- [30] *IBM Supply Chain Intelligence Suite: Food Trust*. (2022). Obtenido de <https://www.ibm.com/products/supply-chain-intelligence-suite/food-trust>
- [31] *Factom PRO*. (2019). Obtenido de <https://factom.pro/>
- [32] *Vietnam Blockchain Components*. (2019). Obtenido de <https://vietnamblockchain.asia/post/5666316/5-basic-components-of-blockchain>
- [33] Molina-Jimenez, C. (2018). *Implementation of Smart Contracts Using Hybrid Architectures with On and Off-Blockchain Components*. París: 2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2).
- [34] D. Mingxiao, M. X. (2017). *A review on consensus algorithm of blockchain*. IEEE International Conference on Systems, Man, and Cybernetics (SMC).
- [35] Illinois, U. o. (Marzo de 2023). *Temario Asignatura Computer Security*.
- [36] W. Dai, T. P. (2012). *Signing, Enhancing Data Trustworthiness via Assured Digital*. IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 6, pp. 838-851,.
- [37] *IBM - Funcion Hash*. (2022). Obtenido de <https://www.ibm.com/docs/es/psfa/7.1.0?topic=reference-hash-function>
- [38] *Iberdrola - Machine Learning*. (2022). Obtenido de <https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico>
- [39] *Machine Learning Steps: A complete guide*. (16 de Febrero de 2023). Obtenido de <https://www.simplilearn.com/tutorials/machine-learning-tutorial/>
- [40] Nasteski, V. (2017). *An overview of the supervised machine learning methods*. Horizons.

- [41] *Aprende Machine Learning - Regresión Lineal*. (2020). Obtenido de <https://www.aprendemachinlearning.com/tag/regresion-lineal/>
- [42] Bonaccorso, G. (2017). *Machine Learning Algorithms*. Birmingham: Packt.
- [43] ICAO - *Organización de Aviación Civil Internacional (International Civil Aviation Organization)*. (2023). Obtenido de <https://www.icao.int/>
- [44] EASA - *Agencia Europea de Seguridad Aérea (European Union Aviation Safety Agency)*. (2023). Obtenido de <https://www.easa.europa.eu/en>
- [45] FAA - *Administración Federal de Aviación (Federal Aviation Administration)*. (2023). Obtenido de <https://www.faa.gov/>
- [46] ICAO. (1994). *Convention on International Civil Aviation*.
- [47] ISO 9001 Standard. <https://www.iso.org/standard/62085.html>.
- [48] ISO 9000 Tips. (2023). Obtenido de <https://the9000store.com/articles/iso-9000-tips-document-control-requirements/>
- [49] FAA - *Federal Aviation Administration Licenses and Certificates*. (Enero de 2023). Obtenido de https://www.faa.gov/aircraft/air_cert/airworthiness_certification
- [50] EASA *Regulation Standards*. (2023). Obtenido de <https://www.easa.europa.eu/regulations-standards/airworthiness-certification>
- [51] *Airline Suppliers*. (2023). Obtenido de <https://www.airlinesuppliers.com/article/applying-iso-9001-in-the-aviation-industry>
- [52] ISO 9001 Requirements - *Evaluating Performance*. (2023). Obtenido de <https://the9000store.com/iso-9001-2015-requirements/iso-9001-2015-evaluating-performance/>

- [53] *Aviation Group - Maintenance Types.* (2023). Obtenido de <https://www.aviationgroup.es/actualidad/mantenimiento-de-aeronaves-clave-para-la-seguridad-en-los-vuelos>
- [54] *Grupo iberia.* (20 de Diciembre de 2020). Obtenido de <https://grupo.iberia.es/pressrelease/details/12704>
- [55] *Statista - Number of worldwide air traffic fatalities from 2006 to 2021.* (Febrero de 2023). Obtenido de <https://www.statista.com/statistics/263443/worldwide-air-traffic-fatalities/>
- [56] *Aviation Safety Networks.* (2023). Obtenido de <http://aviation-safety.net/statistics/period/stats.php?cat=A1>
- [57] *The Doan Law Firm - How Mechanical Failures Can Cause Aviation Accidents.* (2023). Obtenido de <https://www.thedoanlawfirm.com/aviation-accidents/aviation-accidents-caused-by-mechanical-failure/how-mechanical-failures-can-cause-aviation-accid/#:~:text=The%20Federal%20Aviation%20Administration%20maintains,result%20in%20catastrophic%20aviation%20accid>
- [58] Tena, M. (20 de Noviembre de 2018). *BBVA - Metodología AGILE.* Obtenido de <https://www.bbva.com/es/innovacion/metodologia-agile-la-revolucion-las-formas-trabajo/>
- [59] A. Srivastava, S. B. (2017). *SCRUM model for agile methodology.* Noida, India: International Conference on Computing, Communication and Automation (ICCCA).
- [60] *HP Laptop 14s-dq1021ns.* (2023). Obtenido de <https://www.gizcomputer.com/hp-laptop-14s-dq1021ns-comprar/>

- [61] *HP Monitor 27f.* (2023). Obtenido de <https://support.hp.com/es-es/document/c06435426>
- [62] *Apple - iPad Air.* (2023). Obtenido de <https://www.apple.com/ipad-air/>
- [63] *Talent - Salario progradores.* (2023). Obtenido de <https://es.talent.com/salary?job=programador>
- [64] *Aircraft Cost Calculator.* (2023). Obtenido de <https://www.aircraftcostcalculator.com/>
- [65] Atilano Fernandez-Pacheco Sanchez-Migallón, U. P. (s.f.). *Apuntes Requisitos Funcionales y No Funcionales Ingeniería del Software.*
- [66] *Oauth 2.0.* (s.f.). Obtenido de [hNps://oauth.net/2/](https://oauth.net/2/)
- [67] *ODS - Objetivos de Desarrollo Sostenible Agenda 2030.* (2015). Obtenido de <https://www.un.org/sustainabledevelopment/es/2015/09/la-asamblea-general-adopta-la-agenda-2030-para-el-desarrollo-sostenible/>

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

Los Objetivos de Desarrollo Sostenible (ODS) son una serie de 17 objetivos interconectados aprobados por las Naciones Unidas en el año 2015 para lograr un futuro sostenible para todos. Estos objetivos abarcan una amplia gama de temas, desde la erradicación de la pobreza y el hambre, hasta la acción climática y la promoción de la igualdad de género; llamando a todos los países a aunar fuerzas y trabajar juntos para construir un mundo más justo, equitativo y sostenible para generaciones presentes y futuras. Son los que se muestran en la Figura.



Figura 48. Objetivos de Desarrollo Sostenible (ODS). Fuente: ONU

El desarrollo de este Trabajo de Fin de Grado se alinea con los siguientes objetivos:

- **Trabajo decente y crecimiento económico (número 8):** Entre las metas marcadas por el octavo Objetivo de Desarrollo Sostenible, se incluye lograr niveles más

elevados de productividad económica mediante la modernización, y promover políticas orientadas a fomentar la creatividad o la innovación. Con este proyecto, se renovará considerablemente el modelo de trabajo de las aerolíneas, acercándolas a la tecnología actual, y permitiendo a los jóvenes que accedan a ellas a trabajar, rodearse de un entorno de actualidad, y adquirir nuevas habilidades y conocimientos.

Asimismo, al utilizar tecnología Blockchain y Machine Learning para mejorar el mantenimiento de los aviones, se aumenta la eficiencia en el trabajo de los mecánicos y se reducen los tiempos de inactividad de los aviones, lo que a su vez aumenta la productividad y la rentabilidad de las aerolíneas.

