



Escuela Técnica Superior de
Ingeniería Informática

PREDICCIÓN DE ERUPCIONES VOLCÁNICAS MEDIANTE INTELIGENCIA ARTIFICIAL

REALIZADO POR:

Aitor Rodríguez Dueñas

TRABAJO DE FIN DE GRADO PARA EL TÍTULO:

Grado en Ingeniería Informática – Ingeniería del Software

TUTOR:

Manuel Jesús Jiménez Navarro

COTUTORA:

María del Mar Martínez Ballesteros

DEPARTAMENTO:

Lenguajes y Sistemas Informáticos

2º Convocatoria del Curso 23/24

Agradecimientos

Reservo este espacio de la memoria para expresar mi agradecimiento a mi familia y amigos por apoyarme no solo en la realización y redacción de este trabajo de fin de grado, sino durante toda mi trayectoria en este grado de ingeniería del software.

También agradecer a mis tutores Manuel Jesús Jiménez Navarro y María del Mar Martínez Ballesteros por la paciencia infinita, fluida comunicación y facilidades a la hora de desarrollar el trabajo. A partir de esta propuesta he descubierto un ámbito del software totalmente desconocido para mí, y gracias a la vocación que tenéis por lo que hacéis, no ha sido difícil contagiarme esa ilusión e interés por la materia.

Un especial agradecimiento a mis compañeros de promoción, gracias por los momentos divertidos y experiencias inolvidables en este grado universitario y por el apoyo en los proyectos compartidos he podido salir adelante y desarrollarme plenamente como ingeniero software, especialmente quiero agradecer a Paula Peña, David Reyes, Pablo Mera, Julio Ribas, Felipe Peña, Diego García, Ramón José Guerrero, Javier García, Manuel Roberto López, José Ramón Baños, Íñigo Ruíz y Alejandro Campano.

Gracias a todos por haberme acompañado y apoyado en el cierre de esta importante etapa de mi vida académica.

Resumen

En este Trabajo de Fin de Grado se profundiza en la predicción de erupciones volcánicas mediante el análisis de datos de series temporales recogidos a partir de sensores volcánicos. Estos datos provienen del Instituto Nacional de Geofísica y Vulcanología de Italia [1]. La inteligencia artificial se emplea para desarrollar modelos de aprendizaje automático que anticipen eventos eruptivos, con el fin de reducir el riesgo volcánico y apoyar a la comunidad científica en la toma de decisiones y en la implementación de medidas preventivas. Estos modelos pueden integrarse en sistemas de monitoreo y alerta para mejorar su fiabilidad y precisión.

El desarrollo de la experimentación se hará por medio de iteraciones en las que mejorará cada fase del proceso de ciencia de datos de manera progresiva. Durante la fase inicial, se aplicarán métodos de procesamiento de datos e ingeniería de características para mejorar la calidad de los datos base. Se desarrollarán modelos de aprendizaje automático como el k -vecinos cercanos, el árbol de decisión o el bosque aleatorio. La efectividad de los modelos de aprendizaje automático se medirá con métricas como el Error Absoluto Medio y el Error Cuadrático Medio y se buscará la automatización del proceso completo. Conforme el avance de las iteraciones, se integrarán nuevas técnicas de procesamiento, validación, extracción y análisis de características por medio de librerías de Python y se implementarán nuevos modelos como el *boosting*.

La validación de estos modelos se utilizará para determinar nuestra posición dentro de la clasificación de la plataforma Kaggle, siendo esta la fuente donde se descargaron los datos de los sensores volcánicos.

La culminación del proyecto es una aplicación web interactiva que presenta los resultados de la experimentación. Esta herramienta permite a los usuarios navegar por los datos y predicciones a través de un mapa interactivo, mejorando la comprensión de la actividad volcánica.

Durante toda la memoria se explicará la planificación del proyecto, el desarrollo de la experimentación y la aplicación web, detallando el seguimiento de las mismas.

Palabras clave: Erupción volcánica, Ciencia de Datos, Series Temporales, Ingeniería de Características, Aprendizaje Automático, Validación, Dataset, Aplicación Web.

Abstract

This thesis explores the prediction of volcanic eruptions by analysing time series data collected from volcanic sensors. These data come from the National Institute of Geophysics and Volcanology of Italy. Artificial intelligence is used to develop machine learning models to anticipate eruptive events in order to reduce volcanic risk and support the scientific community in decision-making and preventive measures. These models can be integrated into monitoring and warning systems to improve their reliability and accuracy.

The development of the experimentation will be carried out through iterations in which each phase of the data science process will be progressively improved. During the initial phase, data processing and feature engineering methods will be applied to improve the quality of the base data. Machine learning models such as k -nearest neighbor, decision tree or random forest will be developed. The effectiveness of machine learning models will be measured by metrics such as mean absolute error and mean square error, and automation of the entire process will be pursued. As iterations progress, new processing, validation, feature extraction and analysis techniques will be integrated using Python libraries, and new models will be implemented such as *boosting*.

Validation of these models will be used to determine our position within the Kaggle platform classification, this being the source where the volcanic sensor data was downloaded.

The culmination of the project is an interactive web application that presents the results of the experimentation. This tool allows users to navigate through the data and predictions via an interactive map, enhancing the understanding of volcanic activity.

Throughout the report, the planning of the project, the development of the experimentation and the web application will be explained, detailing its monitoring.

Keywords: Volcanic Eruption, Data Science, Time Series, Feature Engineering, Machine Learning, Validation, Dataset, Web Application.

Índice general

I	Introducción	1
1.	Introducción	2
1.1.	Contexto	2
1.2.	Objetivos	3
1.3.	Motivación	3
1.4.	Estructura	4
II	Planificación	5
2.	Planificación	6
2.1.	Equipo de trabajo	6
2.1.1.	Roles del Sistema	7
2.2.	Registro de Interesados	7
2.3.	Registro de Requisitos	8
2.3.1.	Requisitos de Información	8
2.3.2.	Requisitos Funcionales	10
2.3.3.	Requisitos No Funcionales	13

2.3.4.	Reglas de Negocio	14
2.3.5.	Casos de Uso	15
2.3.6.	Matriz de Trazabilidad	22
2.4.	Planificación del Tiempo	22
2.4.1.	Lista de hitos	23
2.4.2.	EDT	23
2.4.3.	Diccionario de la EDT	24
2.4.4.	Cronograma	34
2.5.	Estimación de Costes	37
2.5.1.	Coste Directo	37
2.5.2.	Coste Material	37
2.5.3.	Costes Indirectos	38
2.5.4.	Presupuesto	38
2.6.	Gestión	38
2.6.1.	Gestión de Riesgos	39
2.6.2.	Gestión de la Calidad	42
2.6.3.	Gestión de las Comunicaciones	42
2.7.	Metodologías de Desarrollo	43
2.7.1.	Flujo de Trabajo	43
2.7.2.	Política de Commits	44
2.7.3.	Herramientas	44
2.8.	Desviación	45
2.8.1.	Coste Real/Estimado	45
2.8.2.	Replanificación	46
2.8.3.	Desviaciones Totales	47

III	Ejecución	48
3.	Investigación	49
3.1.	Estudio Previo	49
3.1.1.	Ciencia de Datos	49
3.1.2.	Competición de Kaggle	50
3.1.3.	Resumen de los cursos de Kaggle	51
3.1.4.	Modelos	52
3.1.5.	Adaboost	54
3.1.6.	GradientBoost	54
3.1.7.	Validación	55
3.1.8.	Optimización Bayesiana	57
3.2.	1º Iteración	59
3.2.1.	Desarrollo	59
3.2.2.	Seguimiento	61
3.2.3.	Riesgos	62
3.3.	2º Iteración	62
3.3.1.	Desarrollo	63
3.3.2.	Seguimiento	64
3.3.3.	Riesgos	65
3.4.	3º Iteración	65
3.4.1.	Desarrollo	65
3.4.2.	Seguimiento	67
3.4.3.	Riesgos	67
3.5.	4º Iteración	68
3.5.1.	Desarrollo	68
3.5.2.	Seguimiento	70

3.5.3. Riesgos	70
3.6. 5º Iteración	70
3.6.1. Desarrollo	71
3.6.2. Clasificación en la competición	72
3.6.3. Seguimiento	73
3.6.4. Riesgos	74
4. Aplicación Web	75
4.1. Desarrollo	75
4.1.1. Justificación	75
4.1.2. Seguimiento	79
4.1.3. Riesgos	80
4.2. Manual de instalación	81
4.2.1. Manual de instalación del software de experimentación	81
4.2.2. Manual de instalación de la aplicación web	82
4.3. Manual de usuario	84
4.3.1. Manual de usuario del software de experimentación	84
4.3.2. Manual de usuario de la aplicación web	86
4.4. Despliegue	90
IV Cierre	92
5. Cierre	93
5.1. Lecciones Aprendidas	93
5.2. Conclusiones	94
5.3. Trabajo futuro	94
V Anexo	96
Bibliografía	100

Índice de figuras

2.1. Diagrama de Casos de Uso del Software de Experimentación	21
2.2. Diagrama de Casos de Uso de la Aplicación Web	21
2.3. EDT del Trabajo	24
2.4. Cronograma 1. Planificación	34
2.5. Cronograma 2.1. Estudio Previo	34
2.6. Cronograma 2.2. Iteración 1	35
2.7. Cronograma 2.3. Iteración 2	35
2.8. Cronograma 2.4. Iteración 3	35
2.9. Cronograma 2.5. Iteración 4	36
2.10. Cronograma 2.6. Iteración 5	36
2.11. Cronograma 2.7. Desarrollo de la Aplicación Web	36
2.12. Cronograma 3. Cierre	37
2.13. Evolución del Coste Real vs Estimado	46
3.1. Proceso de Validación Cruzada	57
3.2. Proceso de Optimización Bayesiana	58
3.3. Estructura del software de experimentación	63

3.4. Clasificación de Kaggle	72
3.5. Histograma de Frecuencias de tte en meses	73
4.1. Arquitectura de Django	77
4.2. Arquitectura de Vue.js	77
4.3. Diagrama de Componentes	78
4.4. Cobertura de la App	80
4.5. Estructura Datos de Kaggle	82
4.6. Pantalla Principal	86
4.7. Vista del formulario de registro	87
4.8. Rellenar formulario de registro	87
4.9. Vista del formulario de login	88
4.10. Rellenar formulario de login	88
4.11. Encabezado tras login	89
4.12. Vista de perfil de usuario	89
4.13. Popup de volcán	89
4.14. Vista de detalles del volcán	90
4.15. Botón de Logout	90

Índice de cuadros

2.1. Roles del Equipo de Proyecto	6
2.2. Registro Interesado - Alumno.	7
2.3. Registro Interesado - Tutores.	8
2.4. Requisito de Información - RI-001.	8
2.5. Requisito de Información - RI-002.	9
2.6. Requisito de Información - RI-003.	9
2.7. Requisito de Información - RI-004.	9
2.8. Requisito de Información - RI-005.	9
2.9. Requisito Funcional - RF-001.	10
2.10. Requisito Funcional - RF-002.	10
2.11. Requisito Funcional - RF-003.	10
2.12. Requisito Funcional - RF-004.	11
2.13. Requisito Funcional - RF-005.	11
2.14. Requisito Funcional - RF-006.	11
2.15. Requisito Funcional - RF-007.	11
2.16. Requisito Funcional - RF-008.	12

2.17. Requisito Funcional - RF-009.	12
2.18. Requisito Funcional - RF-010.	12
2.19. Requisito Funcional - RF-011.	12
2.20. Requisito Funcional - RF-012.	13
2.21. Requisito Funcional - RF-013.	13
2.22. Requisito No Funcional - RNF-001.	13
2.23. Requisito No Funcional - RNF-003.	14
2.24. Requisito No Funcional - RNF-004.	14
2.25. Requisito No Funcional - RNF-006.	14
2.26. Regla de Negocio - RN-001.	14
2.27. Regla de Negocio - RN-002.	15
2.28. Regla de Negocio - RN-003.	15
2.29. Caso de Uso - CU-001.	15
2.30. Caso de Uso - CU-002.	15
2.31. Caso de Uso - CU-003.	16
2.32. Caso de Uso - CU-004.	16
2.33. Caso de Uso - CU-005.	17
2.34. Caso de Uso - CU-006.	17
2.35. Caso de Uso - CU-007.	18
2.36. Caso de Uso - CU-008.	18
2.37. Caso de Uso - CU-009.	19
2.38. Caso de Uso - CU-010.	19
2.39. Caso de Uso - CU-011.	19
2.40. Caso de Uso - CU-012	20
2.41. Matriz de Trazabilidad de Requisitos	22
2.42. Lista de Hitos del Trabajo	23
2.43. Diccionario EDT: 1.Planificación.	25

2.44. Diccionario EDT: 2.1. Estudio Previo.	26
2.45. Diccionario EDT: 2.2. Iteración 1.	27
2.46. Diccionario EDT: 2.3. Iteración 2.	28
2.47. Diccionario EDT: 2.4. Iteración 3.	29
2.48. Diccionario EDT: 2.5. Iteración 4.	30
2.49. Diccionario EDT: 2.6. Iteración 5.	31
2.50. Diccionario EDT: 2.7. Desarrollo de la Aplicación.	32
2.51. Diccionario EDT: 3. Cierre.	33
2.52. Costes Directos	37
2.53. Costes Material	38
2.54. Costes Indirectos	38
2.55. Presupuesto Total	38
2.56. Estimación Cualitativa de los Riesgos del Proyecto	39
2.57. Riesgos Priorizados y Contingencias	40
2.58. Gestión de los Comunicaciones	43
2.59. Lista de Hitos del Trabajo Replanificado	46
2.60. Tabla de Desviaciones Completa	47
3.1. 1º Iteración (Validación por Retención)	61
3.2. Tareas Iteración 1º	62
3.3. 2º Iteración (Validación Cruzada)	64
3.4. Tareas Iteración 2º	64
3.5. 3º Iteración (Extracción de Características con tsfresh)	67
3.6. Tareas Iteración 3º	67
3.7. 4º Iteración (Optimización Bayesiana)	69
3.8. Tareas Iteración 4º	70
3.9. 5º Iteración (Selección de Características)	72
3.10. Tareas Iteración 5º	73
4.1. Tareas Aplicación Web	79

Índice de algoritmos

1.	KNN	53
2.	Decision Tree	53
3.	Random Forest	54

Índice de códigos extraídos

3.1. Método de validación por retención	59
3.2. Método de KNN	60
3.3. Método de Decision Tree	60
3.4. Método de Randon Forest	60
3.5. Métricas	61
3.6. Método del cálculo del MAPE	61
3.7. KFold	64
3.8. Extracción de características con tsfresh	66
3.9. Modelo de AdaBoosting	68
3.10. Modelo de GradientBoosting	69
3.11. Optimización Bayesiana	69
4.1. Modelos de Django	76
4.2. Urls de volcanes	77

Parte I

Introducción

CAPÍTULO 1

Introducción

En este capítulo se describe el contexto, objetivos y motivación, así como la estructura de este Trabajo Fin de Grado (TFG).

1.1. Contexto

La predicción de erupciones volcánicas es un desafío científico crucial que podría salvar vidas y minimizar los daños causados por estos catastróficos eventos naturales. Por ejemplo, la erupción, en 2021, del volcán Cumbre Vieja, en La Palma, Islas Canarias [2], ha destacado la urgente necesidad de avanzar en la predicción de erupciones volcánicas. Esta erupción tomó a muchas personas por sorpresa y tuvo un impacto significativo en la comunidad local.

En este contexto, se ha decidido utilizar datos de sensores volcánicos proporcionados por una competencia de Kaggle [3], una plataforma reconocida por sus competiciones de ciencia de datos. Esta competencia específica proporciona un conjunto de datos amplio y detallado, ideal para entrenar y evaluar modelos de predicción de erupciones volcánicas. Estos datos provienen del Instituto Nacional de Geofísica y Vulcanología de Italia [1]. La participación en este concurso no sólo proporciona acceso a datos de calidad, sino que también permite comparar resultados con otros investigadores y grupos, creando colaboración y competencia en educación.

1.2. Objetivos

Los objetivos principales de este TFG son los siguientes:

- **Obj1 - Crear un software de experimentación:** Este software servirá como una plataforma para procesar datos de sensores volcánicos, entrenar modelos de aprendizaje automático y evaluar su rendimiento. Deberá ser flexible y escalable para manejar grandes volúmenes de datos y probar diferentes algoritmos de aprendizaje automático, este es el enlace al repositorio del software [4].
- **Obj2 - Realizar análisis de los resultados:** Los resultados obtenidos tras el entrenamiento y optimización de los modelos de aprendizaje automático serán el punto de partida de este análisis en el que se incluirá la evaluación del rendimiento del modelo utilizando métricas, la comparación de diferentes modelos para determinar el más efectivo y el desarrollo de visualizaciones que representen claramente los resultados del análisis, permitiendo una comprensión más intuitiva y efectiva de los datos y las predicciones.
- **Obj3 - Crear una aplicación web:** Desarrollar una aplicación web interactiva que permita visualizar los resultados de la experimentación de manera intuitiva. Esta aplicación proporcionará una interfaz amigable para que los usuarios exploren los datos y las predicciones a través de un mapa interactivo, mejorando la comprensión de la actividad volcánica.
- **Obj4 - Crear una memoria para el TFG:** Documentar todo el proceso de investigación, desde la recopilación y preprocesamiento de datos hasta el desarrollo de modelos de aprendizaje automático y la implementación de la aplicación web. La memoria servirá como un registro detallado de los pasos seguidos y los resultados obtenidos en el proyecto de Trabajo de Fin de Grado.
- **Obj5 - Obtener una posición alta en la clasificación de la competición:** Utilizar los datos proporcionados por Kaggle para entrenar y evaluar los modelos de aprendizaje automático. El objetivo es obtener un rendimiento destacado en la clasificación de la competición, lo que indica la eficacia de los modelos desarrollados en la predicción de erupciones volcánicas.

1.3. Motivación

La motivación principal para este trabajo surge del interés en el desarrollo y la aplicación de técnicas avanzadas de ingeniería informática, particularmente en el ámbito del aprendizaje automático junto con el análisis de series temporales. Este proyecto ofrece una oportunidad única

para profundizar en áreas que me apasionan el diseño de algoritmos de aprendizaje automático, mejorar en la creación de aplicaciones web interactivas y descubrir nuevas temáticas como las series temporales y la ingeniería de características.