- **Industria, innovación e infraestructura (número 9):** Este objetivo promueve desarrollar infraestructuras sostenibles, de calidad, y accesibles, sobre todo por países en desarrollo. La tecnología que se permite implantar con este proyecto podrá ser acogida en cualquier aeropuerto del mundo sin generar residuos masivos derivados de ella, ya que únicamente serán necesarios ordenadores; y será completamente disruptiva porque no será necesario utilizar nada de papel para gestionarse.

Además, la implementación de la tecnología blockchain en el mantenimiento de aviones también puede mejorar la eficiencia de la industria aérea al reducir los tiempos de inactividad de los aviones debido a reparaciones innecesarias o inspecciones de seguridad prolongadas. Al permitir que los registros de mantenimiento se almacenen de manera segura en una base de datos descentralizada y resistente a la manipulación, la información puede ser accesible para las partes interesadas relevantes de manera oportuna y precisa, lo que puede llevar a una toma de decisiones más rápida y precisa, y a una reducción de costos en la industria aérea en general. Además, el proyecto puede fomentar la innovación en la industria aérea al introducir nuevas tecnologías y soluciones a los problemas existentes.

- **Producción y consumo responsables (número 12):** En línea con el objetivo número 9, este decimosegundo objetivo persigue de nuevo la sostenibilidad en los

procedimientos, tal y como ocurrirá con la irrupción del Blockchain y la Inteligencia Artificial en el mantenimiento de aviones, ya que todas las ordenes de revisión se almacenarán, gestionarán y enviarán de manera telemática, eliminando las montañas de papeles en todos los aeropuertos del mundo, evitando así la tala de millones de árboles.

Además, la tecnología de inteligencia artificial utilizada en el proyecto puede mejorar la eficiencia y la precisión del mantenimiento, lo que puede reducir la necesidad de reemplazar piezas o equipos innecesariamente, lo que a su vez reduce la cantidad de residuos generados por la industria de la aviación.

- **Acción por el clima (número 13):** Incorporar medidas relacionadas con la sensibilización por el entorno y el cambio climático, es una de las metas más importantes de este Objetivo de Desarrollo Sostenible. Con este tipo de gestión de las revisiones que se realizará, se pretende detectar altos niveles de contaminación de manera precoz, para evitar dañar el medio ambiente, así como piezas dañadas que puedan provocar desastres ecológicos como ha ocurrido previamente con accidentes, ya sea de aviones o de otros medios de transporte.

Además, el uso de tecnología blockchain en la aviación también puede ayudar a reducir las emisiones de gases de efecto invernadero al mejorar la eficiencia de los procesos de mantenimiento. Al permitir un seguimiento más preciso de la vida útil de las piezas y componentes, se puede reducir la cantidad de piezas de repuesto que se fabrican y envían, lo que a su vez reduce la huella de carbono de la industria de la aviación. También puede ayudar a identificar áreas donde se pueden hacer mejoras en la eficiencia de combustible de los aviones, lo que reduciría aún más las emisiones de gases de efecto invernadero.

ANEXO II: MANUAL DE INSTALACIÓN

A continuación, se detallan las aplicaciones necesarias, cómo instalarlas y cómo configurar el sistema para el correcto funcionamiento de la aplicación.

En primer lugar, se requiere la instalación de los siguientes programas:

- Ganache
- Remix IDE (instalado en VS Code)
- MetaMask
- PgAdmin (opcional, para gestión de bases de datos)

ANEXO II.1 GANACHE

Para el correcto funcionamiento de Ganache se deben seguir los siguientes pasos:

- En primer lugar, se debe crear el espacio de trabajo, usando QUICKSTART

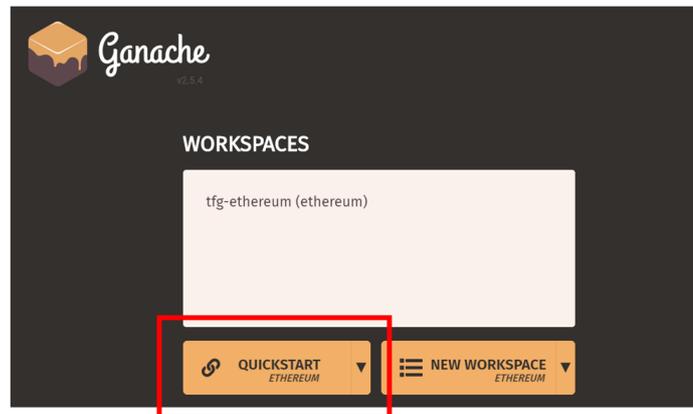
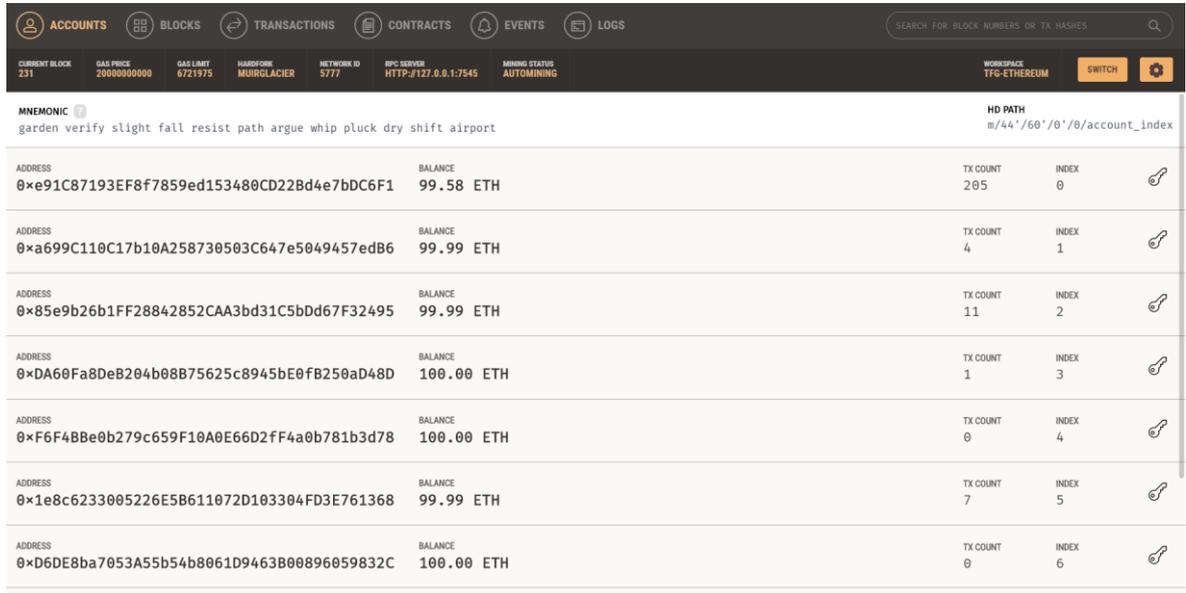


Figura 49. Paso 1 Configuración Ganache

- Una vez dentro, se muestra la información de las cuentas asociadas al workspace, junto a su balance. Si se pulsa en la llave, se muestra la clave privada, utilizada para añadir la cuenta a MetaMask posteriormente.