Además, este trabajo me permite poner en práctica los conocimientos adquiridos durante el grado en Ingeniería Informática de Software, aplicando principios y buenas prácticas en la planificación y gestión de proyectos. A través de este proyecto, tengo la oportunidad de demostrar habilidades en la organización y seguimiento de tareas, así como en la implementación de metodologías ágiles para asegurar un desarrollo eficiente y de alta calidad. Integrar estas competencias en un problema real y desafiante no solo reforzará mis capacidades técnicas, sino que también contribuirá significativamente a mi desarrollo profesional como ingeniero informático.

1.4. Estructura

El trabajo está organizado en 5 partes:

- En la **Parte I Introducción** se describe el marco conceptual del trabajo y la estructura del mismo.
- En la **Parte II Planificación** se describe la planificación del trabajo.
- En la **Parte III Ejecución** se describe todo el desarrollo de la experimentación, la aplicación web y su seguimiento.
- En la **Parte IV Cierre** se presentan las conclusiones y cierre del trabajo.
- En la **Parte V Anexo** se añaden las actas de reunión.

Parte II

Planificación

CAPÍTULO 2

Planificación

En esta parte se describirá toda la planificación y todos los registros necesarios previos al desarrollo del proyecto de investigación, experimentación y desarrollo web. La planificación seguirá en cierta medida la estructura del PMBOK (Project Management Body of Knowledge), la cual proporciona un marco estándar y reconocido internacionalmente para la gestión de proyectos. [5].

2.1. Equipo de trabajo

Lo primero para planificar el proyecto será presentar los roles necesarios del equipo de proyecto:

Cuadro 2.1: Roles del Equipo de Proyecto

Rol	Descripción
Jefe de Proyecto	Encargado de liderar y coordinar el equipo de trabajo, asegurando que se cumplan los objetivos del proyecto en tiempo y forma. Además de ser el liderar las comunicaciones con el resto de interesados.

Rol	Descripción
Analista	Responsable de analizar los requisitos del proyecto, identificar las necesidades de los interesados y proponer soluciones técnicas adecuadas.
Especialista de Calidad	Responsable de asegurar que los productos, servicios o procesos de una organización cumplan con los estándares de calidad establecidos.
Desarrollador	Encargado de implementar las soluciones propuestas, escribiendo y probando el código necesario para el funcionamiento del proyecto.
Tester	Responsable de realizar pruebas de calidad sobre el software desarrollado, identificando y reportando posibles fallos o errores.

2.1.1. Roles del Sistema

- **Usuario Final** : Científicos, investigadores y usuarios interesados en la predicción de erupciones volcánicas.
- **Investigador** : Responsable de supervisar el desarrollo del proyecto y garantizar que se cumplan los objetivos.

2.2. Registro de Interesados

El registro de interesados es fundamental para identificar a todas las partes que tienen un interés en el proyecto. En este registro se exponen sus datos, así como sus intereses, expectativas y el nivel de influencia que pueden tener en el desarrollo del proyecto.

Cuadro 2.2: Registro Interesado - Alumno.

Nombre	Aitor Rodríguez Dueñas.
Rol	Alumno. (Jefe proyecto, Especialista de Calidad, Desarrollador, Tester y Analista).
Organización	Estudiante Grado Ingeniería Informática del Software.

Intereses	<ul style="list-style-type: none"> ▪ Desarrollar un proyecto de alta calidad que cumpla con los requisitos académicos. ▪ Aprender y aplicar técnicas de inteligencia artificial y desarrollo web y obtener una buena calificación en el TFG.
Influencia	Alta.

Cuadro 2.3: Registro Interesado - Tutores.

Nombre	Manuel Jesús Jiménez Navarro y María del Mar Martínez Balles- teros.
Rol	Especialistas de Calidad.
Organización	Departamento Lenguajes y Sistemas Informáticos de la ETSII.
Intereses	<ul style="list-style-type: none"> ▪ Evaluar el trabajo del estudiante de manera objetiva. ▪ Ofrecer orientación y asegurar los requisitos de la entrega.
Influencia	Alta.

2.3. Registro de Requisitos

Para definir el alcance del proyecto y la línea base se hará uso de la recopilación de requisitos, casos de uso y matriz de trazabilidad.

2.3.1. Requisitos de Información

Estos requisitos detallan las necesidades y expectativas relacionadas con la información que debe ser gestionada, almacenada y comunicada durante el proyecto. A continuación, se recopilan los requisitos de información:

Cuadro 2.4: Requisito de Información - RI-001.

Nombre	RI-001 - Datos de Sensores Volcánicos.
---------------	--

Descripción	COMO Investigador QUIERO un conjunto de atributos (máximos, mínimos, cruces por cero, medias y desviaciones típicas) a partir de los 10 sensores volcánicos de las series temporales y los tiempos de erupción reales de cada volcán PARA poder analizar y predecir erupciones.
Prioridad	Alta.

Cuadro 2.5: Requisito de Información - RI-002.

Nombre	RI-002 - Características de los Volcanes.
Descripción	COMO Usuario Final QUIERO ver el nombre, localización (coordenadas), altura y tiempo de erupción restante de cada volcán PARA estar informado sobre la situación actual de los volcanes.
Prioridad	Media.

Cuadro 2.6: Requisito de Información - RI-003.

Nombre	RI-003 - Datos de Usuario.
Descripción	COMO Usuario Final QUIERO que los datos de los usuarios sean nombre de usuario, contraseña y correo electrónico PARA gestionar y autenticar a los usuarios de la aplicación.
Prioridad	Baja.

Cuadro 2.7: Requisito de Información - RI-004.

Nombre	RI-004 - Métricas de los Modelos.
Descripción	COMO Investigador QUIERO generar un registro con las predicciones, métricas de los modelos, nombre de los modelos y sus hiperparámetros correspondientes tras cada iteración PARA evaluar y mejorar los modelos predictivos.
Prioridad	Alta.

Cuadro 2.8: Requisito de Información - RI-005.

Nombre	RI-005 - Gráficos para Métricas.
---------------	----------------------------------

Descripción	COMO Investigador QUIERO generar gráficos que muestren datos de interés como los volcanes con menor y mayor tiempo de erupción y distribuciones de los tiempos de erupción PARA validar y comparar los modelos predictivos.
Prioridad	Alta.

2.3.2. Requisitos Funcionales

A continuación, se recopilan los requisitos funcionales del software de experimentación y la aplicación web:

Cuadro 2.9: Requisito Funcional - RF-001.

Nombre	RF-001 - Procesamiento de Datos.
Rol	Investigador
Descripción	El software de experimentación debe leer y procesar los datos de entrada, aplicando las técnicas de ingeniería de características pertinentes según la iteración, formando un dataset para entrenar los modelos.
Prioridad	Alta.

Cuadro 2.10: Requisito Funcional - RF-002.

Nombre	RF-002 - Entrenamiento de Modelos.
Rol	Investigador
Descripción	El software de experimentación debe entrenar diversos modelos de aprendizaje automáticos siendo posible ajustar los hiperparámetros mediante un archivo externo.
Prioridad	Alta

Cuadro 2.11: Requisito Funcional - RF-003.

Nombre	RF-003 - Validación de Modelos.
Rol	Investigador

Descripción	El software debe procesar los datos de entrada formando un dataset para entrenar los modelos.
Prioridad	Alta.

Cuadro 2.12: Requisito Funcional - RF-004.

Nombre	RF-004 - Optimización de Modelos.
Rol	Investigador
Descripción	El software de experimentación debe procesar los datos de entrada formando un dataset para entrenar los modelos.
Prioridad	Alta.

Cuadro 2.13: Requisito Funcional - RF-005.

Nombre	RF-005 - Generar archivo de salida.
Rol	Investigador
Descripción	El software de experimentación debe generar un archivo con los volcanes y su tiempo de erupción (time to eruption).
Prioridad	Alta.

Cuadro 2.14: Requisito Funcional - RF-006.

Nombre	RF-006 - Generar gráficos.
Rol	Investigador
Descripción	El software de experimentación debe generar gráficos sobre distribuciones referentes a los datos de entrada, post-procesados o de salida o métricas de modelos y rendimiento.
Prioridad	Media.

Cuadro 2.15: Requisito Funcional - RF-007.

Nombre	RF-007 - Generación de Reportes.
Rol	Investigador

Descripción	El software de experimentación de experimentación debe permitir la generación de reportes personalizados del rendimiento de los modelos.
Prioridad	Alta.

Cuadro 2.16: Requisito Funcional - RF-008.

Nombre	RF-008 - Autenticación de Usuarios.
Rol	Usuario Final
Descripción	La aplicación web debe permitir a los usuarios autenticarse mediante un nombre de usuario y contraseña.
Prioridad	Media.

Cuadro 2.17: Requisito Funcional - RF-009.

Nombre	RF-009 - Gestión de Perfiles.
Rol	Usuario Final
Descripción	La aplicación web debe permitir a los usuarios gestionar su perfil, incluyendo la actualización de información personal y la configuración de preferencias.
Prioridad	Baja.

Cuadro 2.18: Requisito Funcional - RF-010.

Nombre	RF-010 - Filtro de Información.
Rol	Usuario Final
Descripción	La aplicación web debe proporcionar una funcionalidad de búsqueda que permita a los usuarios filtrar mediante palabras claves y atributos los volcanes que se muestran.
Prioridad	Media.

Cuadro 2.19: Requisito Funcional - RF-011.

Nombre	RF-011 - Mapa Interactivo.
Rol	Usuario Final

Descripción	La aplicación web debe mostrar todos los volcanes y cada uno de ellos debe ser interactivo, mostrando así un <i>popup</i> o algo similar.
Prioridad	Alta.

Cuadro 2.20: Requisito Funcional - RF-012.

Nombre	RF-012 - Alarma de Erupción.
Rol	Usuario Final
Descripción	La aplicación web debe de mostrar cuando un volcán está cerca de erupcionar, es decir, supere el umbral de tiempo de una semana previa a la erupción debe de diferenciarse de otros volcanes.
Prioridad	Alta.

Cuadro 2.21: Requisito Funcional - RF-013.

Nombre	RF-013 - Volcanes Favoritos.
Rol	Usuario Final
Descripción	La aplicación web debe tener una forma de añadir un volcán a un registro personal del usuario a modo de favoritos.
Prioridad	Baja.

2.3.3. Requisitos No Funcionales

A continuación se recopilan los requisitos no funcionales:

Cuadro 2.22: Requisito No Funcional - RNF-001.

Nombre	RNF-001 - Código refactorizado.
Rol	Investigador
Descripción	Tanto el software de experimentación como la aplicación web deben desarrollarse siguiendo los estándares de calidad.
Prioridad	Media.

Cuadro 2.23: Requisito No Funcional - RNF-003.

Nombre	RNF-002 - Idioma.
Rol	Usuario Final
Descripción	El idioma del software de experimentación y la web será inglés.
Prioridad	Media.

Cuadro 2.24: Requisito No Funcional - RNF-004.

Nombre	RNF-003 - Aplicación web Responsive.
Rol	Usuario Final
Descripción	La aplicación web debe poder utilizarse de manera adecuada en distintas resoluciones.
Prioridad	Baja.

Cuadro 2.25: Requisito No Funcional - RNF-006.

Nombre	RNF-004 - Memoria del trabajo
Rol	Investigador
Descripción	El documento debe mantener una estructura y consistencia en términos de estilo, terminología y formato a lo largo de toda la memoria. Además todos los gráficos, tablas y figuras incluidas en la memoria deben tener una calidad adecuada, ser legibles y estar debidamente etiquetados y referenciados en el texto.
Prioridad	Alta.

2.3.4. Reglas de Negocio

A continuación, se recopilan las reglas de negocio:

Cuadro 2.26: Regla de Negocio - RN-001.

Nombre	RN-001 - Limitar modelos de entrada.
Descripción	Si se introduce un modelo que no esté implementado debe aparecer un mensaje de aviso.

Cuadro 2.27: Regla de Negocio - RN-002.

Nombre	RN-002 - Limitar tipo de procesado de entrada.
Descripción	Si se introduce un tipo de procesado que no esté implementado debe aparecer un mensaje de aviso.

Cuadro 2.28: Regla de Negocio - RN-003.

Nombre	RN-003 - Garantizar omisión de procesos.
Descripción	Se debe dar opción a omitir procesos como el de procesado u optimización y el software siga funcionando correctamente.

2.3.5. Casos de Uso

Para diseñar los casos de uso del sistema, haremos uso de los roles del sistema definidos en esta sección 2.1.1 y luego describimos las interacciones entre estos actores y el sistema:

Cuadro 2.29: Caso de Uso - CU-001.

Nombre	CU-001: Procesamiento de Datos.
Actor	Investigador
Descripción	El sistema debe leer y procesar los datos de entrada, aplicando técnicas de ingeniería de características para formar un dataset para entrenar los modelos.
Precondiciones	Datos de sensores volcánicos disponibles.
Flujo Principal	<ul style="list-style-type: none"> ▪ El Investigador carga los datos de entrada. ▪ El sistema aplica técnicas de ingeniería de características. ▪ Se genera un dataset para entrenamiento.
Postcondiciones	Dataset preparado para el entrenamiento de modelos.

Cuadro 2.30: Caso de Uso - CU-002.

Nombre	CU-002: Entrenamiento de Modelos.
Actor	Investigador

Descripción	El sistema debe entrenar modelos predictivos utilizando el dataset procesado.
Precondiciones	Dataset procesado disponible.
Flujo Principal	<ul style="list-style-type: none"> ▪ El Investigador selecciona el modelo a entrenar. ▪ El sistema entrena el modelo con el dataset. ▪ Se generan métricas de rendimiento del modelo.
Postcondiciones	Modelo entrenado con métricas de rendimiento.

Cuadro 2.31: Caso de Uso - CU-003.

Nombre	CU-003: Validación de Modelos.
Actor	Investigador
Descripción	El sistema debe validar los modelos entrenados utilizando un conjunto de datos de validación.
Precondiciones	Modelos entrenados disponibles.
Flujo Principal	<ul style="list-style-type: none"> ▪ El Investigador selecciona el modelo a validar. ▪ El sistema valida el modelo con los datos de validación. ▪ Se generan métricas de validación MAE, MSE y MAPE.
Postcondiciones	Métricas de validación generadas.

Cuadro 2.32: Caso de Uso - CU-004.

Nombre	CU-004: Optimización de Modelos.
Actor	Investigador
Descripción	El sistema debe optimizar los modelos entrenados para mejorar su rendimiento.
Precondiciones	Modelos entrenados disponibles.

Flujo Principal	<ul style="list-style-type: none"> ▪ El Investigador selecciona los parámetros a optimizar. ▪ El sistema realiza la optimización utilizando técnicas como la optimización Bayesiana. ▪ Se generan modelos optimizados con nuevas métricas de rendimiento.
Postcondiciones	Modelos optimizados.

Cuadro 2.33: Caso de Uso - CU-005.

Nombre	CU-005: Automatización de Procesos.
Actor	Investigador
Descripción	El sistema debe permitir la automatización de procesos de procesamiento, optimización, entrenamiento y validación.
Precondiciones	Funcionalidades de procesamiento, optimización, entrenamiento y validación disponibles.
Flujo Principal	<ul style="list-style-type: none"> ▪ Al Investigador se le pregunta por consola que procesos quiere realizar. ▪ El sistema va ejecutando los procesos automáticamente según las respuestas. ▪ Se generan resultados de manera automática.
Postcondiciones	Resultados generados automáticamente.

Cuadro 2.34: Caso de Uso - CU-006.

Nombre	CU-006: Generar Archivo de Salida.
Actor	Investigador
Descripción	El sistema debe generar un archivo con los volcanes y su tiempo de erupción.
Precondiciones	Modelos validados disponibles.

Flujo Principal	<ul style="list-style-type: none"> ■ El Investigador selecciona el modelo y tipo de procesado a incluir en el nombre del archivo. ■ El sistema genera el archivo de salida.
Postcondiciones	Archivo de salida generado.

Cuadro 2.35: Caso de Uso - CU-007.

Nombre	CU-007: Generar Gráficos
Actor	Investigador
Descripción	El sistema debe generar gráficos sobre las distribuciones de los datos y las métricas de rendimiento de los modelos.
Precondiciones	Datos procesados o modelos validados disponibles.
Flujo Principal	<ul style="list-style-type: none"> ■ El Investigador selecciona los datos o métricas a graficar. ■ El sistema genera los gráficos correspondientes.
Postcondiciones	Gráficos generados y visualizados.

Cuadro 2.36: Caso de Uso - CU-008.

Nombre	CU-008: Autenticación de Usuarios
Actor	Usuario Final
Descripción	La aplicación web debe permitir a los usuarios autenticarse mediante un nombre de usuario y contraseña.
Precondiciones	Funcionalidad de autenticación implementada.
Flujo Principal	<ul style="list-style-type: none"> ■ El Usuario Final ingresa sus credenciales. ■ El sistema verifica las credenciales. ■ El usuario accede al sistema si las credenciales son correctas.
Postcondiciones	Usuario autenticado.

Cuadro 2.37: Caso de Uso - CU-009.

Nombre	CU-009: Gestión de Perfiles
Actor	Usuario Final
Descripción	La aplicación web debe permitir a los usuarios gestionar su perfil.
Precondiciones	Usuario autenticado.
Flujo Principal	<ul style="list-style-type: none"> ▪ El Usuario Final accede a su perfil. ▪ El usuario actualiza la información personal y preferencias. ▪ El sistema guarda los cambios.
Postcondiciones	Perfil actualizado.

Cuadro 2.38: Caso de Uso - CU-010.

Nombre	CU-010: Búsqueda de Información.
Actor	Usuario Final
Descripción	La aplicación web debe proporcionar una funcionalidad de búsqueda que permita a los usuarios buscar información específica.
Precondiciones	Datos de información disponibles.
Flujo Principal	<ul style="list-style-type: none"> ▪ El Usuario Final ingresa palabras clave en la barra de búsqueda. ▪ El sistema busca la información relevante. ▪ Se muestran solo los volcanes que coincidan con los resultados de la búsqueda.
Postcondiciones	Resultados de búsqueda mostrados.

Cuadro 2.39: Caso de Uso - CU-011.

Nombre	CU-011: Mapa Interactivo.
Actor	Usuario Final
Descripción	La aplicación web debe mostrar todos los volcanes y permitir la interacción con ellos.
Precondiciones	Datos de volcanes disponibles.

Flujo Principal	<ul style="list-style-type: none"> ■ El Usuario Final accede al mapa interactivo. ■ El usuario selecciona un volcán. ■ El sistema muestra información detallada del volcán seleccionado.
Postcondiciones	Información del volcán mostrada.

Cuadro 2.40: Caso de Uso - CU-012

Nombre	CU-012: Volcanes Favoritos.
Actor	Usuario Final
Descripción	La aplicación web debe permitir a los usuarios añadir un volcán a un registro personal de favoritos.
Precondiciones	Usuario autenticado.
Flujo Principal	<ul style="list-style-type: none"> ■ El Usuario Final navega al volcán que desea añadir a favoritos. ■ El usuario selecciona la opción para añadir a favoritos. ■ El sistema guarda el volcán en el registro personal de favoritos del usuario.
Postcondiciones	Volcán añadido a la lista de favoritos del usuario.

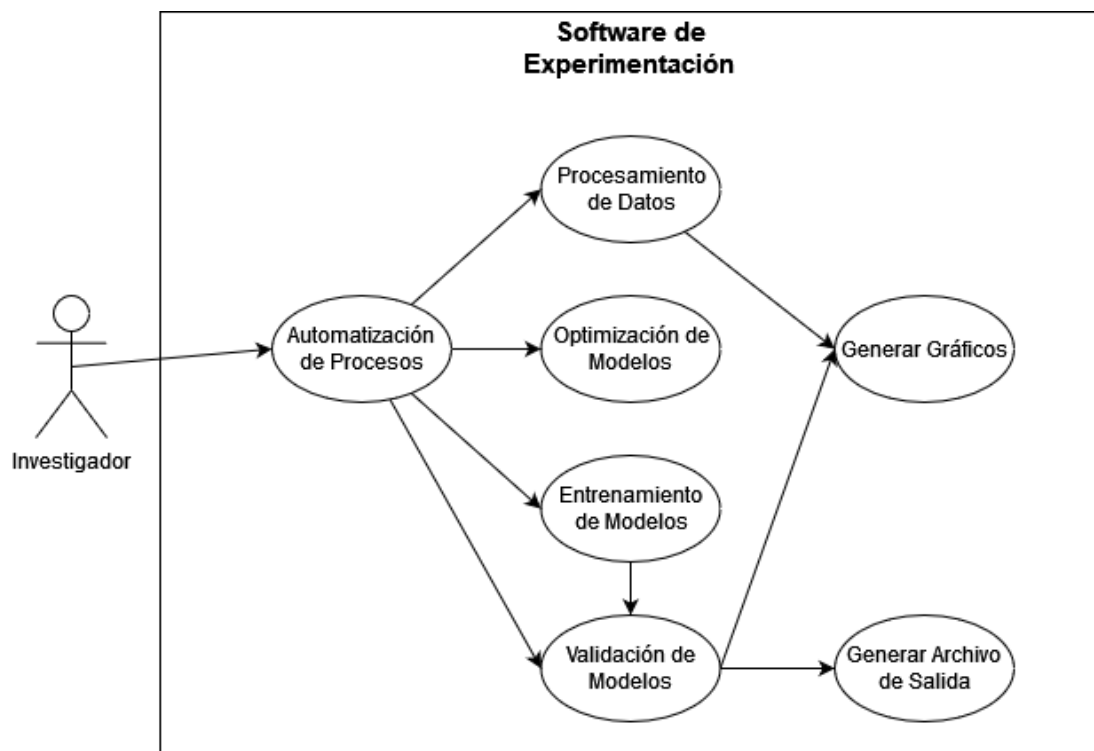


Figura 2.1: Diagrama de Casos de Uso del Software de Experimentación

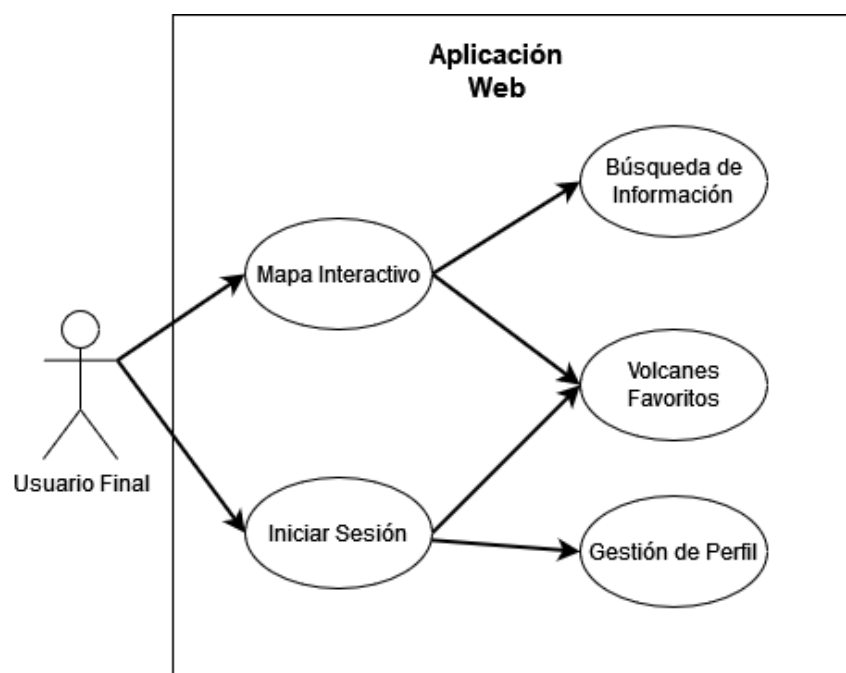


Figura 2.2: Diagrama de Casos de Uso de la Aplicación Web

2.3.6. Matriz de Trazabilidad

La siguiente tabla muestra la matriz de trazabilidad de requisitos y los objetivos definidos previamente en la sección 1.2 que cumplen.

Cuadro 2.41: Matriz de Trazabilidad de Requisitos

Requisitos	Obj1	Obj2	Obj3	Obj4	Obj5
RI-001	X	X			X
RI-002			X		
RI-003			X		
RI-004	X	X			X
RI-005	X	X			X
RF-001	X				X
RF-002	X				X
RF-003	X				X
RF-004	X				X
RF-005	X				
RF-006		X			
RF-007		X			
RF-008			X		
RF-009			X		
RF-010			X		
RF-011			X		
RF-012			X		
RF-013			X		
RNF-001	X		X		
RNF-002	X		X		
RNF-003			X		
RNF-004				X	

2.4. Planificación del Tiempo

En esta sección se listarán los hitos del trabajo, se describirá la Estructura de Desglose del Trabajo (EDT) y se secuencian las actividades con el cronograma.

2.4.1. Lista de hitos

La lista de hitos del trabajo se muestra en la siguiente tabla 2.42:

Cuadro 2.42: Lista de Hitos del Trabajo

Id	Nombre	Fecha de Inicio	Fecha de Finalización
Hito-001	Planificación Completa	18/12/2023	10/01/2024
Hito-002	Estudio Previo	10/01/2024	21/01/2024
Hito-003	Primera Iteración Completa	21/01/2024	18/02/2024
Hito-004	Segunda Iteración Completa	18/02/2024	10/03/2024
Hito-005	Tercera Iteración Completa	10/03/2024	31/03/2024
Hito-006	Cuarta Iteración Completa	31/03/2024	14/04/2024
Hito-007	Quinta Iteración Completa	14/04/2024	28/04/2024
Hito-008	Aplicación Web Completa	28/04/2024	16/05/2024
Hito-009	Entrega de Memoria	16/05/2024	27/05/2024
Hito-010	Proyecto Cerrado	27/05/2024	05/06/2024

2.4.2. EDT

La EDT del trabajo se muestra en el siguiente cuadro 2.3:

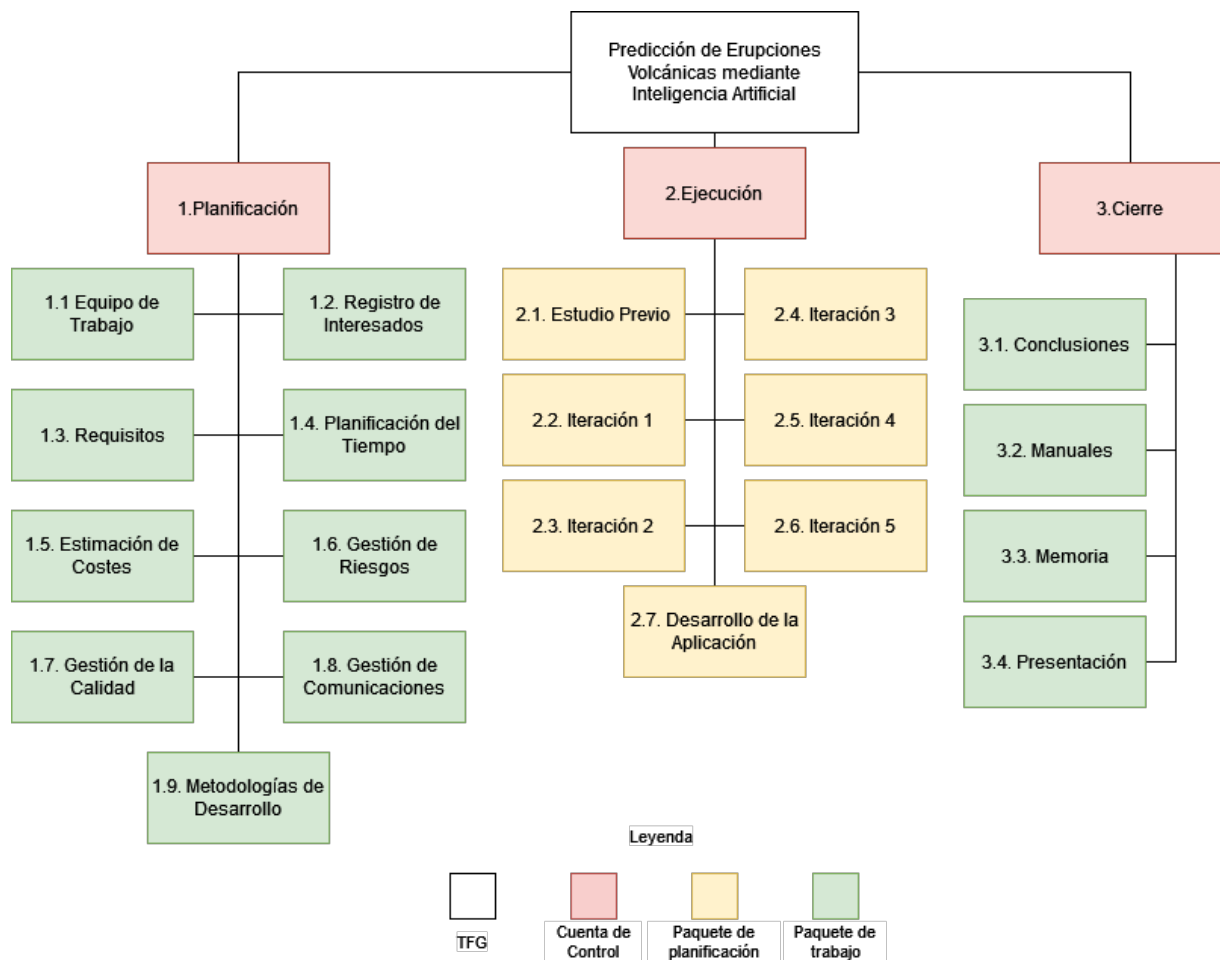


Figura 2.3: EDT del Trabajo

2.4.3. Diccionario de la EDT

En esta sección se exponen el diccionario de la EDT, especificando las actividades de las cuentas de control y paquetes de planificación, así como las fechas de inicio y final, las horas de trabajo y los costes estimados de las actividades.

Cuadro 2.43: Diccionario EDT: 1.Planificación.

ID	Nombre	Descripción	Fecha Inicio	Fecha Final	Rol	Horas	Coste
1.1	Equipo de Trabajo	Selección de roles necesarios del proyecto.	18/12/2023	19/12/2023	Jefe de Proyecto	3	118,44 €
1.2	Registro de Interesados	Definir y clasificar los Interesados del Proyecto.	19/12/2023	21/12/2023	Jefe de Proyecto	5	197,40 €
1.3	Requisitos	Recopilar, clasificar y priorizar los requisitos.	21/12/2023	27/12/2023	Analista	27	888,30 €
1.4	Planificación del Tiempo	Identificar los hitos, crear la EDT y cronograma.	27/12/2023	31/12/2023	Jefe de Proyecto	17	671,16 €
1.5	Estimación de Costes	Estimación de Costes siguiendo la estimación PERT.	31/12/2023	03/01/2024	Jefe de Proyecto	15	592,20 €
1.6	Gestión de Riesgos	Identificar los riesgos del proyecto y crear planes de contingencias.	03/01/2024	05/01/2024	Analista	9	296,10 €
1.7	Gestión de la Calidad	Diseñar el plan para conservar y auditar la calidad del trabajo.	05/01/2024	07/01/2024	Especialista Calidad	8	286,80 €
1.8	Gestión de las Comunicaciones	Crear el plan de Comunicaciones.	07/01/2024	08/01/2024	Jefe de Proyecto	5	197,40 €
1.9	Metodologías de Desarrollo	Selección de las metodologías de desarrollo y herramientas.	08/01/2024	10/01/2024	Analista	7	230,30 €
Total	Planificación		18/12/2023	10/01/2024		96	3478,10 €

Cuadro 2.44: Diccionario EDT: 2.1. Estudio Previo.

ID	Nombre	Descripción	Fecha Inicio	Fecha Final	Rol	Horas	Coste
2.1.1.	Curso Python	Curso Introductorio de Kaggle de Python.	10/01/2024	11/01/2024	Desarrollador	5	119,25 €
2.1.2.	Curso Pandas	Curso Introductorio de Kaggle de la librería Pandas.	11/01/2024	12/01/2024	Desarrollador	4	95,40 €
2.1.3.	Curso Introducción a Machine Learning	Curso Introductorio de Kaggle de Machine Learning con la librería SKLearn.	12/01/2024	13/01/2024	Desarrollador	3	71,55 €
2.1.4.	Curso Avanzado a Machine Learning	Curso Avanzado de Kaggle de Machine Learning con la librería SKLearn.	13/01/2024	15/01/2024	Desarrollador	4	95,40 €
2.1.5.	Curso Feature Engineering	Curso Introductorio de Kaggle de Ingeniería de Características.	15/01/2024	16/01/2024	Desarrollador	5	119,25 €
2.1.6.	Curso Data Visualization	Curso Introductorio de Kaggle de Visualización de Datos con la librería Seaborn.	16/01/2024	17/01/2024	Desarrollador	4	95,40 €
2.1.7.	Documentar el Estudio Previo	Documentación de formación y conceptos adquiridos en el estudio previo a las iteraciones.	18/01/2024	21/01/2024	Jefe Proyecto	5	197,40 €
Total	Estudio Previo		10/01/2024	21/01/2024		30	793,65 €

Cuadro 2.45: Diccionario EDT: 2.2. Iteración 1.

ID	Nombre	Descripción	Fecha Inicio	Fecha Final	Rol	Horas	Coste
2.2.1	Crear Proyecto	Creación del entorno de trabajo y configuración inicial del proyecto.	22/01/2024	25/01/2024	Desarrollador	2	47,70€
2.2.2	Procesamiento de Datos 1	Procesamiento inicial de los datos para su posterior análisis.	25/01/2024	01/02/2024	Desarrollador	4	95,40€
2.2.3	Modelos 1	Implementación de modelos de aprendizaje automático.	01/02/2024	08/02/2024	Desarrollador	4	95,40€
2.2.4	Validación 1	Validación inicial de los modelos implementados.	08/02/2024	14/02/2024	Desarrollador	3	71,55€
2.2.5	Gráficos 1	Generación de gráficos para visualización de datos.	14/02/2024	16/02/2024	Desarrollador	3	71,55€
2.2.6	Revisión de Iteración 1	Revisión general de los avances de la iteración y planificación para la siguiente fase.	16/02/2024	18/02/2024	Jefe de Proyecto	4	157,92€
Total	Iteración 1		22/01/2024	18/02/2024		20	539,52€

Cuadro 2.46: Diccionario EDT: 2.3. Iteración 2.

ID	Nombre	Descripción	Fecha Inicio	Fecha Final	Rol	Horas	Coste
2.3.1	Diseño de arquitectura	Diseñar y crear la arquitectura del proyecto para automatizar procesos y configurarlos.	18/02/2024	21/02/2024	Desarrollador	5	119,25 €
2.3.2	Procesamiento de Datos 2	Mejorar el procesamiento de los datos para su posterior análisis.	21/02/2024	25/02/2024	Desarrollador	3	71,55 €
2.3.3	Modelos 2	Adaptar la implementación de modelos de aprendizaje automático a las mejoras.	25/02/2024	29/02/2024	Desarrollador	4	95,40 €
2.3.4	Validación 2	Implementar la validación cruzada.	29/02/2024	03/03/2024	Desarrollador	5	119,25 €
2.3.5	Gráficos 2	Generación de gráficos para visualización de datos.	03/03/2024	07/03/2024	Desarrollador	3	71,55 €
2.3.6	Revisión de Iteración 2	Revisión general de los avances de la iteración y planificación para la siguiente fase.	07/03/2024	10/03/2024	Jefe de Proyecto	5	197,40 €
Total	Iteración 2		18/02/2024	10/03/2024		25	674,40€

Cuadro 2.47: Diccionario EDT: 2.4. Iteración 3.