ADDRESS	BALANCE	TX COUNT	INDEX
0xe91C87193EF8f7859ed153480CD228d4e7bDC6F1	99.58 ETH	205	0
0xa699C110C17b10A258730503C647e5049457edB6	99.99 ETH	4	1
0x85e9b26b1FF28842852CAA3bd31C5bDd67F32495	99.99 ETH	11	2
0xDA60Fa8DeB204b08B75625c8945bE0fB250aD48D	100.00 ETH	1	3
0xF6F4BBe0b279c659F10A0E66D2fF4a0b781b3d78	100.00 ETH	0	4
0x1e8c6233005226E5B611072D103304FD3E761368	99.99 ETH	7	5
0xD6DE8ba7053A55b54b8061D9463B00896059832C	100.00 ETH	0	6

Figura 50. Paso 2 Configuración Ganache

Con estos pasos, se habría desplegado Ganache de manera correcta.

ANEXO II.2 REMIX IDE

En este apartado se explica como desplegar el contrato inteligente en la red de Ethereum que se ha creado con Ganache. Se usará la extensión de REMIX IDE en VS Code. Los pasos que seguir son los siguientes:

- Abrir la extensión de Remix en VS Code (1).
- Elegir la versión de compilador, acorde con la que esté en el contrato (2).
- Pulsar en Run & Deploy para comenzar con el despliegue (3).
- Activar el despliegue (4)

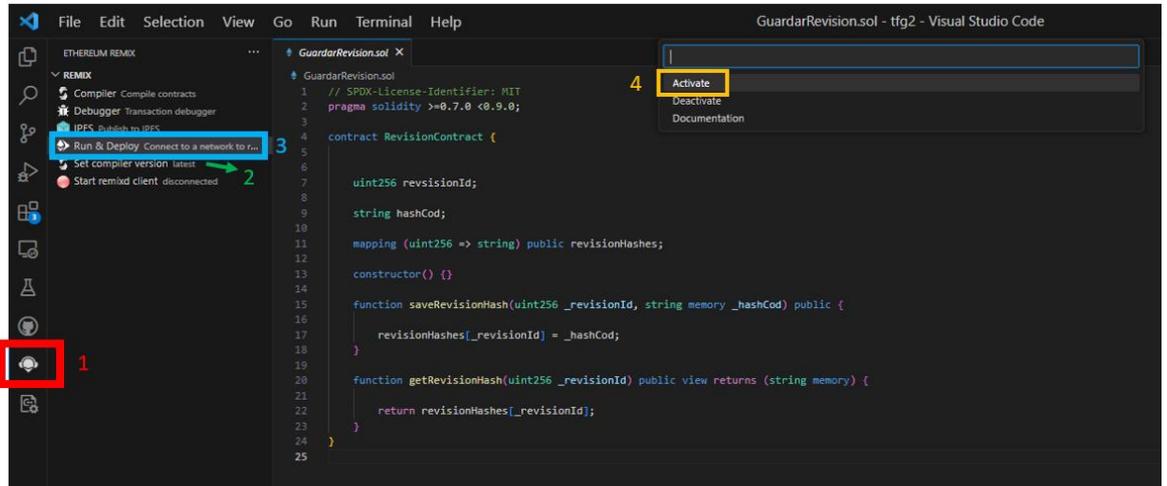


Figura 51. Abrir Remix IDE en VS Code

- Dentro de Remix IDE, debemos asegurarnos de que la dirección sea la misma que en Ganache para conectar Remix a la red de Ethereum que hemos creado.

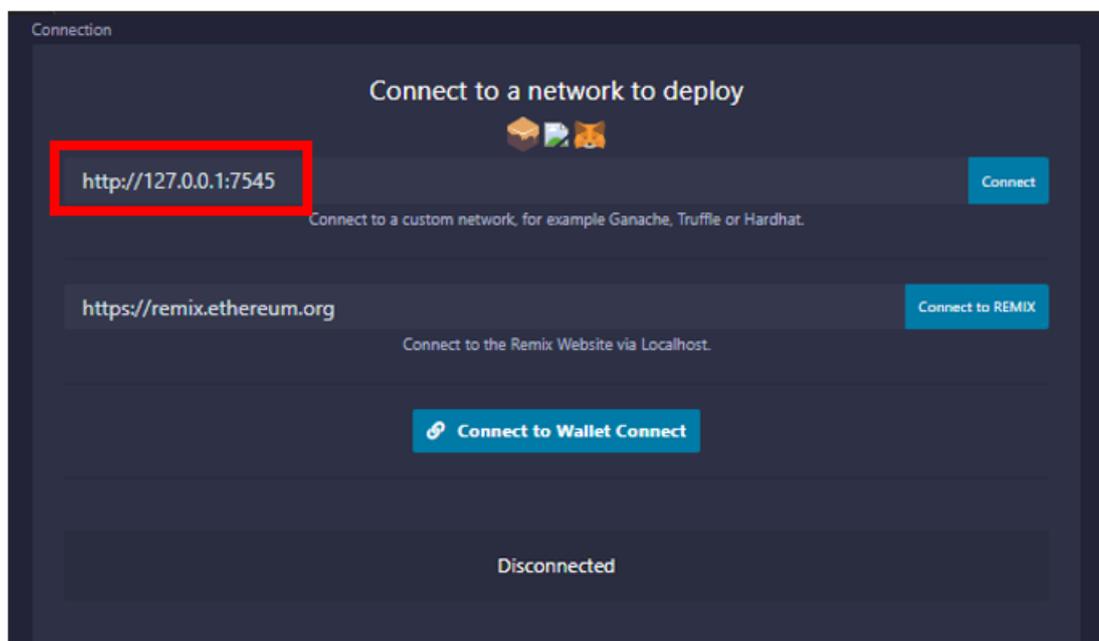


Figura 52. Cómo conectar Remix con Ethereum

- Una vez conectados, deberemos compilar el contrato pulsando el botón 'Compile'. Si el campo para elegir el contrato aparece vacío, se deberá abrir el contrato en una nueva pestaña aparte, de manera que Remix posteriormente lo reconozca.