ID	Nombre	Descripción	Fecha Inicio	Fecha Final	Rol	Horas	Coste
2.4.1	Utilizar tsfresh	Instalar, comprender y utilizar la librería tsfresh como manera alternativa de procesar los datos.	10/03/2024	13/03/2024	Desarrollador	5	119,25 €
2.4.2	Procesamiento de Datos 3	Mejorar el procesamiento de los datos para su posterior análisis.	13/03/2024	15/03/2024	Desarrollador	3	71,55 €
2.4.3	Modelos 3	Adaptar la implementación de modelos de aprendizaje automático a las mejoras.	15/03/2024	20/03/2024	Desarrollador	4	95,40 €
2.4.4	Validación 3	Adaptar la validación de los modelos implementados.	20/03/2024	23/03/2024	Desarrollador	5	119,25 €
2.4.5	Gráficos 3	Generación de gráficos para visualización de datos.	23/03/2024	26/03/2024	Desarrollador	3	71,55 €
2.4.6	Revisión de Iteración 3	Revisión general de los avances de la iteración y planificación para la siguiente fase.	26/04/2024	31/04/2024	Jefe de Proyecto	5	197,40 €
Total	Iteración 3		10/03/2024	31/03/2024		25	674,40€

Cuadro 2.48: Diccionario EDT: 2.5. Iteración 4.

ID	Nombre	Descripción	Fecha Inicio	Fecha Final	Rol	Horas	Coste
2.5.1	Utilizar optimización Bayesiana	Instalar, comprender y aplicar la optimización bayesiana a los modelos.	31/03/2024	04/04/2024	Desarrollador	6	142,20€
2.5.2	Procesamiento de Datos 4	Mejorar el procesamiento de los datos para su posterior análisis.	04/04/2024	06/04/2024	Desarrollador	3	71,55€
2.5.3	Modelos 4	Implementación de modelos de aprendizaje automático ADABOOST y Gradient BOOST.	06/04/2024	08/04/2024	Desarrollador	3	71,55€
2.5.4	Validación 4	Adaptar la validación de los modelos implementados.	08/04/2024	10/04/2024	Desarrollador	1	23,85€
2.5.5	Gráficos 4	Generación de gráficos para visualización de datos.	10/04/2024	12/04/2024	Desarrollador	3	71,55€
2.5.6	Revisión de Iteración 4	Revisión general de los avances de la iteración y planificación para la siguiente fase.	12/04/2024	14/04/2024	Jefe de Proyecto	4	157,92€
Total	Iteración 4		31/03/2024	14/04/2024		20	539,52€

Cuadro 2.49: Diccionario EDT: 2.6. Iteración 5.

ID	Nombre	Descripción	Fecha Inicio	Fecha Final	Rol	Horas	Coste
2.6.1	Procesamiento de Datos 5	Mejorar el procesamiento de los datos para su posterior análisis.	14/04/2024	16/04/2024	Desarrollador	3	71,55€
2.6.2	Modelos 5	Adaptar la implementación de modelos de aprendizaje automático a las mejoras.	16/04/2024	18/04/2024	Desarrollador	3	71,55€
2.6.3	Validación 5	Adaptar la validación de los modelos implementados.	18/04/2024	20/04/2024	Desarrollador	3	71,55€
2.6.4	Gráficos 5	Generación de gráficos para visualización de datos.	20/04/2024	22/04/2024	Desarrollador	3	71,55€
2.6.5	Revisión de Iteración 5	Revisión general de los avances de la iteración.	22/04/2024	24/04/2024	Jefe de Proyecto	4	157,92€
2.6.6	Revisión global de la experimentación	Finalizar la experimentación con un análisis de los resultados y extracción de conclusiones de ellos.	24/04/2024	28/04/2024	Jefe Proyecto	4	157,92€
Total	Iteración 5		14/04/2024	28/04/2024		20	602,04€

Cuadro 2.50: Diccionario EDT: 2.7. Desarrollo de la Aplicación.

ID	Nombre	Descripción	Fecha Inicio	Fecha Final	Rol	Horas	Coste
2.7.1	Crear el Proyecto	Crear el repositorio de Github y configurar las tecnologías.	28/04/2024	30/04/2024	Desarrollador	2	47,70€
2.7.2	Mapa	Crear el mapa interactivo y crear los modelos.	30/04/2024	02/05/2024	Desarrollador	1	23,85€
2.7.3	Vista de Detalles	Crear la vista de detalles de un volcán.	02/05/2024	04/05/2024	Desarrollador	3	71,55€
2.7.4	Usuarios	Implementar la gestión de usuarios y autenticación.	04/05/2024	06/05/2024	Desarrollador	4	95,40€
2.7.5	Volcanes Favoritos	Implementar la función de añadir los volcanes a una lista de favoritos.	06/05/2024	08/05/2024	Desarrollador	3	71,55€
2.7.6	Filtro de Búsqueda	Añadir un filtro de búsqueda de volcanes por ciertos criterios.	08/05/2024	12/05/2024	Desarrollador	5	119,25€
2.7.7	Testing	Desarrollar los test de la aplicación.	12/05/2024	14/05/2024	Tester	3	71.04€
2.7.8	Revisión global y despliegue de la aplicación	Revisión completa de la aplicación y desplegarla en alguna plataforma de despliegue.	14/05/2024	16/05/2024	Jefe Proyecto	4	157,92€
Total	Desarrollo de la Aplicación		28/04/2024	16/05/2024		25	658,26€

Cuadro 2.51: Diccionario EDT: 3. Cierre.

ID	Nombre	Descripción	Fecha Inicio	Fecha Final	Rol	Horas	Coste
3.1	Lecciones Aprendidas	Redactar las lecciones aprendidas.	16/05/2024	17/05/2024	Jefe de Proyecto	4	157,92€
3.2	Redactar Manuales	Redactar el manual de instalación de la aplicación.	17/05/2024	19/05/2024	Jefe de Proyecto	5	197,40€
3.3	Finalizar Memoria	Finalizar y revisar la memoria del TFG.	19/06/2024	25/05/2024	Jefe de Proyecto	15	592,20€
3.4	Preparar la entrega	Revisión de los criterios de entrega del TFG y ver su cumplimentación.	25/05/2024	27/05/2024	Jefe de Proyecto	5	197,40€
3.5	Preparar la presentación	Realizar y preparar la presentación de la defensa del TFG.	27/05/2024	05/06/2024	Jefe de Proyecto	10	394,80€
Total	Cierre		16/05/2024	05/06/2024		39	1539,72€

2.4.4. Cronograma

En los siguientes cuadros se describe la secuenciación de las actividades mediante un cronograma realizado en Microsoft Excel:

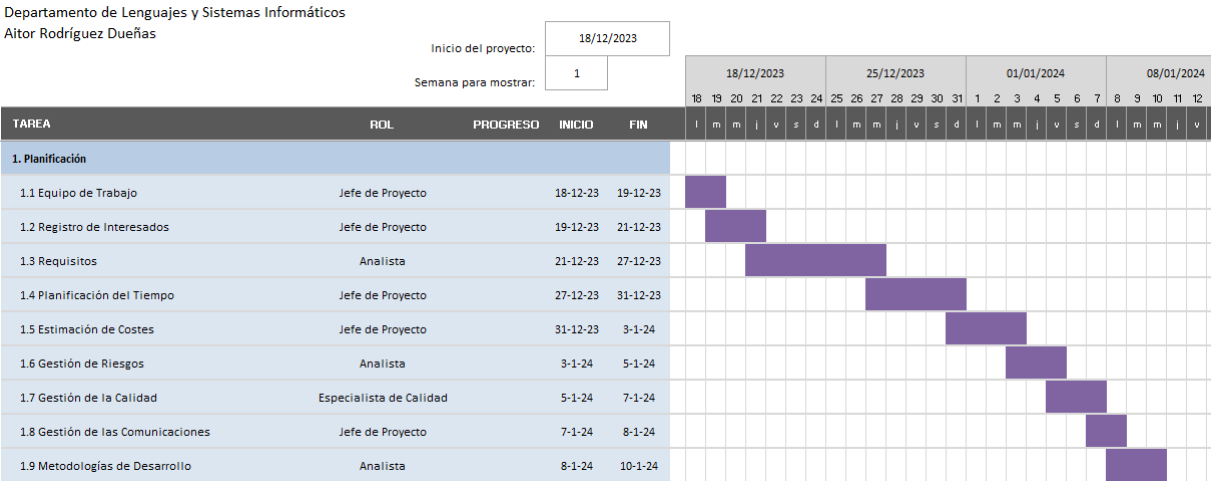


Figura 2.4: Cronograma 1. Planificación



Figura 2.5: Cronograma 2.1. Estudio Previo



Figura 2.6: Cronograma 2.2. Iteración 1



Figura 2.7: Cronograma 2.3. Iteración 2

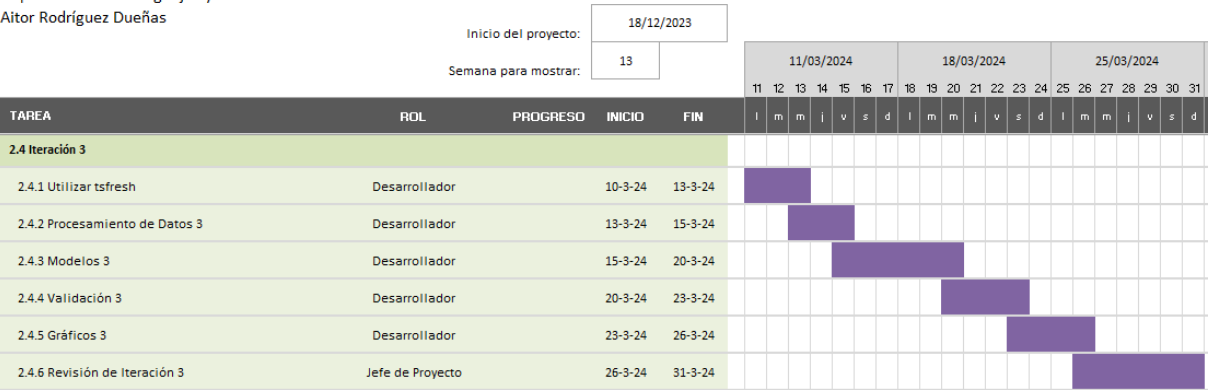


Figura 2.8: Cronograma 2.4. Iteración 3



Figura 2.9: Cronograma 2.5. Iteración 4



Figura 2.10: Cronograma 2.6. Iteración 5

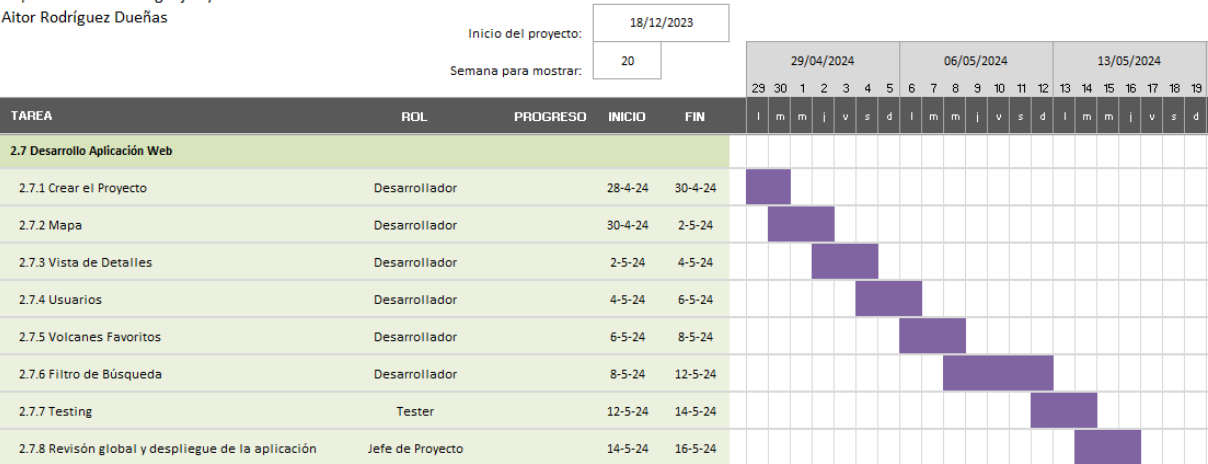


Figura 2.11: Cronograma 2.7. Desarrollo de la Aplicación Web



Figura 2.12: Cronograma 3. Cierre

2.5. Estimación de Costes

En esta sección se describen los distintos costes del trabajo para calcular un presupuesto general.

2.5.1. Coste Directo

Los costes directos vienen dados por los sueldos de los trabajadores. En el siguiente cuadro 2.52 se muestra el cálculo total.

Cuadro 2.52: Costes Directos

Rol	Horas	Precio/h (€)	Coste Total (€)
Jefe de Proyecto	119	39.48	4698.12
Desarrollador	127	23.85	3028.95
Analista	43	32.90	1414.70
Consultor Especialista de Calidad	8	35.85	286.80
Tester	3	23.68	71.04
Total	300		9499.61

2.5.2. Coste Material

Se dispone de un ordenador con una gráfica NVIDIA GeForce RTX 2060, un procesador AMD Ryzen 7 2700X y 16GB de RAM. Se amortiza a lo largo de 4 años con 6 horas de uso de lunes a viernes. En el siguiente cuadro 2.53 se muestra el cálculo total.

Cuadro 2.53: Costes Material

Nombre	Coste Total (€)	Coste/Hora (€)	Horas Totales	Coste Total (€)
PC	1000	0.16	300	48

2.5.3. Costes Indirectos

El precio medio de la factura mensual de electricidad en España se ha obtenido calculando la media de facturas mensuales de 2023 de este informe [6]. En cuanto al acceso a Internet, la tarifa al mes de la fibra es de 24€. En el siguiente cuadro 2.54 se muestra el cálculo total.

Cuadro 2.54: Costes Indirectos

Nombre	Unidad	Horas	Coste/mes (€)	Coste (€/h)	Coste Total (€)
Electricidad	kWh	300	58.78	0.0816	24.48
Internet	mes	-	24	-	144
Total					168.48

2.5.4. Presupuesto

El presupuesto total se calcula sumando los costes directos, indirectos y fijos, y añadiendo un 10 % de contingencia y un 5 % de gestión de riesgos, y restando un 10 % de beneficio. En el siguiente cuadro 2.55 se muestra el cálculo total del presupuesto.

Cuadro 2.55: Presupuesto Total

Concepto	Coste (€)
Costes Directos	9500.61
Costes Material	48
Costes Indirectos	168.48
Subtotal	9717.09
Contingencia (10 %)	971.71
Beneficios (-5 %)	485.85
Total	10202.95

2.6. Gestión

En esta sección se explican los planes de gestión de riesgos, calidad y comunicaciones que se llevarán a cabo en este TFG.

2.6.1. Gestión de Riesgos

En esta sección se identifican una lista de riesgos en el siguiente cuadro 2.56, cada uno evaluado no solo en términos de probabilidad sino también de impacto: en el alcance, los costos y el tiempo del proyecto.

Cuadro 2.56: Estimación Cualitativa de los Riesgos del Proyecto

ID	Riesgo	Impacto Alcance	Impacto Costes	Impacto Tiempo	Probabilidad
RISK-001	Dificultad para realizar las actividades usando las nuevas tecnologías sin experiencia previa.	Alta	Media	Alta	Media
RISK-002	Falta de potencia de procesamiento o excesiva tardanza en los procesos del software de experimentación.	Media	Alta	Alta	Alta
RISK-003	Pérdida de datos o problemas de integridad de datos durante el procesamiento.	Alta	Alta	Media	Baja
RISK-004	Falta de acceso a herramientas o software necesarios para el desarrollo.	Media	Media	Baja	Media
RISK-005	Problemas de compatibilidad y rendimiento de la aplicación web en diferentes dispositivos y navegadores.	Media	Baja	Media	Baja
RISK-006	Dificultad para cumplir con los requisitos de seguridad y autenticación de usuarios en la aplicación web.	Alta	Media	Alta	Media
RISK-007	Inconsistencias y errores en la generación de reportes y gráficos.	Baja	Media	Media	Baja
RISK-008	Falta de claridad en la definición de los requisitos y/o ambigüedad.	Alta	Baja	Media	Baja

(Continúa en la siguiente página)

(Continuación desde la anterior página)

ID	Riesgo	Impacto Alcance	Impacto Costes	Impacto Tiempo	Probabilidad
RISK-009	Cambios en los requisitos del proyecto a medida que avanza el desarrollo.	Alta	Media	Alta	Alta
RISK-010	Problemas en la comunicación y coordinación con el tutor.	Baja	Baja	Media	Baja
RISK-011	Dificultad para mantener la documentación del proyecto actualizada y consistente.	Media	Baja	Media	Baja
RISK-012	Fallos del hardware utilizado.	Baja	Alta	Alta	Baja
RISK-013	No alcanzar las fechas de los hitos.	Alta	Alta	Alta	Media

A continuación se realizará un análisis y priorizarán los riesgos se hará uso de esta fórmula para calcular el impacto de forma cuantitativa:

$$\text{Impacto Total} = \text{Probabilidad} \times (\text{Impacto Alcance} + \text{Impacto Costes} + \text{Impacto Tiempo})$$

Para ello necesitamos cuantificar los valores cualitativos asignados previamente donde Alto = 3, Medio = 2 y Bajo = 1. Para aplicar el principio de Pareto, primero calculamos el impacto total de cada riesgo multiplicando la prioridad por la suma de los impactos en alcance, costes y tiempo. Luego, ordenamos los riesgos de mayor a menor impacto total y calculamos el porcentaje acumulado del impacto total. Finalmente, identificamos aquellos riesgos cuyo impacto acumulado alcance al menos el 80 % del total y se asignarán como prioridad Alta (principio de Pareto). Procedemos además a describir los planes de contingencia para cada riesgo. Cada uno de estos requisitos y planes será revisado en cada auditoría de calidad, como se explica en esta sección 2.6.2. Aquí está el resultado:

Cuadro 2.57: Riesgos Priorizados y Contingencias

ID	Resultado del Impacto	Prioridad	Plan de Contingencia
RISK-001	16	Media	Proporcionar formación y recursos adicionales para acelerar la curva de aprendizaje en nuevas tecnologías.

(Continúa en la siguiente página)

(Continuación desde la anterior página)

ID	Resultado del Impacto	Prioridad	Plan de Contingencia
RISK-002	24	Alta	Reajustar parámetros de procesos y refactorizar el código para evitar redundancias y/o iteraciones que alarguen los procesos.
RISK-003	8	Baja	Implementar protocolos de seguridad de datos robustos y realizar copias de seguridad frecuentes.
RISK-004	10	Media	Asegurar acceso temprano a herramientas y software mediante licencias o alternativas de código abierto.
RISK-005	5	Baja	Realizar pruebas exhaustivas en diferentes dispositivos y navegadores para garantizar la compatibilidad.
RISK-006	16	Media	Desarrollar e implementar un plan de seguridad integral para la autenticación de usuarios.
RISK-007	5	Baja	Establecer un proceso de revisión y validación para los reportes y gráficos generados.
RISK-008	6	Baja	Clarificar los requisitos con todas las partes interesadas y documentarlos detalladamente.
RISK-009	27	Alta	Establecer un proceso de gestión de cambios para manejar las modificaciones de requisitos de manera eficiente.
RISK-010	4	Baja	Mejorar la comunicación con el tutor a través de reuniones regulares y actualizaciones de progreso.
RISK-011	5	Baja	Utilizar herramientas de gestión de documentos para mantener la documentación sincronizada y actualizada.

(Continúa en la siguiente página)

(Continuación desde la anterior página)

ID	Resultado del Impacto	Prioridad	Plan de Contingencia
RISK-012	7	Baja	Realizar mantenimiento preventivo y tener hardware de respaldo disponible.
RISK-013	18	Alta	Replanificar a otra convocatoria o recortar el alcance de la aplicación y/o la experimentación.

2.6.2. Gestión de la Calidad

Para evaluar la calidad del proyecto, se establecen los siguientes indicadores:

- **Grado de cumplimiento.** Durante los seguimientos se revisarán todos los objetivos y requisitos que se deben cumplir y medir ese grado de cumplimentación a modo de porcentaje.
- **Objetividad en la experimentación.** La experimentación deberá seguir de manera rigurosa los procedimientos de manera estricta que posibilite la reproducibilidad.
- **Código estructurado y probado.** El código del proyecto debe estar bien estructurado y refactorizado. Intentando siempre que las funciones hagan solo una tarea. Además que la cobertura de pruebas debe mantenerse como mínimo en un 60 %, siendo lo ideal un 80 %.
- **Reuniones de auditoría de calidad.** Las reuniones periódicas con el tutor no solo servirán para revisar el progreso del proyecto, sino también para realizar auditorías de calidad.

2.6.3. Gestión de las Comunicaciones

El tiempo destinado a las comunicaciones está planificado dentro de la actividad de las iteraciones *Revisión de XX* donde XX es la iteración en cuestión, las cuáles se pueden consultar en la sección de diccionario de la EDT 2.4.3. Las reuniones por tanto tiene una periodicidad por iteración, salvo alguna reunión extraordinaria que surja. El medio de las reuniones será de manera presencial en el despacho de Manuel Jesús Jiménez Navarro o si es en línea por una videoconferencia en Microsoft Teams. De cada reunión se genera un acta de reunión con lo hablado en la reunión, fecha, duración y asistentes. Cualquier otra información que se quiera comunicar se hará a través del correo electrónico de la Universidad de Sevilla.

Cuadro 2.58: Gestión de los Comunicaciones

Información	Fecha	Interesado/a	Propósito
Reunión Inicial	18/12/2023	Tutor y Cotutora	Explicación de la propuesta y discusión de la planificación de alto nivel.
Reunión Estudio Previo	22/01/2024	Tutor y Cotutora	Revisar y auditar los resultados de la iteración realizada para organizar la posterior iteración.
Reunión Iteración 1	19/02/2024	Tutor y Cotutora	Revisar y auditar los resultados de la iteración realizada para organizar la posterior iteración.
Reunión Iteración 2	11/03/2024	Tutor y Cotutora	Revisar y auditar los resultados de la iteración realizada para organizar la posterior iteración.
Reunión Iteración 3	01/04/2024	Tutor y Cotutora	Revisar y auditar los resultados de la iteración realizada para organizar la posterior iteración.
Reunión Iteración 4	15/04/2024	Tutor y Cotutora	Revisar y auditar los resultados de la iteración realizada para organizar la posterior iteración.
Reunión Iteración 5	29/04/2024	Tutor y Cotutora	Revisar y auditar los resultados de la iteración realizada para organizar la aplicación web a crear.
Reunión Web	16/05/2024	Tutor y Cotutora	Desplegar y probar la aplicación y auditar el resultado final.
Reunión Final	27/05/2024	Tutor y Cotutora	Revisar la entrega y todos los requisitos de esta.

2.7. Metodologías de Desarrollo

En esta sección se describen las metodologías que se emplearán en el desarrollo del TFG y herramientas que se utilizarán.

2.7.1. Flujo de Trabajo

Será un trabajo que se desarrollará por iteraciones donde al final de éstas se realizará una revisión y posteriormente una reunión con el tutor para correcciones y organización de la siguiente iteración.

2.7.2. Política de Commits

Se usará Conventional Commits, es decir, el título del commit (no más de 50 caracteres) viene precedido por un tag según lo que se intente subir al remoto y posteriormente una descripción opcional (no más de 80 caracteres). La estructura pues sería la siguiente: `git commit -m "tag: título commit" -m "Descripción commit"` Algunas de estos tags:

- **fix:** Utilizado cuando se realiza una corrección de errores.
- **feat:** Se emplea cuando se agrega una nueva característica o funcionalidad.
- **docs:** Usado para cambios relacionados con la documentación.
- **style:** Se utiliza para cambios que no afectan el comportamiento del código, como cambios en el formato o estilo de código.
- **refactor:** Utilizado para cambios en el código que no corrigen errores ni añaden nuevas funcionalidades, pero mejoran la estructura o legibilidad del código.
- **test:** Se emplea para agregar o modificar pruebas unitarias o de integración.
- **chore:** Utilizado para cambios en el proceso de compilación o tareas de mantenimiento.
- **perf:** Usado para mejoras de rendimiento.
- **revert:** Se emplea cuando se revierte un commit previo.
- **deploy:** Se utiliza para se agregan archivos o cambios para el despliegue.

2.7.3. Herramientas

Las herramientas que se utilizarán serán:

- **Github** para gestión de código. GitHub es una plataforma de desarrollo colaborativo que permite gestionar y almacenar versiones de proyectos de software. Facilita la colaboración mediante el control de versiones y las funcionalidades de integración continua. Esta es la página oficial [7].
- El lenguaje de programación tanto para el software de experimentación como para la aplicación web será **Python**. Python es un lenguaje de programación interpretado de alto nivel, conocido por su legibilidad y simplicidad. Es ampliamente utilizado en ciencia de datos, desarrollo web y automatización. Esta es la página oficial [8].
- En el software de experimentación se utilizarán algunas librerías:

- **SKLearn** (scikit-learn). Una librería de aprendizaje automático en Python que proporciona herramientas simples y eficientes para el análisis de datos y la modelación predictiva. Enlace a la página oficial [9].
 - **Seaborn**. Una librería de visualización de datos basada en matplotlib que proporciona una interfaz de alto nivel para dibujar gráficos estadísticos atractivos e informativos. Enlace a la página oficial [10].
 - **Numpy**. Una librería fundamental para la computación científica en Python que soporta grandes matrices y arreglos multidimensionales, así como una colección de funciones matemáticas de alto nivel para operar con estos arreglos. Enlace a la página oficial [11].
 - **Pandas**. Una librería de manipulación y análisis de datos que ofrece estructuras de datos y funciones de análisis de alto rendimiento y fáciles de usar. Enlace a la página oficial [12].
- Para la aplicación Web se utilizarán:
- **Vue.js** para el Frontend de la aplicación. Un framework progresivo de JavaScript para construir interfaces de usuario. Es conocido por su fácil integración y su capacidad para escalar a aplicaciones más complejas mediante bibliotecas oficiales y soluciones de soporte. Enlace a la página oficial [13].
 - **Django** para el Backend de la aplicación. Un framework de alto nivel de Python que fomenta el desarrollo rápido y el diseño limpio y pragmático. Django incluye una gran cantidad de funcionalidades listas para usar, lo que facilita la creación de aplicaciones web seguras y mantenibles. Enlace a la página oficial [14].
 - **Render** para el Despliegue de la aplicación. Una plataforma de alojamiento en la nube que permite desplegar aplicaciones web de manera fácil y eficiente. Render soporta una variedad de lenguajes y frameworks, y simplifica el proceso de despliegue continuo y gestión de aplicaciones. Enlace a la página oficial [15].

2.8. Desviación

En esta sección se muestran las desviaciones y costes finales del trabajo.

2.8.1. Coste Real/Estimado

A continuación, la evolución del coste acumulado planificado y real en el siguiente cuadro 2.13:

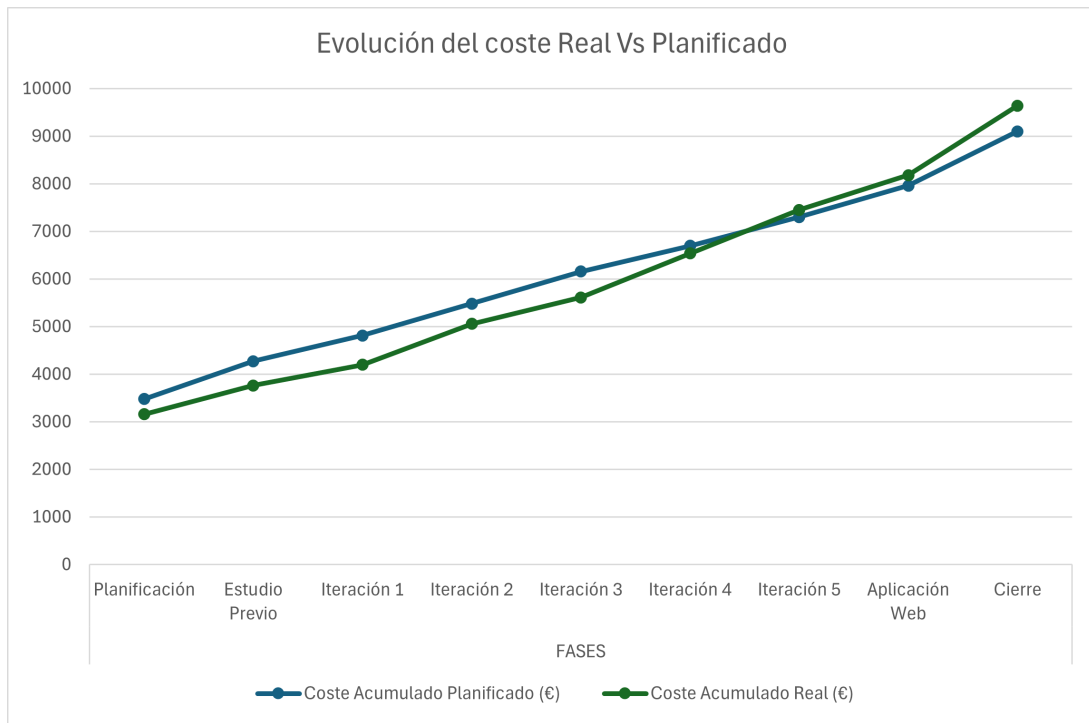


Figura 2.13: Evolución del Coste Real vs Estimado

2.8.2. Replanificación

Tras la tercera iteración se realizó una replanificación a la segunda convocatoria, y se acortó el alcance de la aplicación web para llegar al presupuesto y trabajo planificados. Por tanto, las nuevas fechas de los hitos restantes se ven en el siguiente cuadro 2.59.

Cuadro 2.59: Lista de Hitos del Trabajo Replanificado

Id	Nombre	Fecha de Inicio	Fecha de Finalización
Hito-006	Cuarta Iteración Completa	26/04/2024	10/05/2024
Hito-007	Quinta Iteración Completa	10/05/2024	27/05/2024
Hito-008	Aplicación Web Completa	27/05/2024	14/06/2024
Hito-009	Entrega de Memoria	14/06/2024	24/06/2024
Hito-010	Proyecto Cerrado	24/06/2024	03/07/2024

Las actividades de la aplicación web que quedaron fuera del alcance son:

- 2.7.5 - Volcanes Favoritos.
- 2.7.6 - Filtro de Búsqueda.

Ver el diccionario de la EDT para más información sobre las actividades mencionadas en la sección 2.50.

2.8.3. Desviaciones Totales

En el siguiente cuadro 2.60 se mostrarán las desviaciones de cada fase en cuento a trabajo y costes. En la fase de cierre no se contarán las 10 horas dedicadas para la presentación puesto que la memoria se entrega previo a la entrega de la propia presentación.

Cuadro 2.60: Tabla de Desviaciones Completa

Fase	Trabajo Estimado	Trabajo Real	Coste Estimado	Coste Real	Coste Acumulado Estimado	Coste Acumulado Real
Planificación	96h	87h (-9h)	3478.10€	3159.93€	3478.10€	3159.93€
Estudio Previo	30h	22h (-8h)	793.65€	602.85€	4271.75€	3762.78€
Iteración 1	20h	17h (-3h)	539.52€	436.71€	4811.27€	4199.49€
Iteración 2	25h	32h (+7h)	674.40€	860.60€	5485.67€	5060.09€
Iteración 3	25h	20h (-5h)	674.40€	555.15€	6160.07€	5615.24€
Iteración 4	20h	36h (+16h)	539.52€	920.22€	6699.59€	6535.46€
Iteración 5	20h	24h (+4h)	602.04€	917.88€	7301.63€	7453.34€
Aplicación Web	25h	28h (+3h)	658.26€	729.81€	7959.89€	8183.15€
Cierre	29h	37h (+8h)	1144.92€	1460.76€	9104.81€	9643.91€
Total	290h	303h (13+h)			9104.81€	9643.91€

Como podemos observar en la tabla anterior 2.60 el coste final se ha podido mantener dentro del presupuesto gracias al porcentaje guardado de contingencia y las decisiones de replanificación y recorte de alcance tomadas durante el trabajo.

Parte III

Ejecución

CAPÍTULO 3

Investigación

En esta parte de la memoria se explicará todo el desarrollo en las iteraciones mostrando y describiendo los avances.

Primero se comenzará con el estudio previo de la investigación, luego el desarrollo de las iteraciones de la experimentación y finalmente el desarrollo de la aplicación.

3.1. Estudio Previo

El estudio previo de este trabajo se ha realizado a partir de dos fuente principales, los apuntes de la asignatura Inteligencia Artificial del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Sevilla y los cursos de Kaggle recomendados por el tutor de este trabajo.

3.1.1. Ciencia de Datos

Comenzamos el estudio previo describiendo qué es la ciencia de datos (*data science*) y los procesos que lo conforman.

La **ciencia de datos** es una disciplina fundamental para explotar datos y generar conocimiento. Su objetivo es encontrar modelos que describan patrones y comportamientos a partir de los datos para tomar decisiones o hacer predicciones. Ha crecido significativamente debido

al acceso a grandes volúmenes de datos y su tratamiento en tiempo real, requiriendo técnicas sofisticadas para abordar problemas como la escalabilidad, robustez, y adaptabilidad. Esta área involucra a numerosos grupos de investigación en computación, estadística, matemáticas, ingeniería, entre otros, que trabajan en el desarrollo de nuevos algoritmos, técnicas de computación e infraestructuras para la captura, almacenamiento y procesamiento de datos. [16]

Los procesos principales que conforman la ciencia de datos incluyen:

- **Selección de los Datos:** Este proceso implica identificar y seleccionar las fuentes de datos relevantes para el problema que se desea resolver. Esto puede incluir bases de datos, archivos CSV, datos de sensores, datos de redes sociales, entre otros. La calidad y la relevancia de los datos seleccionados son cruciales para el éxito del análisis.
- **Procesamiento de los Datos:** Una vez seleccionados, los datos suelen necesitar ser limpiados y transformados. Esto puede incluir la eliminación de valores atípicos, el manejo de datos faltantes, la normalización de datos y la transformación de variables. El objetivo es preparar los datos para que sean adecuados para el análisis y modelado.
- **Ingeniería de Características:** La ingeniería de características consiste en crear nuevas variables o características a partir de los datos existentes que ayuden a mejorar el rendimiento predictivo del modelo, reducir las necesidades computacionales o mejorar la interpretabilidad de los resultados. Por ejemplo, se puede determinar la importancia de las características con información mutua o inventar nuevas características en dominios de problemas reales, entre otras técnicas [17].
- **Modelos:** En esta etapa se construyen modelos matemáticos y estadísticos que pueden aprender de los datos. Esto incluye la selección de algoritmos de aprendizaje automático, el entrenamiento de los modelos con datos y la optimización de los parámetros del modelo para mejorar su rendimiento.
- **Validación:** La validación es el proceso de evaluar la precisión y la generalización del modelo construido. Esto se realiza utilizando un conjunto de datos independiente que no se utilizó durante el entrenamiento del modelo. La validación asegura que el modelo funcione bien no solo con los datos de entrenamiento sino también con datos nuevos y no vistos.

3.1.2. Competición de Kaggle

El punto de partida del trabajo es una competición lanzada por Kaggle, la cual es una plataforma para la ciencia de datos y el aprendizaje automático donde organizaciones diseñan y publican competiciones.

Esta competición se titula **INGV - Volcanic Eruption Prediction** [3], dado que los datos provienen del *Istituto Nazionale di Geofisica e Vulcanologia* [1] de Italia. Detectar erupciones volcánicas antes de que ocurran es un problema importante que históricamente ha demostrado ser muy difícil. Esta competición proporciona lecturas de varios sensores sísmicos alrededor de un volcán y desafía a los participantes a estimar cuánto tiempo falta para la próxima erupción. Los datos representan un clásico problema de procesamiento de señales que ha resistido los métodos tradicionales. La competición tiene como objetivo mejorar la predicción de erupciones volcánicas mediante el análisis de datos sísmicos. Utilizando técnicas de ciencia de datos, los participantes deberán desarrollar algoritmos que identifiquen signos tempranos de una erupción inminente, permitiendo así evacuaciones más oportunas y salvando potencialmente miles de vidas.

Los datos están estructurados de la siguiente manera:

- **train.csv**: Metadata para los archivos de entrenamiento. Es un archivo csv individual que guarda la siguiente información:
 - **segment_id**: Código de identificación del segmento de datos. Coincide con el nombre del archivo de datos asociado.
 - **time_to_eruption**: El valor objetivo, el tiempo hasta la próxima erupción.
- **[train—test]/*.csv**: Archivos de datos. Cada archivo contiene diez minutos de registros de diez sensores diferentes ubicados alrededor de un volcán.