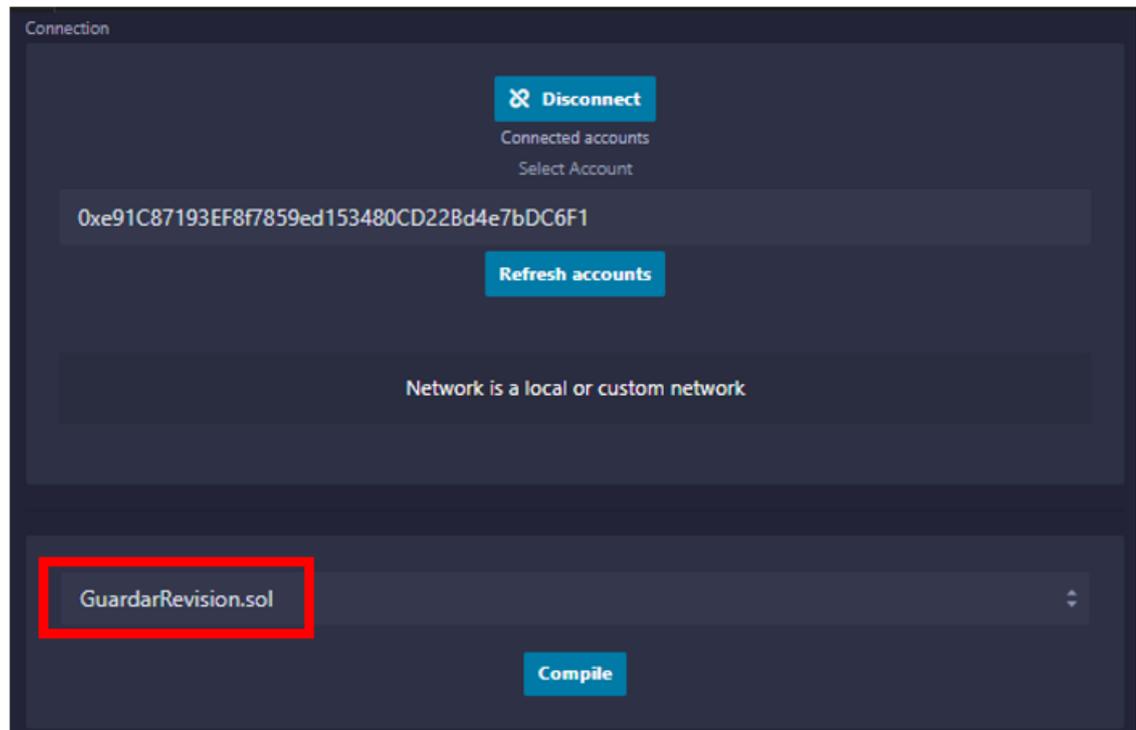


Figura 53. Cómo compilar contrato en Remix

- El siguiente paso será desplegar el contrato (1). Para ello, debemos seleccionar el contrato y pulsar sobre 'Deploy' (2).
- Una vez desplegado, se deberá copiar el ABI (3) y la dirección del contrato desplegado (4) en el HTML destinado a guardar la información en Blockchain.

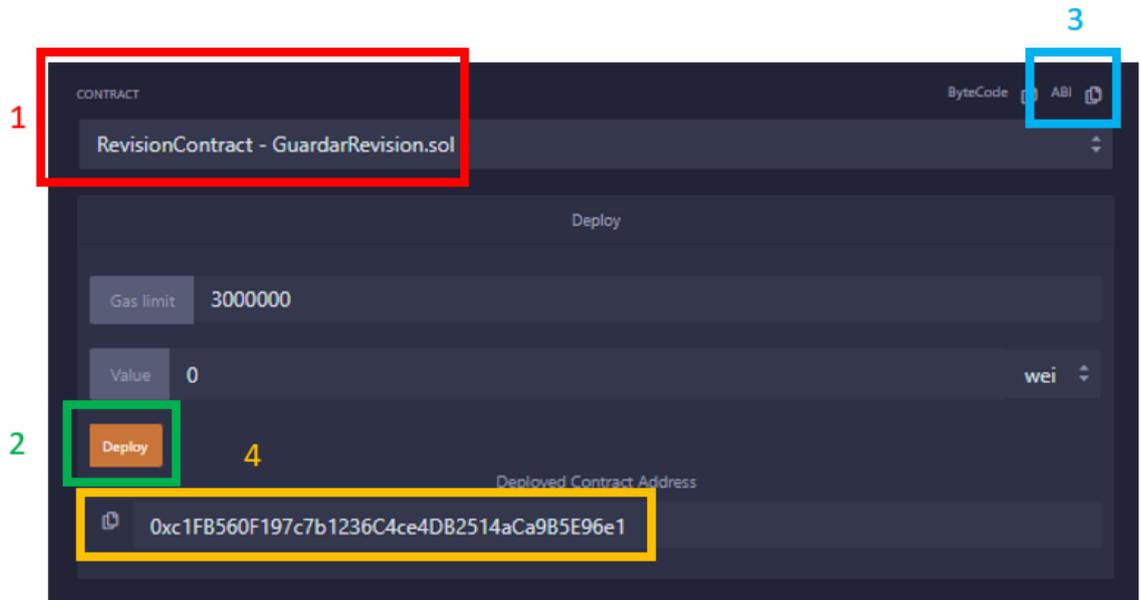


Figura 54. Despliegue del contrato en Remix

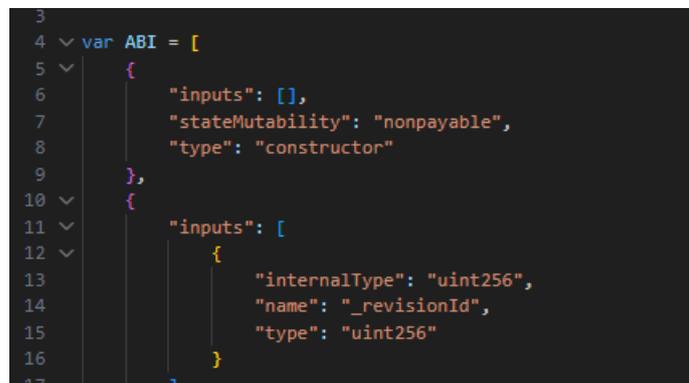


Figura 55. Lugar del ABI copiado

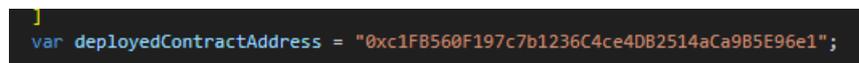


Figura 56. Lugar de la dirección del contrato desplegado.

ANEXO II.3 METAMASK

Para el correcto funcionamiento de Blockchain, es necesario elegir la cuenta asociada al usuario que va a utilizar la aplicación.

En el caso de querer crear un nuevo usuario en la Intranet, también se debe añadir una nueva cuenta, pulsando sobre Importar cuenta y pegando de Ganache la clave privada de la cuenta. (1)

En el caso de querer crear un nuevo usuario en la Intranet, también se debe añadir una nueva cuenta, pulsando sobre Importar cuenta y pegando de Ganache la clave privada de la cuenta. (2)

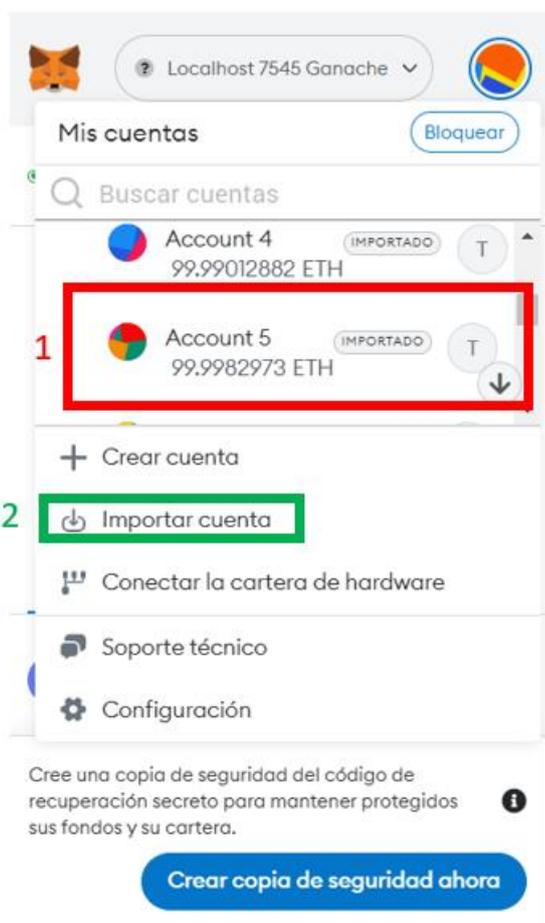


Figura 57. Elegir cuenta Blockchain e Importar Cuenta

Importar cuenta

Las cuentas importadas no se asociarán con la frase secreta de recuperación de la cuenta original de MetaMask. Más información sobre las cuentas importadas [aquí](#)

Seleccionar tipo

Clave privada ▼

Pegue aquí la cadena de clave privada:

Cancelar

Importar

Figura 58. Pantalla Importar Cuenta

Tras ejecutar estos pasos, ya se puede acceder a la web a través de la dirección <http://localhost:8080/>

ANEXO III: MANUAL DE USO

A lo largo de este capítulo se explicará el funcionamiento al detalle de la página web. Se mostrarán página a página, todos los elementos, empezando por lo que ve un usuario sin autenticar, posteriormente el contenido en el perfil de mecánico, y terminando con el contenido del perfil de administrador.

ANEXO III.1 USUARIO SIN AUTENTICAR

ANEXO III.1. 1 INICIO SESIÓN

La primera página que el usuario verá cuando acceda a la web, será la de inicio de sesión, puesto que no puede acceder a ningún otro lugar sin autenticar. Si dispone de cuenta (1), deberá introducir su ID de usuario y contraseña para poder acceder. En cambio, si es un nuevo usuario, deberá pulsar directamente sobre el botón de Registrarse (2), que le redirigirá a una página en la que rellenar su información.



Figura 59. Pantalla de Inicio de Sesión

ANEXO III.1. 2 REGISTRO

Si, por el contrario, el usuario no tiene cuenta creada, y ha pulsado la segunda opción en la pantalla de inicio de sesión; se le redirigirá hasta aquí. En esta página, el usuario debe

introducir toda su información, incluidos su DNI y su cuenta de Blockchain válida y conectada con MetaMask.

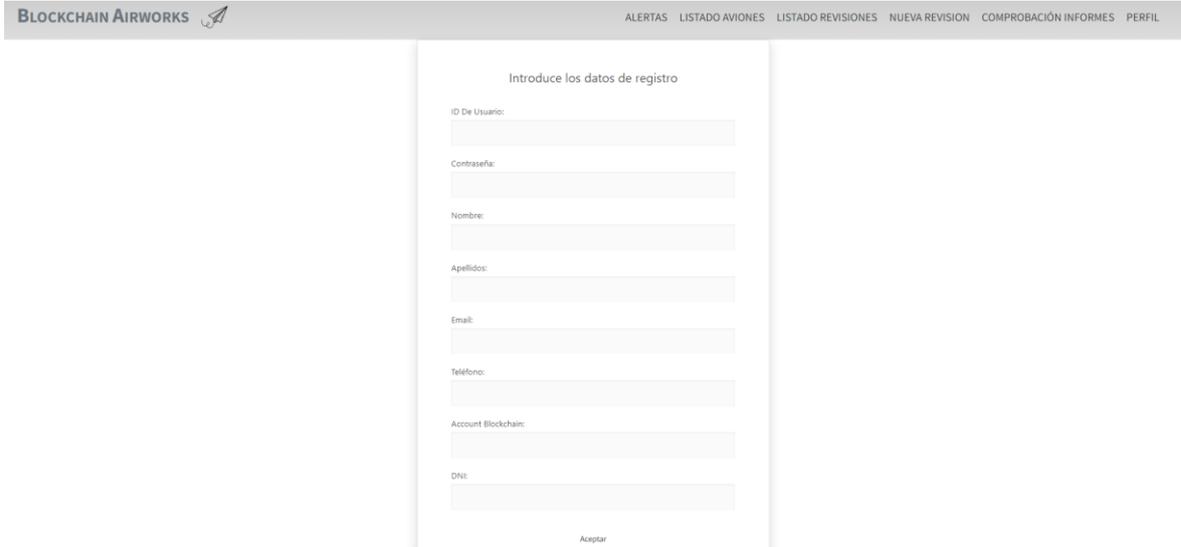


Figura 60. Pantalla de Registro

ANEXO III.2 USUARIO MECÁNICO

Una vez el usuario haya iniciado sesión o se haya creado un perfil, podrá navegar entre las pantallas de: alertas, listado de aviones, listado de revisiones, pantalla para nueva revisión, pantalla para comprobar la veracidad de los informes y su perfil.

ANEXO III.2.1 ALERTAS

En esta pantalla se mostrarán todas las revisiones que están previstas para el día de hoy, así como las que están previstas para la próxima semana. Desde esta pantalla, el usuario podrá ir directamente a iniciar la revisión, o mostrar la información de los aviones. Según se vayan completando las revisiones, desaparecerán de esta pantalla tipo muro.

BLOCKCHAIN AIRWORKS 

ALERTAS LISTADO AVIONES LISTADO REVISIONES NUEVA REVISION COMPROBACIÓN INFORMES PERFIL

ALERTAS

REVISIONES PROGRAMADAS PARA HOY

32111
A321

Número elementos a revisar: 3

[Iniciar revisión](#) [Mostrar info](#)

REVISIONES PROGRAMADAS PARA LA PRÓXIMA SEMANA

<p>31903 A319</p> <p>Número elementos a revisar: 3</p> <p>Iniciar revisión Mostrar info</p>	<p>31912 A319</p> <p>Revisión general</p> <p>Iniciar revisión Mostrar info</p>	<p>32010 A320</p> <p>Número elementos a revisar: 1</p> <p>Iniciar revisión Mostrar info</p>
<p>32010011 A320NEO</p> <p>Número elementos a revisar: 2</p> <p>Iniciar revisión Mostrar info</p>	<p>32010016 A320NEO</p> <p>Número elementos a revisar: 5</p> <p>Iniciar revisión Mostrar info</p>	<p>32012 A320</p> <p>Revisión general</p> <p>Iniciar revisión Mostrar info</p>
<p>32102 A321</p> <p>Número elementos a revisar: 7</p> <p>Iniciar revisión Mostrar info</p>	<p>32107 A321</p> <p>Número elementos a revisar: 1</p> <p>Iniciar revisión Mostrar info</p>	<p>32109 A321</p> <p>Número elementos a revisar: 6</p> <p>Iniciar revisión Mostrar info</p>

Figura 61. Pantalla de Alertas

ANEXO III.2.2 LISTA AVIONES

En esta pantalla se puede inspeccionar el estado actual de los aviones, las fechas de sus próximas revisiones, etc. Para facilitar la tarea, se muestran en verde aquellos que son aptos para volar y en rojo aquellos que tienen algún inconveniente que les permite funcionar con normalidad. Como funcionalidad extra, en esta pantalla se puede filtrar los aviones que se muestran: se puede ver todo (1), filtrar por modelos (2) o por número de elementos pendientes de revisar en la próxima inspección (3).

BLOCKCHAIN AIRWORKS  ALERTAS LISTADO AVIONES LISTADO REVISIONES NUEVA REVISION COMPROBACIÓN INFORMES PERFIL

LISTADO AVIONES

1 2 3

Numero de serie	Modelo	Fuselaje	Ala izquierda	Ala derecha	Cola	Motor	Tren aterrizaje	Cabina pasajeros	Cabina tripulación técnica	Horas vuelo total	Última revisión	Próxima revisión	Permiso volar	Horas entre revisiones	Numero elementos a revisar
31901	A319	apto	apto	apto	apto	apto	revisar	apto	apto	22	2023-05-12	2023-06-03	false	543	1
31902	A319	revisar	revisar	revisar	revisar	revisar	revisar	revisar	revisar	239	2023-06-22	2023-06-25	false	80	8
31903	A319	apto	apto	revisar	revisar	apto	apto	apto	revisar	171	2023-06-04	2023-06-17	false	320	3
31904	A319	apto	apto	apto	apto	apto	apto	apto	apto	2	2023-05-04	2023-05-25	true	521	0
31905	A319	revisar	apto	apto	apto	revisar	revisar	apto	apto	119	2023-05-23	2023-06-09	false	378	3
31906	A319	revisar	revisar	apto	apto	revisar	revisar	apto	apto	122	2023-04-12	2023-04-17	false	121	4
31907	A319	apto	apto	apto	apto	revisar	revisar	apto	apto	81	2023-06-01	2023-06-14	false	320	2
31908	A319	revisar	revisar	apto	revisar	revisar	revisar	apto	apto	160	2023-05-28	2023-06-08	false	280	5
31909	A319	apto	apto	apto	apto	apto	revisar	apto	apto	41	2023-06-13	2023-06-30	false	429	1
31910	A319	apto	revisar	revisar	revisar	revisar	revisar	revisar	revisar	200	2023-06-18	2023-06-25	false	179	7
31911	A319	apto	revisar	revisar	revisar	revisar	apto	revisar	revisar	166	2023-06-23	2023-06-28	false	143	6
31912	A319	apto	apto	apto	revisar	revisar	revisar	apto	apto	44	2023-06-23	2023-07-08	false	366	3
31913	A319	revisar	apto	apto	apto	revisar	revisar	apto	apto	119	2023-05-21	2023-06-05	false	380	3