Como se ha hecho mención cada archivo contiene diez minutos de registros de sensores, por lo que estamos tratando series temporales. Las **series temporales** son archivos donde los datos han sido registrados en tramos sucesivos de tiempo. Cada archivo tiene 60000 filas por lo que deducimos que el tramo sucesivo de tiempo es 0,01 segundos.

Las predicciones se evalúan en base al error absoluto medio (*Mean Absolute Error*, MAE). Al final de la experimentación debe obtenerse un archivo con la estructura del archivo train.csv con las predicciones.

3.1.3. Resumen de los cursos de Kaggle

Como se ha mencionado previamente parte de los fundamentos teóricos y prácticos de este trabajo provienen de los propios cursos de Kaggle. Los cursos realizados son los siguiente:

- **Python**: Este curso introduce los fundamentos de Python, cubre los principales usos de este lenguaje, inclusive este curso tenía como objetivo secundario revisar conceptos ya aprendidos en el grado [18].

- **Pandas:** Pandas es una biblioteca crucial en Python para la manipulación y análisis de datos estructurados. Este curso enseña cómo utilizar Pandas para cargar, limpiar, agrupación y transformaciones de datos [19].
- **Intro to Machine Learning:** El curso proporciona una introducción básica y práctica al aprendizaje automático, cubriendo desde conceptos básicos como la selección de modelos y evaluación hasta la construcción de modelos de regresión y clasificación utilizando bibliotecas como Scikit-Learn en Python [20].
- **Intermediate Machine Learning:** Avanzando desde los fundamentos, este curso aborda técnicas más avanzadas en aprendizaje automático. Incluye temas como la optimización de modelos, manejo de datos faltantes, aprendizaje no supervisado, validación cruzada y técnicas de optimización de hiperparámetros [21].
- **Data Visualization:** Este curso cubre cómo utilizar bibliotecas como Matplotlib y Seaborn en Python para crear gráficos informativos y visualmente atractivos, puesto que son cruciales para comunicar resultados [22].
- **Feature Engineering:** En este curso se explora cómo mejorar los datos para mejorar el rendimiento de los modelos de aprendizaje automático. Incluye técnicas como la selección de características, creación de nuevas características, codificación de variables categóricas y reducción de dimensionalidad [17].

3.1.4. Modelos

Los modelos utilizados en la investigación serán cinco e irán incluyéndose al software de manera progresiva. Los tres primeros modelos son el KNN, el árbol de decisión (*Decision Tree*) y el bosque aleatorio (*Random Forest*).

El **KNN** o clasificador k-vecinos más cercanos (en inglés *k nearest neighbours*) es un modelo adecuado para abordar tanto tareas de clasificación como de regresión. Dado un ejemplo del dominio de la tarea, el modelo le asocia una clase, en el caso de una tarea de clasificación, o un valor numérico, en el caso de una tarea de regresión, en función de las clases o valores asociados a ejemplos similares del conjunto de entrenamiento.

El modelo recibe un ejemplo x como entrada actúa de la siguiente manera:

Algoritmo 1 KNN

```
1: Inicializar una lista vacía distances
2: for  $(x_i, y_i)$  en entrenamiento do
3:   Calcular la distancia entre  $x$  y  $x_i$ :  $dist \leftarrow \text{distancia}(x, x_i)$ 
4:   Agregar  $(dist, y_i)$  a distances
5: end for
6: Ordenar distances por la distancia (ascendente) y seleccionar los primeros  $k$  elementos
7: Calcular el valor medio de los  $y_i$  seleccionados
8: return Predicción para  $x$ 
```

El cálculo de la distancia se puede hacer de varias maneras, en este trabajo se optó por la distancia euclidiana entre dos vectores \mathbf{x} y \mathbf{x}' . Que se define como:

$$d(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^n (x_i - x'_i)^2$$

El **Árbol de Decisión** (del inglés, *Decision Tree*) es un tipo de modelo adecuado tanto para abordar tareas de clasificación como de regresión, a partir tanto de atributos discretos como continuos, pero necesariamente numéricos. En este caso se utilizó un árbol decisión de regresión pues buscamos predecir un valor numérico. Estos árboles son binarios y en cada nodo interno está etiquetado con un atributo y un valor umbral para ese atributo y cada nodo hoja está etiquetado un valor numérico. Dado un ejemplo del dominio de la tarea, el modelo le asocia una salida recorriendo el árbol desde la raíz hasta una de las hojas: en cada nodo interno con atributo X y umbral u elige la rama de la izquierda si en el ejemplo el valor de X no supera u y la rama de la derecha en caso contrario; la respuesta para el ejemplo es el valor numérico asociado a la hoja a la que se llegue.

Algoritmo 2 Decision Tree

```
1: if nodivisible( $\mathcal{D}$ ) then
2:   return un nodo etiquetado con etiqueta( $\mathcal{D}$ )
3: else
4:   Elegir el par  $(X, u)$  que proporcione la mejor partición  $(\mathcal{D}_{\text{Izq}}, \mathcal{D}_{\text{Der}})$  de  $\mathcal{D}$ 
5:    $T1 \leftarrow \text{Decision Tree}(\mathcal{D}_{\text{Izq}})$ 
6:    $T2 \leftarrow \text{Decision Tree}(\mathcal{D}_{\text{Der}})$ 
7:   return un nodo etiquetado con  $(X, u)$  y cuyos hijos sean  $T1$  y  $T2$ 
8: end if
```

El **Bosque Aleatorio** (del inglés, *Random Forest*) es un modelo que combina múltiples árboles de decisión para mejorar la precisión y robustez del modelo predictivo. Cada árbol de

decisión en el Random Forest se entrena en un subconjunto aleatorio de los datos de entrenamiento y luego predice el valor. El resultado final se determina por votación, donde la clase o valor más frecuente entre todos los árboles se elige como la predicción del Random Forest. El modelo trabaja de la siguiente manera:

Algoritmo 3 Random Forest

- 1: **Entrada:** Conjunto de datos de entrenamiento, número de árboles N , tamaño del subconjunto k
 - 2: Inicializar un conjunto vacío de árboles: T_1, T_2, \dots, T_N
 - 3: **for** $i = 1$ to N **do**
 - 4: Seleccionar aleatoriamente k muestras del conjunto de entrenamiento con reemplazo
 - 5: Entrenar un árbol de decisión T_i usando las muestras seleccionadas
 - 6: Agregar el árbol entrenado T_i al conjunto de árboles
 - 7: **end for**
 - 8: Promediar las predicciones de cada árbol T_i
 - 9: **return** Determinar el valor con más votos o el promedio como predicción final
-

Progresivamente durante las iteraciones se implementarán dos nuevos modelos el AdaBoost y el GradientBoost.

3.1.5. Adaboost

El **AdaBoost** (Adaptive Boosting) es un método de aprendizaje automático diseñado para mejorar la precisión de los modelos predictivos mediante la combinación de múltiples clasificadores débiles en un clasificador fuerte. Este algoritmo se resume en que comienza usando clasificadores débiles con un rendimiento deficiente y posteriormente los siguientes clasificadores irán corrigiendo los errores de los anteriores clasificadores porque los pesos mayores van a los clasificadores que dieron mal rendimiento. Tras toda las predicciones de los clasificadores se escoge un resultado final mediante una elección o votación en la que los clasificadores que rindieron mejor tiene mayor peso frente a los que rindieron peor. Obteniéndose las predicciones finales de este modo. Gran parte del conocimiento teórico adquirido del boosting y de este algoritmo en específico es gracias a este artículo de Dragos D. Margineantu y Thomas G. Dietterich [23].

3.1.6. GradientBoost

El **GradientBoost** funciona de manera similar al AdaBoost solo que tras cada clasificador lo que se mide es el gradiente para ajustar los parámetros de ese modelo. El gradiente indica la dirección y magnitud del máximo cambio positivo de la función en el que esté, pues teniendo ese

gradiente calculado se actualizan los parámetros en el sentido opuesto para minimizar la función objetivo. Se sigue este proceso hasta que el gradiente es cercano a 0 o finalicen las iteraciones.

Gran parte del conocimiento teórico adquirido del boosting y de este algoritmo en específico es gracias a este artículo de Alexey Natekin y Alois Knoll [23].

3.1.7. Validación

Para un modelo f que realiza una tarea de regresión, su rendimiento sobre un conjunto de ejemplos \mathcal{D} (con $|\mathcal{D}|$ ejemplos en total) se mide mediante alguna función que compare los valores predichos con los valores correctos. En la competición se utiliza como métrica el MAE, pero en este trabajo las métricas adicionales que se utilizarán son el MSE y el MAPE, las cuales voy a definir:

El error absoluto medio (MAE, mean absolute error, en inglés) calcula el promedio del error cometido por el valor predicho con respecto al valor correcto para cada ejemplo. El error se calcula como el valor absoluto de la diferencia, ya que no nos interesa si es por exceso o por defecto.

$$\text{MAE} = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, y) \in \mathcal{D}} |\hat{y} - y| \quad \text{donde} \quad \hat{y} = f(\mathbf{x})$$

El **error cuadrático medio** (MSE, mean squared error, en inglés) calcula el promedio del error cometido por el valor predicho con respecto al valor correcto para cada ejemplo. El error se calcula como la diferencia al cuadrado, obteniéndose por tanto una función diferenciable más fácil de optimizar matemáticamente que el MAE. Por contra, y al contrario de lo que ocurre con el MAE, esta función penaliza los errores grandes mucho más que los errores pequeños. Esto quiere decir que basta con que en un ejemplo el modelo proporcione una respuesta lejos de la correcta para que el error se incremente en exceso.

$$\text{MSE} = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, y) \in \mathcal{D}} (\hat{y} - y)^2 \quad \text{donde} \quad \hat{y} = f(\mathbf{x})$$

El **error porcentual absoluto medio** (MAPE, mean absolute percentage error, en inglés) es una métrica comúnmente utilizada para evaluar modelos de regresión, especialmente cuando se desea interpretar el error en términos de porcentaje respecto al valor real. Se calcula como el promedio del valor absoluto de los errores porcentuales para cada ejemplo:

$$\text{MAPE} = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, y) \in \mathcal{D}} \left| \frac{\hat{y} - y}{y} \right| \times 100 \quad \text{donde} \quad \hat{y} = f(\mathbf{x}) \quad \text{es la predicción del modelo para } (\mathbf{x})$$

y (y) es el valor real correspondiente

Además de las métricas de rendimiento sobre el conjunto de entrenamiento, lo que más nos interesa a la hora de validar es medir el rendimiento sobre los datos nuevos. Para ello aplicaremos dos técnicas diferentes:

La **validación por retención** (*holdout validation* en inglés) es una metodología que aborda las dificultades en el aprendizaje automático al dividir el conjunto de ejemplos conocidos en dos subconjuntos: entrenamiento y prueba. En el subconjunto de entrenamiento se construye el modelo, mientras que en el subconjunto de prueba se evalúa su rendimiento utilizando una medida seleccionada. Los ejemplos en el subconjunto de prueba se mantienen retenidos hasta que se completa la construcción del modelo. La separación de los conjuntos se realiza típicamente de manera aleatoria, reservando generalmente entre un 20 % y un 30 % de los ejemplos para prueba. Es crucial mantener las proporciones de clases en ambos conjuntos, lo cual se logra mediante un muestreo estratificado durante la distribución aleatoria.

La **validación cruzada** con k pliegues (*k-fold cross validation* en inglés) es una alternativa común al holdout validation. Consiste en dividir el conjunto de ejemplos en k subconjuntos (llamados pliegues). Esta subdivisión se realiza de manera aleatoria, utilizando muestreo estratificado para mantener las proporciones de clases. Para cada pliegue:

- Se separan los ejemplos que no pertenecen al pliegue y los que sí pertenecen.
- Se entrena un modelo utilizando los ejemplos que no pertenecen al pliegue.
- Se evalúa el rendimiento del modelo utilizando los ejemplos del pliegue como conjunto de prueba.

Este proceso se repite k veces, utilizando cada pliegue como conjunto de prueba una vez. Finalmente, el método calcula la media de las k estimaciones obtenidas, proporcionando así una medida más robusta del rendimiento del modelo frente a variaciones en los datos de entrenamiento y prueba. A continuación se muestra la figura 3.1 que resume el proceso.

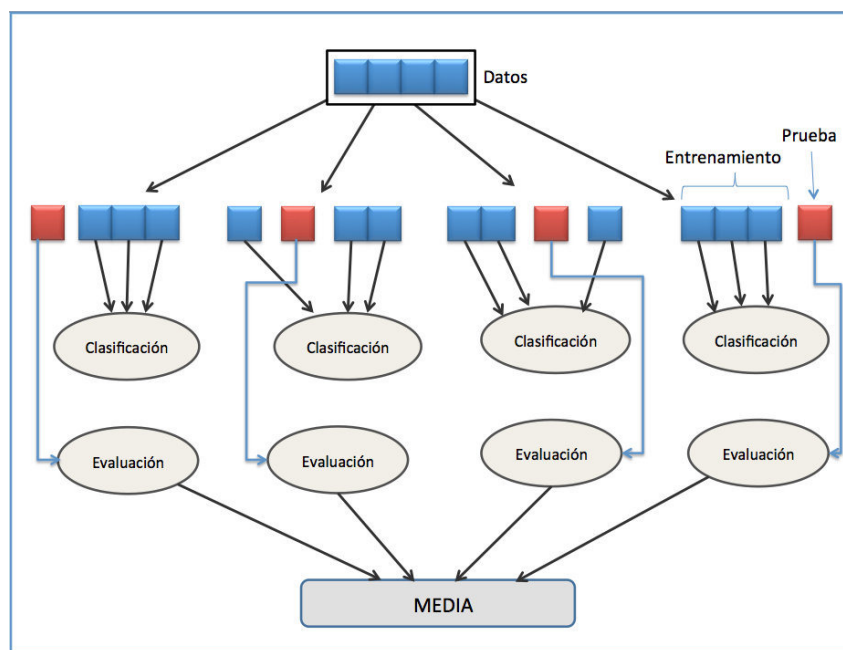


Figura 3.1: Proceso de Validación Cruzada

3.1.8. Optimización Bayesiana

Por último, en lo que respecta a los modelos, es importante tener en cuenta que deben optimizarse de sus hiperparámetros a fin de lograr el mejor rendimiento posible, por lo tanto, tendré que optimizar esos hiperparámetros. La **optimización bayesiana** es una técnica que se utiliza para encontrar la mejor configuración posible para una función dada, es particularmente útil cuando se trata de funciones costosas de evaluar. Hay tres conceptos clave en la optimización bayesiana; el modelo probabilístico, normalmente es un proceso gaussiano, para modelar la función objetivo desconocida, la función de adquisición la cual es la encargada de medir cuanta exploración o explotación se realiza para maximizar la ganancia y el propio concepto de iteración ya que la optimización bayesiana sucede por iteraciones. El proceso de optimización bayesiana se muestra en la figura 3.2.

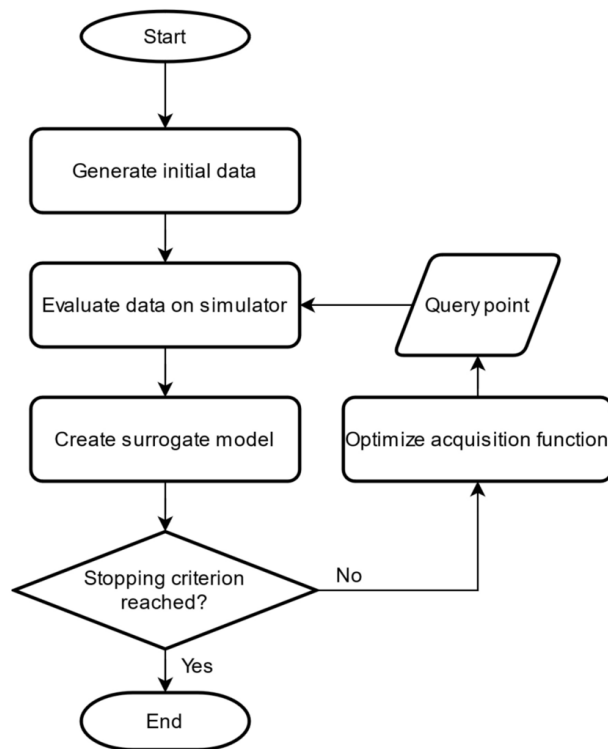


Figura 3.2: Proceso de Optimización Bayesiana

La estrategia se basa en tratar la función objetivo como una función aleatoria y en colocar una distribución a priori sobre la función objetivo que refleje un conocimiento previo sobre cómo se comporta. Luego, a medida que se van adquiriendo más datos sobre la función, se actualiza la distribución a priori en una distribución posterior, esta se utiliza para construir una función de adquisición.

Para información mucho más desarrollada recomiendo el artículo Eric Brochu [24].

Tras todo el fundamento teórico explicado se procede a describir el desarrollo y seguimiento de las iteraciones, las cuáles constan de reunión con el tutor, avances del desarrollo y seguimiento de riesgos e incidencias. Todas las actas de reunión se encuentran en la parte V Anexo.

3.2. 1º Iteración

Se presenta la primera iteración de la investigación.

3.2.1. Desarrollo

En esta primera iteración se ha realizado un proceso completo de entrenamiento y predicción en un único script.py. Este script hacía 4 procesos hasta generar las métricas de rendimiento:

- **Procesamiento de Datos:** Tras configurar los datos en crudo descargados de la aplicación y alojarlos en /data/kaggle/input, se procede a crear el dataframe con la librería Pandas [12]. Este primer dataframe consta de:
 - *volcan-id*: El id de los volcanes los cuales obtenemos del propio nombre de los archivos csv, estos volcanes serán cada fila del dataframe. En esta sección 3.1.2 se describen los datos de kaggle de manera detallada.
 - *media-sensorX*: La media del sensor X de ese volcán en concreto donde X es el número del sensor (del 1 al 10) del que se está haciendo la media, por lo que habrá un total de 10 *media-sensorX*. Como se puede deducir al hacer la media de las columnas, estamos rompiendo la temporalidad de los datos originales. Esto es algo que mejoraremos en las próximas iteraciones.
 - *desv-sensorX*: Lo mismo que la media pero aplicando la desviación típica a la columna, en total tendremos otras 10 columnas de desviación típica, una por sensor.
 - *time-to-eruption*: El valor objetivo el cual obtenemos gracias al archivo train.csv, en esta sección 3.1.2 se explica, haciendo corresponder cada id con su *time-to-eruption* (este atributo se abrevia en tte durante el resto del documento).
- **División de datos:** Los datos se dividen utilizando el siguiente método expuesto en esta extracción de código 3.1 de SKlearn.