Figura 62. Pantalla Mostrar Lista Aviones

ANEXO II.2.3 LISTA REVISIONES

De la misma manera que se pueden ver los aviones, se puede mostrar un listado de las revisiones realizadas, que incluyen su hash almacenado en Blockchain, información varia de la revisión, y el path al fichero, para poder realizar la comprobación de su veracidad.

BLOCKCHAIN AIRWORKS  ALERTAS LISTADO AVIONES LISTADO REVISIONES NUEVA REVISION COMPROBACIÓN INFORMES PERFIL

LISTADO REVISIONES

ID de revisión	Numero de serie	ID del mecánico	Nuevas horas de vuelo	Fecha revisión	Proxima revisión	Permiso volar	Hash	PDF Path	Observaciones
14	35002	111	1	2023-06-23	2023-07-10	false	a434b9113da240857892bbfab39fba5879ee86d48823d7bb873aa4953eac30	C:\Users\evama\OneDrive - Universidad Pontificia Comillas\4\GITT\TFG\src\main\resources\informes\informe14.pdf	Limpiar Ala Izquierda
15	35002	111	2	2023-06-23	2023-07-10	false	8b9687f6e2bb99ae876a3431a01bd7c64b4322936cc11de5cb2d28027f230e	C:\Users\evama\OneDrive - Universidad Pontificia Comillas\4\GITT\TFG\src\main\resources\informes\informe15.pdf	Motor falla
4	35003	666	2	2023-06-26	2023-07-11	false	a8b4ed95267e688c722e1f05ae4ac9ceb1287e8ae84848923d67dd9f14d3af	C:\Users\evama\OneDrive - Universidad Pontificia Comillas\4\GITT\TFG\src\main\resources\informes\informe4.pdf	Revisar ruedas del tren de aterrizaje
1	35001	333	1	2023-06-29	2022-03-10	true	6370fb6e3b32bbc0219f3b47d448af172493af60ced91596a723c6c36aac5f4	C:\Users\evama\OneDrive - Universidad Pontificia Comillas\4\GITT\TFG\src\main\resources\informes\informe1.pdf	Todo OK
2	32013	666	3	2023-06-15	2023-07-05	true	658fbd397ab036423cfc36dea44ac338cea43382d0521c00e1060d9d782e4e19	C:\Users\evama\OneDrive - Universidad Pontificia Comillas\4\GITT\TFG\src\main\resources\informes\informe2.pdf	Todo OK
3	35001	666	3	2023-05-29	2022-02-07	true	a2d46c7fe773528e59a7eab52daee59b91825c8175f8798cda0431efe098c2	C:\Users\evama\OneDrive - Universidad Pontificia Comillas\4\GITT\TFG\src\main\resources\informes\informe3.pdf	Todo OK
5	35003	666	14	2023-07-18	2023-07-31	false	648c6098c61c75721d2c78f564e0ff908e3a064994a64c3da0f2b488cea7b3a7	C:\Users\evama\OneDrive - Universidad Pontificia Comillas\4\GITT\TFG\src\main\resources\informes\informe5.pdf	Limpiar ventanas cabina pasajeros

Figura 63. Pantalla Mostrar Lista Revisiones

ANEXO II.2.4 INTRODUCIR REVISIÓN

En esta pantalla se desarrolla una de las funcionalidades más importantes. El usuario debe introducir la información de la revisión que ha realizado, para que, a continuación, se cree

el PDF, y su correspondiente hash; el cual se almacenará en Blockchain. Los campos de número de serie y número de horas serán inputs (1), el modelo de avión un desplegable entre los disponibles en la base de datos (2), el estado de los elementos serán radio boxes (3) y las observaciones input de nuevo (1). Se mostrará en primer lugar la pantalla con los campos vacíos. Tras completarlos, se deberá pulsar el botón ‘Calcular fecha estimada próxima revisión’ (4).



Figura 64. Pantalla Introducir Revisión Vacía

Tras introducir toda la información, generar la fecha estimada y aceptarla, se creará el informe PDF, que solicitará permiso para acceder a los archivos, para encontrar la ubicación donde almacenarlo:

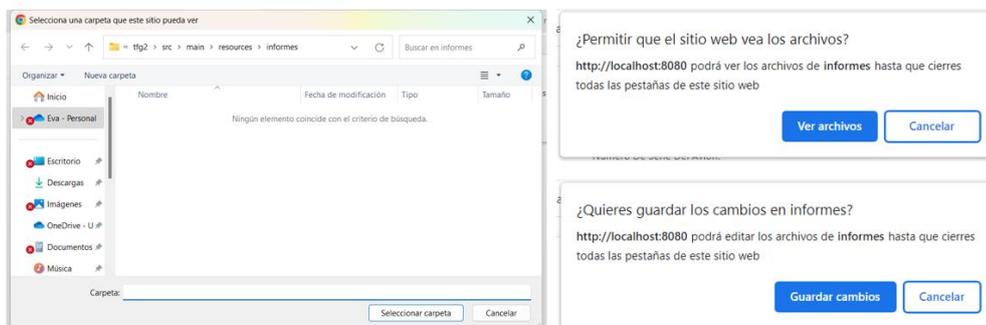


Figura 65. Permisos para almacenar informe PDF

Finalmente, tras aceptar todo lo que se solicita, se mostrará una ventana final con toda la información introducida, la nueva fecha (1) y el hash de la transacción en Blockchain (2), probando que esta transacción efectivamente ha sido realizada.

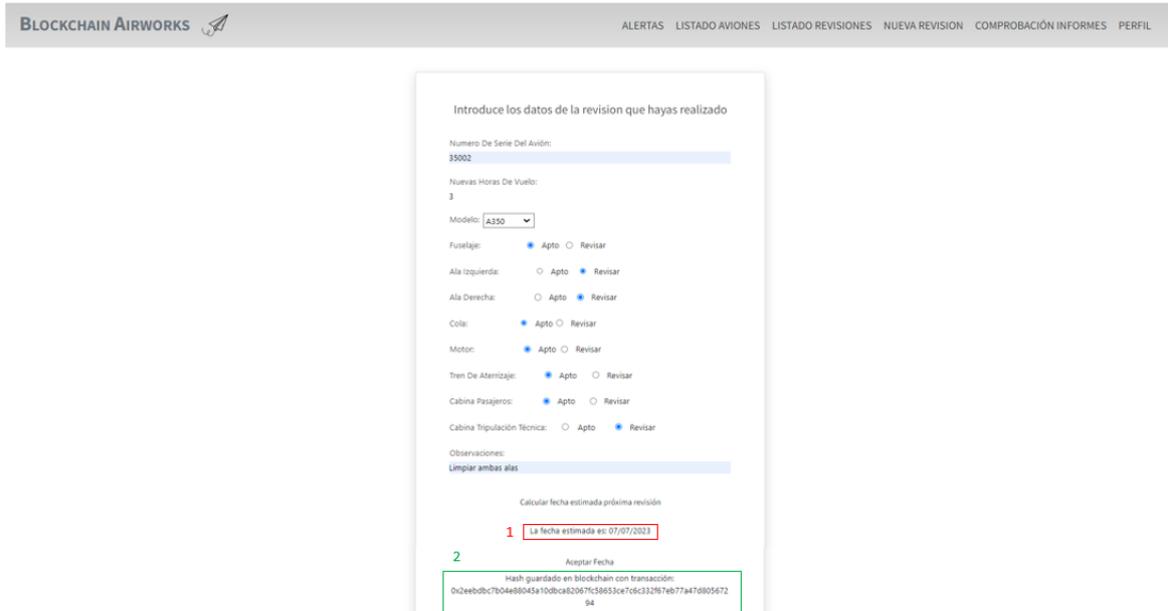


Figura 66. Pantalla Introducir Revisión Parte Final

ANEXO III.2.5 COMPROBAR INFORME

Un segundo pilar de vital importancia para el proyecto es la comprobación de la veracidad de los informes, llevada a cabo introduciendo el número de la revisión que desees comprobar (1), junto al archivo PDF de esa revisión (2). Tras la comprobación, se informará de si todo es correcto o si han ocurrido fallos.



Figura 67. Pantalla Comprobar Informe

ANEXO III.2.6 PERFIL

Por último, todos los usuarios podrán ver su información personal en el apartado perfil, desde donde podrán incluso cerrar su sesión.



Figura 68. Pantalla de Perfil de Usuario

ANEXO III.3. USUARIO ADMINISTRADOR

El usuario que inicia sesión con el ID admin, será considerado administrador de la página web. Entre sus funciones se encontrarán: revisar muro de alertas, ver listado de aviones y de revisiones, ver y gestionar listado de usuarios, ver y gestionar listado de tipos (o modelos) de avión; y, por último, ver su perfil, desde donde podrán simular ser mecánicos.

ANEXO III.3.1 ALERTAS

El muro de alertas es semejante al de la **¡Error! No se encuentra el origen de la referencia.** Únicamente difiere en que el administrador no puede iniciar una revisión desde ahí; simplemente puede mostrar la información de los aviones.

ANEXO III.3.2 LISTADO AVIONES

El listado de aviones disponible para el administrador es semejante al de la **¡Error! No se encuentra el origen de la referencia.**, teniendo a su alcance las mismas opciones de filtrado.

ANEXO III.3.2 LISTADO REVISIONES

De nuevo, el listado de revisiones disponible para el administrador se asemeja al del mecánico, mostrado en la **¡Error! No se encuentra el origen de la referencia.**

ANEXO III.3.3 LISTADO TIPO AVIONES

El administrador de la Intranet dispone del permiso para revisar qué tipos de avión están disponibles en la flota de la compañía, pudiendo añadir nuevos a la intranet (1), si estos se añaden a la flota de la compañía.

BLOCKCHAIN AIRWORKS
ALERTAS LISTADO AVIONES LISTADO TIPO AVIONES LISTADO USUARIOS PERFIL

LISTADO TIPO AVIONES

1 Introducir nuevo tipo de avión

Nombre de modelo	Descripción
A220	El Airbus A220 es un avión de pasajeros de tamaño medio diseñado para vuelos de corto y medio alcance. Ofrece una cabina espaciosa y cómoda, capacidad para entre 100 y 150 pasajeros y tecnologías avanzadas de eficiencia de combustible y reducción de ruido. Es una opción popular para aerolíneas que buscan mejorar su rentabilidad y reducir su impacto ambiental.
A330-300	El A330-300 es un avión de pasajeros de fuselaje ancho fabricado por Airbus. Es conocido por su versatilidad y capacidad de largo alcance. Con capacidad para transportar entre 277 y 440 pasajeros, el A330-300 ofrece un equilibrio entre eficiencia y comodidad. Cuenta con una configuración de cabina espaciosa y moderna, con opciones de asientos flexibles y sistemas de entretenimiento a bordo. Su diseño aerodinámico y eficiente motores le permiten volar largas distancias de manera eficiente y reducir el consumo de combustible.
A330-200	El A330-200 es un avión de pasajeros de fuselaje ancho fabricado por Airbus. Es reconocido por su capacidad de largo alcance y su versatilidad en diversas rutas. Con capacidad para transportar entre 246 y 406 pasajeros, el A330-200 ofrece un equilibrio entre eficiencia y comodidad. Cuenta con una cabina espaciosa y moderna, con opciones de asientos configurables y sistemas de entretenimiento a bordo. Su diseño aerodinámico y eficiente motorización le permite realizar vuelos de larga distancia de manera eficiente y reducir el consumo de combustible.
A321	El A321 es un avión de pasajeros de fuselaje estrecho fabricado por Airbus. Es conocido por su eficiencia y versatilidad en rutas de corto y medio alcance. Con capacidad para transportar entre 185 y 240 pasajeros, el A321 ofrece una opción ideal para aerolíneas que buscan una mayor capacidad en comparación con otros modelos de la familia A320. Cuenta con una cabina moderna y cómoda, con opciones de configuración flexibles y sistemas de entretenimiento a bordo. Gracias a su diseño aerodinámico y motores eficientes, el A321 logra un menor consumo de combustible y emisiones reducidas.
A320	El A320 es un avión de pasajeros de fuselaje estrecho fabricado por Airbus. Es ampliamente reconocido por su eficiencia y versatilidad en rutas de corto y medio alcance. Con capacidad para transportar entre 150 y 180 pasajeros, el A320 ofrece una opción popular para aerolíneas y pasajeros. Cuenta con una cabina moderna y confortable, con sistemas de entretenimiento a bordo y opciones de configuración flexibles. Gracias a su diseño aerodinámico y motores eficientes, el A320 logra un menor consumo de combustible y emisiones reducidas. Es un avión confiable y utilizado por aerolíneas de todo el mundo para vuelos cortos y medios.
A320neo	El A320neo (New Engine Option) es una versión actualizada del avión de pasajeros A320, fabricado por Airbus. Esta variante se caracteriza por contar con motores más eficientes y mejoras aerodinámicas. El A320neo ofrece una mayor eficiencia de combustible y una reducción significativa en las emisiones de CO2 y en la huella de ruido. Además, proporciona una mayor autonomía de vuelo y capacidad de carga. La cabina del A320neo mantiene la comodidad y configuración flexible del modelo original, incluyendo sistemas de entretenimiento actualizados.
A319	El A319 es un avión de pasajeros de fuselaje estrecho fabricado por Airbus. Es una versión más pequeña de la familia A320, diseñada para rutas de corto y medio alcance. Con capacidad para transportar entre 124 y 156 pasajeros, el A319 ofrece una opción compacta y eficiente para aerolíneas y operadores regionales. A pesar de su tamaño reducido, conserva las características de comodidad y versatilidad de la familia A320, con una cabina moderna y sistemas de entretenimiento a bordo. Gracias a su diseño aerodinámico y motores eficientes, el A319 logra un menor consumo de combustible y emisiones reducidas.

Figura 69. Pantalla Mostrar Listado Tipo Aviones

ANEXO III.3.4 LISTADO USUARIOS

Así como el usuario puede añadir nuevos tipos de aviones, también puede ver y gestionar la plantilla de empleados. Desde esta pantalla, puede ver qué empleados se encuentran en activo y cuales no (1), así como gestionar su estado (2), y añadir nuevos empleados a la plantilla, a los cuales registrará en la intranet (3) en la página que se mostrará posteriormente.