Listing 3.1: Método de validación por retención

```
train_test_split(X, y, test_size=0.2, random_state=42)
```

En esta primera iteración se sigue por tanto una validación por retención siendo un 80 % los datos para entrenamiento y un 20 % para test. X será el conjunto de características e y el conjunto de etiquetas. El parámetro random-state de todos los métodos será 42 para garantizar reproducibilidad.

- **Entrenamiento de los modelos:** Los modelos KNN, Decision Tree y Random Forest se implementan con la librería SKlearn. Para esta iteración se utilizarán los hiperparámetros genéricos para cada modelo.

Listing 3.2: Método de KNN

```
knn = KNeighborsRegressor(n_neighbors=k)
knn.fit(X,y)
return knn
```

Siendo $k=5$. El método fit de SKlearn es el que entrena el modelo propiamente. Devolvemos el modelo entrenado para validar el rendimiento después. Lo mismo con los otros dos modelos:

Listing 3.3: Método de Decision Tree

```
dt = DecisionTreeRegressor(random_state=42, max_depth=10,
    min_samples_split=2)
dt.fit(X,y)
return dt
```

El hiperparámetro max-depth indica la máxima profundidad que alcanzará el árbol de decisión. El hiperparámetro min-samples-split establece el número mínimo de muestras necesarias para dividir un nodo.

Listing 3.4: Método de Random Forest

```
rf = RandomForestRegressor(n_estimators=100, random_state=42,
    max_depth=5, min_samples_split=2, n_jobs=-1)
rf.fit(X,y)
return rf
```

El nuevo hiperparámetro que se introduce en este modelo es el n-estimator el cuál es el número de árboles que conformarán el bosque completo. El hiperparámetro n-jobs=-1 es para usar todos los núcleos de CPU disponibles para acelerar el entrenamiento.

- **Validación:** Tras entrenar todos los modelos procedemos a medir su rendimiento utilizando tres métricas (MSE, MAE y MAPE):

Listing 3.5: Métricas

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
mape = calculate_mape(y_test, y_pred)
```

Listing 3.6: Método del cálculo del MAPE

```
def calculate_mape(y_true, y_pred):
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
```

Como podemos observar tanto para el MAE como para el MSE utilizamos los métodos que ofrece SKlearn para calcular las métricas del modelo entrenado, mientras que para el MAPE lo hacemos de manera manual. Donde y-pred son los valores de tte predichos por el modelo e y-test los valores de tte reales.

Al final de la iteración obtenemos estos resultados mostrados en este cuadro 3.1:

Modelo	MAE	MSE	MAPE
KNN	1.03e7	1.65e14	243.66
DT	1.02e7	1.57e14	252.54
RF	1.00e7	1.43e14	240.66

Cuadro 3.1: 1º Iteración (Validación por Retención)

Estás métricas serán el punto de partida a mejorar en las siguientes iteraciones, que como veremos, así será.

3.2.2. Seguimiento

En el siguiente cuadro 3.2 se muestran las incidencias de la iteración.

Cuadro 3.2: Tareas Iteración 1º

Código - EDT	Nombre	Incidencia	Solución
2.2.1	Crear Proyecto	Sin incidencias	–
2.2.2	Procesamiento de Datos 1	Las id de los volcanes eran números del 1 hasta las X filas del archivo, cuando debían ser la id del nombre del archivo	Crear una función auxiliar que leyera el nombre del archivo csv y asignarlo a una fila.
2.2.3	Modelos 1	Los modelos tardaban mucho tiempo en ejecutarse.	Separar los modelos, el procesamiento y la validación en archivos separados.
2.2.4	Validación 1	Sin incidencias	–
2.2.5	Gráficos 1	Errores con la librería matplotlib a la hora de leer datos del dataset	Utilizar en su lugar la librería Seaborn, la cual es más versátil.
2.2.6	Revisión de Iteración 1	Sin incidencias	–

3.2.3. Riesgos

Finalmente, mencionamos los riesgos de la iteración:

- RISK-001: Dificultad para realizar las actividades usando las nuevas tecnologías sin experiencia previa.
 - El uso de nuevas tecnologías y adaptación retraso el final de la iteración, puesto que en las librerías que utilicé, excepto Pandas, no tengo ningún tipo de experiencia.
- RISK-013: No alcanzar las fechas de los hitos.
 - El final de iteración se retraso en parte por lo mencionado previamente, sin embargo es asumible el retraso ya que solo ha sido 3 días.

3.3. 2º Iteración

Se presenta la segunda iteración de la investigación.

3.3.1. Desarrollo

En esta iteración cambié la arquitectura del software, ya no es un solo archivo donde añadir los métodos. El software se organiza como se ve en la siguiente figura 3.3:

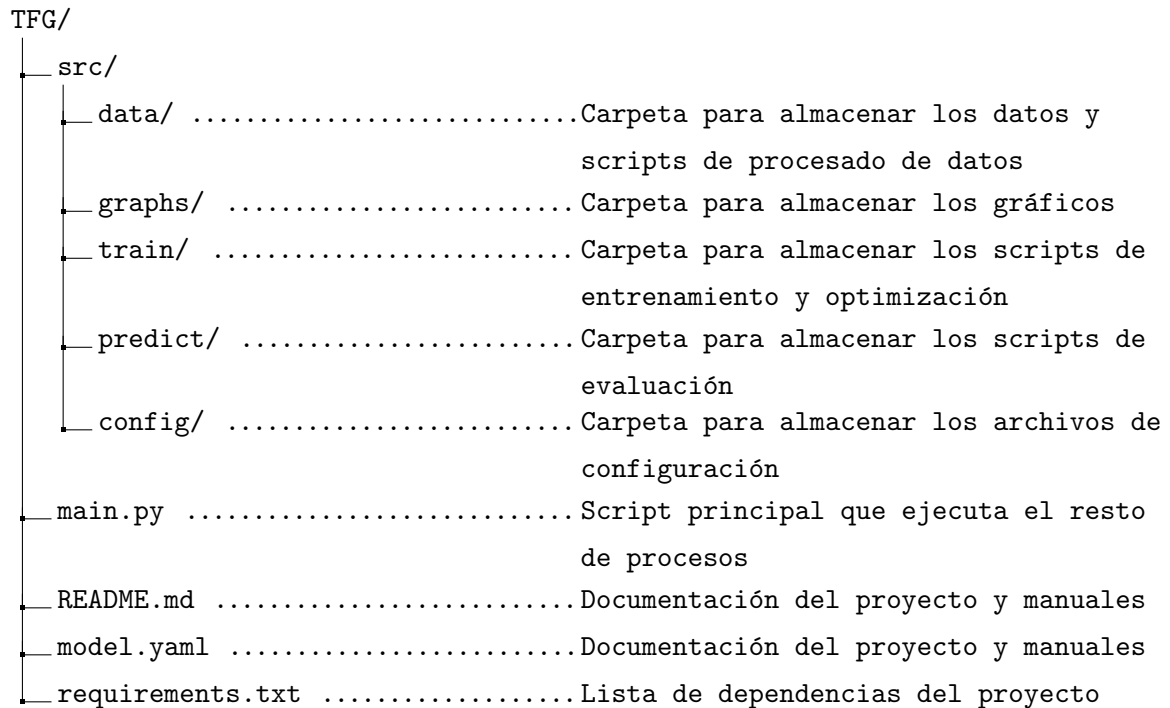


Figura 3.3: Estructura del software de experimentación

Esta arquitectura permite automatizar procesos y omitir algunos como puede ser el de procesamiento de datos si ya los tenemos procesados previamente. He decidido hacerla de esta manera para que el proyecto siga un cierto estándar de calidad además que facilitará el desarrollo de nuevas implementaciones a *posteriori* teniendo solo que añadir el script o función al modulo correspondiente y llamarlo en el main.py. El archivo model.yaml es un archivo de configuración de los parámetros de los modelos cada modelo tiene esta estructura:

```
models:
  - name:
    module:
    function:
    params:
```

Este procesamiento de datos de hecho se ha mejorado conforme a la anterior puesto que ahora se incluyeron 2 nuevas columnas de datos:

- *max-global*: El máximo se calcula a partir de todos los sensores de ese volcán.

- *min-global*: El mínimo se calcula a partir de todos los sensores de ese volcán.

Asimismo implementé la validación cruzada mediante *k*-pliegues. Para crear estos pliegues utilizo SKlearn.

Listing 3.7: KFold

```
kf = KFold(n_splits=5)
kf.split(X)
```

Con este *kf* al usar el método `split(X)`, *X* recordemos que son las características, hago los pliegues del dataframe los cuales guardo en una lista de listas para que al entrenar los modelos, los pliegues se entrenen de uno en uno, para que al final con todos los pliegues entrenados obtenga las métricas MSE, MAE y MAPE de cada pliegue y como resultado final se hace la media de todos los pliegues obteniendo así los siguientes resultados mostrados en este cuadro 3.3:

Modelo	MAE	MSE	MAPE
KNN	8.11e6	1.09e14	211.96
DT	9.87e6	1.43e14	365.09
RF	9.60e6	1.31e14	336.70

Cuadro 3.3: 2^o Iteración (Validación Cruzada)

3.3.2. Seguimiento

En el siguiente cuadro 3.4 se muestran las incidencias de la iteración.

Cuadro 3.4: Tareas Iteración 2^o

Código - EDT	Nombre	Incidencia	Solución
2.3.1	Diseño de arquitectura	El main.py daba problemas durante la ejecución y con la importación de los módulos.	Importar importlib en el main.py.

Continúa en la próxima página

(Continúa en la siguiente página)

Código - EDT	Nombre	Incidencia	Solución
2.3.2	Procesamiento de Datos 2	El dataset resultado era demasiado grande ya que tenía más de 200 columnas.	Se procede a obtener las medias y desviaciones típicas cada 600 filas (6 seg) en vez de cada 50 filas (0.5 seg).
2.3.3	Modelos 2	Sin incidencias	–
2.3.4	Validación 2	La validación cruzada no calculaba las métricas correctamente	Dividir en k-pliegues y luego a los modelos y la validación pasarle esos pliegues a modo de listas.
2.3.5	Gráficos 2	Sin incidencias	–
2.3.6	Revisión de Iteración 2	Sin incidencias	–

3.3.3. Riesgos

Finalmente, mencionamos los riesgos de la iteración:

- RISK-013: No alcanzar las fechas de los hitos.
 - El final de iteración se retraso mucho más de lo esperado, dependiendo del transcurso de la siguiente iteración será necesaria una replanificación y cambio de fechas en los hitos.

3.4. 3º Iteración

Se presenta la tercera iteración de la investigación.

3.4.1. Desarrollo

En esta iteración se ha implementado una nueva manera de procesar y extraer características, utilizando la librería tsfresh [25]. Esta librería te permite extraer características de las series temporales sin perder la temporalidad como ya habíamos sacrificado en las anteriores iteraciones. Para ello tenemos tres clases diferentes con las que podemos extraer características que siguen diferentes estrategias que se explican detalladamente en su documentación [26]:

Listing 3.8: Extracción de características con tsfresh

```
settings = MinimalFCParameters()
from tsfresh.feature_extraction import extract_features
extract_features(df, default_fc_parameters=settings)
```

En este caso usamos la clase `MinimalFCParameters` que como su nombre indica busca las mínimas características posibles. Las otras dos clases (`ComprehensiveFCParameters` y `EfficientFCParameters`) buscan y extraen de manera exhaustiva características en los datos en crudo. Aunque la clase `EfficientFCParameters` evita las características que requieren una mayor capacidad de procesamiento, de igual manera la extracción de las características era demasiado lenta y costosa, por lo que se ha decidido utilizar la clase descrita en el código anterior 3.8. Estas fueron las propiedades extraídas:

- `volcan_id`: Identificador del volcán.
- `media_sensor__sum_values`: Suma de los valores del sensor.
- `media_sensor__median`: Mediana de los valores del sensor.
- `media_sensor__mean`: Media de los valores del sensor.
- `media_sensor__length`: Longitud de los valores del sensor.
- `media_sensor__standard_deviation`: Desviación estándar de los valores del sensor.
- `media_sensor__variance`: Varianza de los valores del sensor.
- `media_sensor__root_mean_square`: Raíz cuadrada de la media cuadrática de los valores del sensor.
- `media_sensor__maximum`: Valor máximo de los valores del sensor.
- `media_sensor__absolute_maximum`: Máximo absoluto de los valores del sensor.
- `media_sensor__minimum`: Valor mínimo de los valores del sensor.

En cuanto a modelos y validación no hay ninguna novedad puesto que la validación cruzada mejoró las métricas frente a la validación por retención. Dicho esto los resultados de esta iteración se observan en la siguiente tabla 3.5.

Modelo	MAE	MSE	MAPE
KNN	9.94e6	1.48e14	354.57
DT	1.02e7	1.49e14	320.81
RF	1.01e7	1.43e14	339.59

Cuadro 3.5: 3º Iteración (Extracción de Características con tsfresh)

3.4.2. Seguimiento

En el siguiente cuadro 3.6 se muestran las incidencias de la iteración.

Cuadro 3.6: Tareas Iteración 3º

Código - EDT	Nombre	Incidencia	Solución
2.4.1	Utilizar tsfresh	La clase elegida para la extracción no fue la más óptima teniendo en cuenta el tiempo necesario para extraer las características.	Cambiar de la clase ComprehensiveFCParameters a MinimalFCParameters.
2.4.2	Procesamiento de Datos 3	Dificultades en la automatización del procesamiento, al tener el módulo tsfresh separado del procesamiento	Hacer una distinción en el main haciendo que la función de procesamiento reciba como parámetro una cadena con el modo de procesamiento deseado.
2.4.3	Modelos 3	Sin incidencias	—
2.4.4	Validación 3	Sin incidencias	—
2.4.5	Gráficos 3	Sin incidencias	—
2.4.6	Revisión de Iteración 3	Sin incidencias	—

3.4.3. Riesgos

Finalmente, mencionamos los riesgos de la iteración:

- RISK-001: Dificultad para realizar las actividades usando las nuevas tecnologías sin experiencia previa.

- El uso `tsfresh` y su entendimiento fue más complejo de lo esperado. Una posible solución hubiera sido añadir la librería al estudio previo.
- RISK-013: No alcanzar las fechas de los hitos.
- La replanificación es inevitable y no se puede asumir llegar a tiempo a la primera convocatoria del TFG.

3.5. 4º Iteración

Se presenta la cuarta iteración de la investigación.

3.5.1. Desarrollo

En esta iteración se ha desarrollado un nuevo dataframe aplicado ingeniería de características de manera manual sin usar `tsfresh` y sin sacrificar la temporalidad totalmente. Lo que se ha hecho es calcular la media de todos los sensores (es decir de las 10 columnas), así obtenemos un solo valor a partir de 10 valores en la misma fila, eso significa que pertenecen al mismo intervalo de tiempo, no se pierde temporalidad. La problemática que surge con esto es que se generarían demasiadas columnas en el nuevo dataframe por lo que para solucionar esto se hará la media de las 10 columnas y cada 600 filas (6 segundos). Al final, obtenemos unas 100 columnas de medias de sensores “globales” y añadido a esto haremos el mismo proceso con las desviaciones típicas. Sin olvidar de incluir en ese proceso el cálculo del máximo global en esa ventana de 10 columnas y 600 filas, el mínimo global y los cruces por 0.

Se han integrado dos nuevos modelos al software los cuales se definen en sus respectivas secciones 3.1.5 y 3.1.6. Hablamos del Adaboost y el Gradientboost, ambos integrados mediante SKlearn:

Listing 3.9: Modelo de AdaBoosting

```
estimator = DecisionTreeRegressor(max_depth=5)
ada_model = AdaBoostRegressor(estimator=estimator, n_estimators=100,
                              learning_rate=1.0, loss='exponential', random_state= 42 )
```

El hiperparámetro `estimator` especifica el estimador base que será utilizado por AdaBoost en este caso será el árbol de decisión con una profundidad máxima de 5, ya que, si aumento demasiado la profundidad puede existir sobreajuste y se ralentizaría demasiado el entrenamiento por la tardanza del proceso al tener mayor profundidad. El hiperparámetro `learning-rate` pondera la

contribución de cada estimador en el conjunto. El hiperparámetro `loss` es la función de pérdida que por defecto en AdaBoost es exponencial.

Listing 3.10: Modelo de GradientBoosting

```
gboost_model = GradientBoostingRegressor(max_depth=5, n_estimators=100,
    learning_rate=1.0, loss='squared_error', random_state= random_state )
```

El hiperparámetro `loss` es la función de pérdida pero en este caso la predeterminada en el GradientBoost es el error cuadrático.

Se ha implementado la optimización bayesiana definida en esta sección 3.1.8, gracias a esta librería `bayes-opt` [27]. Se ha definido un rango de búsqueda de parámetros bastante amplio para los modelos. Luego con los parámetros definidos se realiza la optimización de la función:

Listing 3.11: Optimización Bayesiana