BLOCKCHAIN AIRWORKS  ALERTAS LISTADO AVIONES LISTADO TIPO AVIONES LISTADO USUARIOS PERFIL

LISTADO USUARIOS

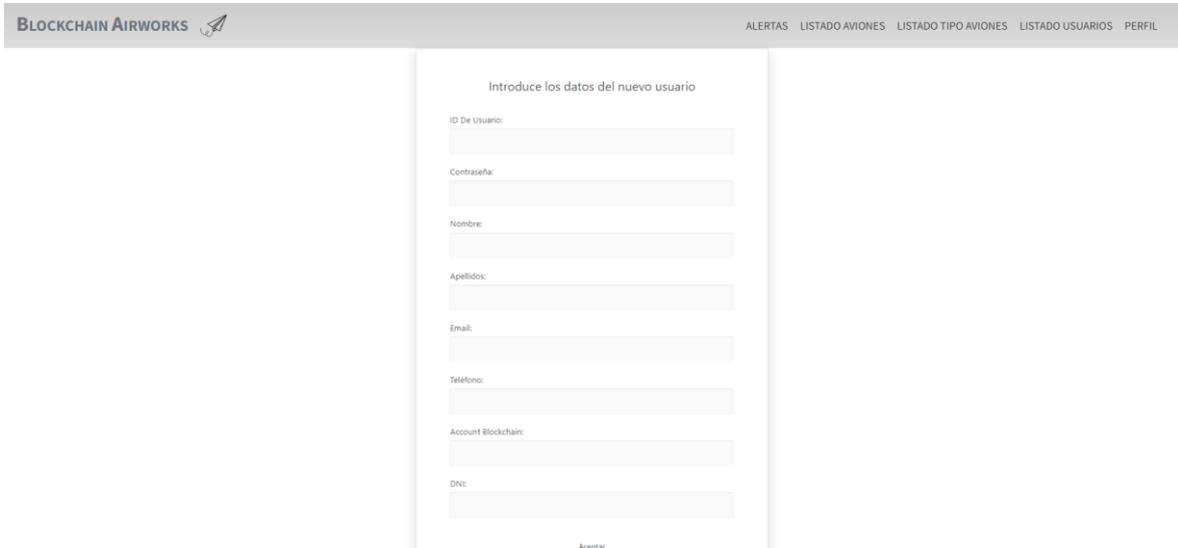
3 Introdúcir nuevo usuario

ID Usuario	Nombre	Apellidos	Email	Teléfono	Blockchain Account	DNI	Activo	Cambiar estado
333	Mario	C T	mario@gmail.com	622622622	0x85e9b26b1f28842852CAA3bd31C5bD467F32495	282024968	false	Cambiar estado activo/inactivo
111	Alba	L R	alba@gmail.com	655677688	0xe91c87193ef87859ed153480CD22Bd4e7bDc6f1	02377464f	true	Cambiar estado activo/inactivo
777	Angeles	S d A	angeles@gmail.com	666555444	0xDd0E8ba7053A55b54b8061D9463B00896059832C	15886816A	true	Cambiar estado activo/inactivo
888	Gonzalo	B A	gonzalo@gmail.com	600632245	0x9326f2FCE2CF29C0D025fD522CfCcBeE471A83b6	955240837	true	Cambiar estado activo/inactivo
444	Julia	S d A	julia@gmail.com	655655655	0xD460fa8Dx8204b08875625c89458E0fB250aD48D	79526691M	true	Cambiar estado activo/inactivo
555	Roberto	M J	roberto@gmail.com	677677677	0xF6f48Bc0b279c659f10A0E66D2fF4a0b781b3d78	893489882	false	Cambiar estado activo/inactivo
666	Claudia	B G	claudia@gmail.com	655655655	0x1e8c6233005226E58611072D103304FD3E761368	40163123V	false	Cambiar estado activo/inactivo
admin	Eva	Manrique Sanz	eva.manriquesz@gmail.com	630720672	0xB256415428e9c3406169f13586ED2514fB6E8667	060281775	true	Cambiar estado activo/inactivo
222	Blanca	M S	blanca@gmail.com	611611611	0xa699C110C17b10A258730503C647e5049457edB6	47529662R	false	Cambiar estado activo/inactivo

Figura 70. Pantalla Mostrar Listado Usuarios

ANEXO III.3.5 INTRODUCIR NUEVOS USUARIOS

Como se ha mencionado en el subapartado anterior, el administrador puede añadir nuevos usuarios a la página web, y deberá incluir la siguiente información, como si el usuario se auto registrase.

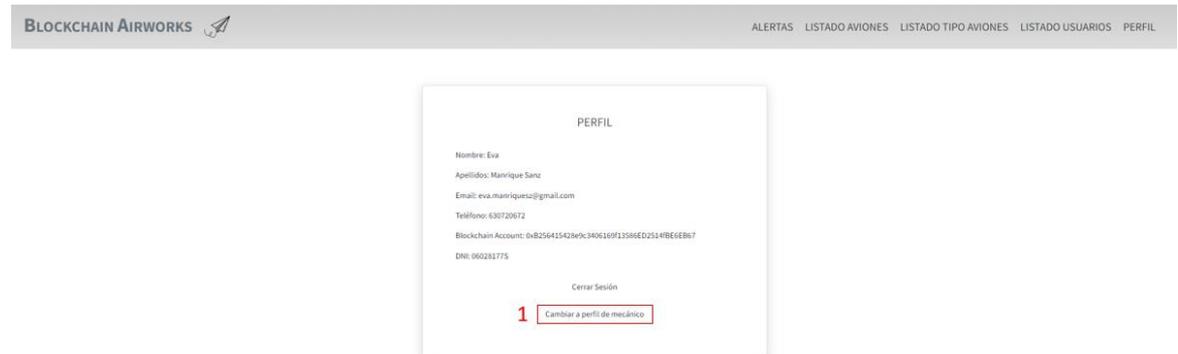


The screenshot shows a web interface for creating a new user. At the top, there is a navigation bar with 'BLOCKCHAIN AIRWORKS' and a paper plane icon on the left, and a list of menu items: 'ALERTAS', 'LISTADO AVIONES', 'LISTADO TIPO AVIONES', 'LISTADO USUARIOS', and 'PERFIL'. The main content area is titled 'Introduce los datos del nuevo usuario' and contains several input fields: 'ID De Usuario:', 'Contraseña:', 'Nombre:', 'Apellidos:', 'Email:', 'Teléfono:', 'Account Blockchain:', and 'DNI:'. Below the fields is an 'Aceptar' button.

Figura 71. Pantalla de Introducir Nuevo Usuario

ANEXO III.3.6 PERFIL

El perfil del administrador es similar al del mecánico, con la salvedad de que el administrador puede cambiar de perfil para simular que es un mecánico (1).



The screenshot shows the 'PERFIL' page for an administrator. The navigation bar is identical to the previous figure. The main content area displays the following user information: 'Nombre: Eva', 'Apellidos: Manrique Sanz', 'Email: eva.manriquez@gmail.com', 'Teléfono: 630720672', 'Blockchain Account: 0xB2564154287c3406109f1358ED2514BE6EB67', and 'DNI: 060283775'. At the bottom, there is a 'Cerrar Sesión' link and a red button labeled '1 Cambiar a perfil de mecánico'.

Figura 72. Pantalla Perfil de Administrador