```
dt_optimizer = BayesianOptimization(
    f=params_dt,
    pbounds=dt_param_bounds,
    random_state=42,
)
dt_optimizer.maximize(init_points=10, n_iter=40)
```

La función objetivo es `params_dt`, es decir, la que se va a maximizar durante la optimización. `pbounds` es el rango de los parámetros que mencioné previamente y dentro del *maximize* el número de iteraciones será 40 donde al inicio de la exploración inicial 10 de las iteraciones serán puntos aleatorios. Tras todas las mejoras en los resultados son esperables y los podemos observar en el siguiente cuadro 3.7:

Modelo	MAE	MSE	MAPE
KNN	8.11e6	1.09e14	211.96
DT	3.01e6	5.39e13	68.31
RF	7.38e6	8.07e13	243.91
ADABOOST	9.30e6	1.08e14	279.19
GBOOST	2.74e6	5.55e13	78.93

Cuadro 3.7: 4^o Iteración (Optimización Bayesiana)

3.5.2. Seguimiento

En el siguiente cuadro 3.8 se muestran las incidencias de la iteración.

Cuadro 3.8: Tareas Iteración 4^o

Código - EDT	Nombre	Incidencia	Solución
2.5.1	Utilizar optimización Bayesiana	Los rangos de los parámetros estaban sobreajustados, es decir, no permitían el correcto funcionamiento de la optimización	Ampliar los rangos y aumentar las iteraciones para que la optimización buscara a priori más posibilidades.
2.5.2	Procesamiento de Datos 4	Sin incidencias	—
2.5.3	Modelos 4	Sin incidencias	—
2.5.4	Validación 4	Sin incidencias	—
2.5.5	Gráficos 4	Sin incidencias	—
2.5.6	Revisión de Iteración 4	Sin incidencias	—

3.5.3. Riesgos

Finalmente, mencionamos los riesgos de la iteración:

- RISK-002: Falta de potencia de procesamiento o excesiva tardanza en los procesos del software de experimentación.
 - Gran parte del tiempo empleado en esta iteración ha sido en la espera de los resultados óptimos, los ajustes y en las iteraciones de la optimización bayesiana. Este tiempo "perdido" de espera viene relacionado directamente con la tardanza de los procesos.

3.6. 5^o Iteración

Se presenta la última iteración de la investigación.

3.6.1. Desarrollo

En esta iteración se vuelve a utilizar el dataframe creado a partir de tsfresh pero se le aplicará un filtro de selección de características mediante SKlearn para intentar mejorar las métricas con el dataframe de tsfresh. Este es el enlace a la documentación [28]. Para la selección de características en este trabajo se utilizarán diversas técnicas y se medirán su rendimiento, para escoger la mejor entre ellas. Aunque la problemática que encontramos aquí realmente es que el propio dataframe base no tiene muchas características extraídas como se vio en la sección 3.4 referente a la tercera iteración. Así que, puede que la selección final de características no sea tan interesante como el dataframe obtenido en la cuarta iteración, como bien se explica en esta sección 3.5.

- La selección mediante varianza: Esta selección más que escoger realmente lo que hace es descartar todas las características que no llegue a un cierto umbral preestablecido de varianza. Sirve para eliminar normalmente características que sean constantes.
- La selección mediante filtro: A partir de una función de estrategia que determina un criterio específico se examinan las características individualmente y luego se seleccionan las k -mejores o el porcentaje de mejores características, según se utilice KBest o SelectPercentile y se defina ese k o porcentaje.
- La selección mediante modelo: Estas utilizan un modelo de aprendizaje automático para seleccionar las mejores características. Aunque puedan ser las más interesantes

Tras probar todas las selecciones las que mejor resultado dieron fueron las de filtro haciendo una selección por percentil utilizando la estrategia “mutual-info“, pues es más versátil que la estrategia “f-regression“, pues está detecta solo relaciones lineales entre características. Además creo que es interesante que previamente a utilizar la selección de filtro de aplique la selección mediante varianza para eliminar datos constantes o que aporten muy poco. Las características escogidas fueron:

- `volcan_id`: Identificador del volcán.
- `media_sensor__sum_values`: Suma de los valores del sensor.
- `media_sensor__median`: Mediana de los valores del sensor.
- `media_sensor__mean`: Media de los valores del sensor.
- `media_sensor__standard_deviation`: Desviación estándar de los valores del sensor.

Tras obtener el nuevo dataframe los resultados con este fueron los que se muestran en este cuadro 3.9:

Modelo	MAE	MSE	MAPE
KNN	1.03e7	1.57e14	391.42
DT	4.74e6	7.75e13	143.72
RF	1.14e7	1.75e14	445.66
ADABOOST	1.15e7	1.77e14	449.76
GBOOST	3.15e6	7.51e13	72.05

Cuadro 3.9: 5ª Iteración (Selección de Características)

3.6.2. Clasificación en la competición

La posición que he conseguido tras registrarme en la competición y hacer un *late submission* ha sido la **584** (la competición terminó mucho antes de iniciar este TFG). En el siguiente cuadro 3.4 aparece la score en la clasificación.

Submission and Description		Private Score ⓘ	Public Score ⓘ
 submission.csv		13669821	13313044
Complete (after deadline)			

Figura 3.4: Clasificación de Kaggle

A partir de la *late submission* se ha generado este gráfico 3.5 para observar las distribuciones generales.

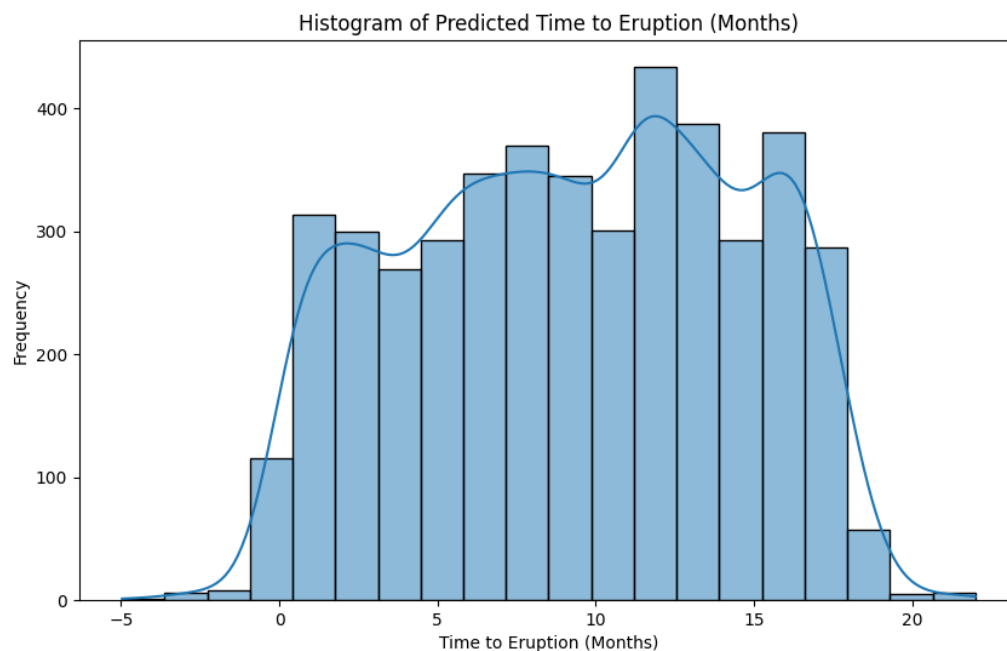


Figura 3.5: Histograma de Frecuencias de tte en meses

3.6.3. Seguimiento

En el siguiente cuadro 3.10 se muestran las incidencias de la iteración.

Cuadro 3.10: Tareas Iteración 5^o

Código - EDT	Nombre	Incidencia	Solución
2.6.1	Procesamiento de Datos 5	Sin incidencias	—
2.6.2	Modelos 5	Sin incidencias	—
2.6.3	Validación 5	Sin incidencias	—
2.6.4	Gráficos 5	Algunos gráficos se perdieron o se corrigieron desde las iteraciones anteriores	Repetir los gráficos faltantes, revisando el historial del repositorio.
2.6.5	Revisión de Iteración 5	Sin incidencias	—

(Continúa en la siguiente página)

Código - EDT	Nombre	Incidencia	Solución
--------------	--------	------------	----------

Continúa en la próxima página

2.6.6	Revisión global de la experimentación	Sin incidencias	—
-------	---------------------------------------	-----------------	---

3.6.4. Riesgos

Finalmente, mencionamos los riesgos de la iteración:

- RISK-002: Inconsistencias y errores en la generación de reportes y gráficos.
 - Como se ha comentado en la incidencia algunos gráficos eran inconsistentes u obsoletos y se han generado de nuevo.

CAPÍTULO 4

Aplicación Web

En esta sección se describe el desarrollo, desviaciones y manual de la aplicación final. La aplicación se desarrolló en un mismo proyecto separando Backend y Frontend. Lo primero que se desarrolló fue el MVP (Mapa, Usuarios y Detalles de Volcán). Luego pasó a estilizar la aplicación y testear la aplicación. Finalmente se realizó el despliegue en Render descrito en esta sección 4.4.

4.1. Desarrollo

En esta sección se explica el desarrollo de la aplicación en cuanto a decisiones tomadas, seguimiento y monitoreo de riesgos.

4.1.1. Justificación

Durante el desarrollo de la aplicación se han tomado una serie de decisiones:

- La arquitectura se divide en backend y frontend. El backend decidí hacerlo en Django ya que durante la carrera he trabajado con ese framework en varios de los proyectos de desarrollo de aplicaciones, por lo que tengo cierta experiencia y soltura. Además la base de datos era bastante pequeña y no era necesario muchas consultas y escrituras así que con el dbsqlite de Django, me parecía la opción de backend más razonable. En cuanto a frontend las opciones eran tres:

- Utilizar las templates de Django y aplicar los estilos desde el mismo framework.
- Utilizar Angular. Este es el enlace a la pagina oficial [29].
- Utilizar Vue.js.

La opción ganadora fue Vue.js pues además de tener experiencia con el propio framework en el propio proyecto donde lo utilice desarrolle un mapa con marcadores interactivos en una aplicación web, así que partía con cierta ventaja si utilizaba este framework. La opción de las templates de Django podría ser tedioso a la larga al tener todo unido y en angular no tengo ningún tipo de experiencia.

- La base de datos como he mencionado se usara la dbsqlite de Django. Los modelos de datos serán el modelo del Volcán y el de Usuario:

Listing 4.1: Modelos de Django

```
class Volcano(models.Model):
    name = models.CharField(max_length=100)
    location = models.CharField(max_length=100)
    country = models.CharField(max_length=100)
    latitude = models.CharField(max_length=50)
    longitude = models.CharField(max_length=50)
    height = models.FloatField()
    eruption_time = models.FloatField()
    def __str__(self):
        return self.name

class UserProfile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.user.username
```

Los datos los obtengo a partir de las predicciones obtenidas por el mejor modelo de la experimentación. A partir de tener esos modelos necesitaba los datos de las características de los volcanes. Como no podía obtenerlos desde el INGV porque cada volcán era un id (anónimo) no podía determinar que volcán era cada id, así que decidí asignar aleatoriamente a partir de un csv con las características que necesitaba [30]. Por lo que la aplicación no podía reflejar datos realistas en cuanto a erupción por esa problemática inicial del ID. Teniendo ese csv con todas las características solo habría que importarlo a la base de datos.

- La API de la aplicación es muy sencilla puesto que las rutas de usuario las gestiono mediante y las de volcán son estas rutas:

Listing 4.2: Urls de volcanes

```
path('volcanoes/', VolcanoListView.as_view(), name='volcano-list'),
path('volcanoes/<int:id>/', VolcanoDetailView.as_view(), name='volcano-
-detail'),
```

Las arquitectura del backend se puede ver resumido en esta figura 4.1 obtenida de mdn web docs [31]:

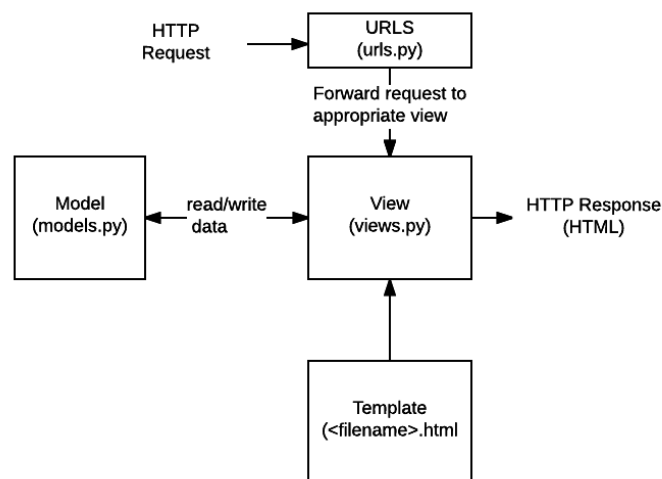


Figura 4.1: Arquitectura de Django

Las arquitectura del frontend se puede ver resumido en esta figura 4.2:

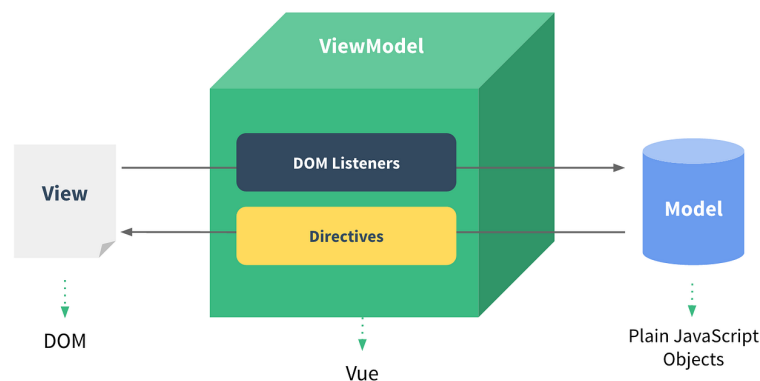


Figura 4.2: Arquitectura de Vue.js

Para un mejor entendimiento de la aplicación he diseñado un diagrama de componentes en este cuadro 4.3:

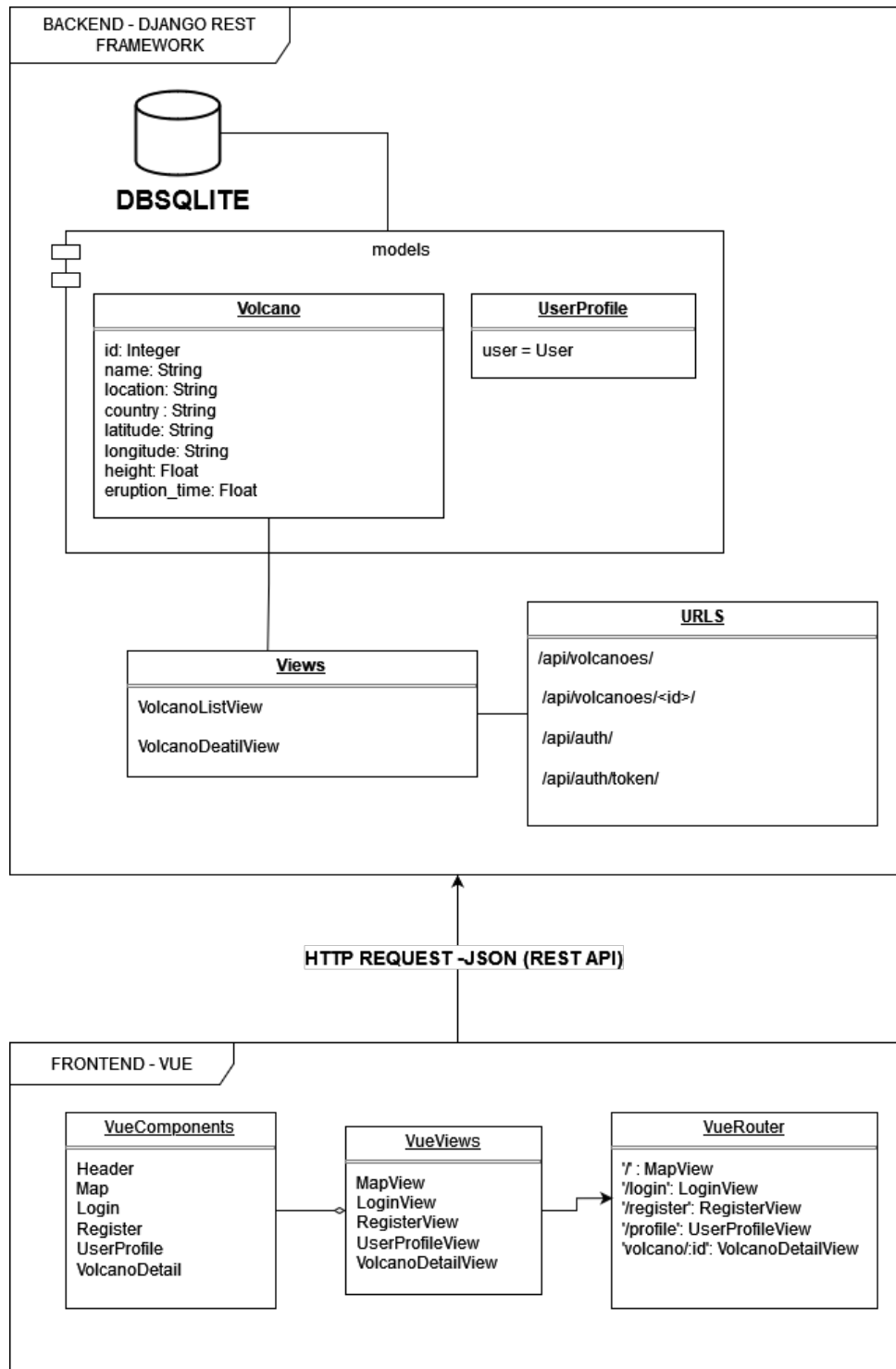


Figura 4.3: Diagrama de Componentes

4.1.2. Seguimiento

En esta sección se desarrollo el seguimiento de la iteración.

Cuadro 4.1: Tareas Aplicación Web

Código - EDT	Nombre	Incidencia	Solución
2.7.1	Crear el Proyecto	Problemas para enlazar los repositorios de backend y frontend por separado	Se hace backend y frontend en el mismo repositorio y se desplegará por separado cada parte.
2.7.2	Mapa	El mapa al hacer zoom de vez en cuando no cargaba correctamente o aparecía error de carga	Cargar el css de la librería leaflet separado de la importación de la propia librería y utilizar los métodos Mount y Unmount de Vuejs para controlar la carga del componente y evitar fallos en producción.
2.7.3	Vista de Detalles	Para acceder a esta vista desde el popup del marcador daba errores con el router de vue	Se accede a la vista de detalles a partir de la
2.7.4	Usuarios	No hay respuesta clara cuando inicias sesión y/o registras un usuario	Hacer un router-push a la vista del mapa y mostrar un mensaje.
2.7.5	Volcanes Favoritos	El botón para añadir los volcanes desde el popup no aparecía correctamente	Añadir el botón en la vista de detalles del volcán.
2.7.6	Filtro de Búsqueda	El filtro no se ha conseguido realizar de la manera en que se planificó	Se recorta del alcance de la aplicación
2.7.7	Testing	Problemas para medir correctamente la cobertura	Añadir la herramienta coverage al proyecto.

Continúa en la próxima página

(Continuación desde la anterior página)

Código - EDT	Nombre	Incidencia	Solución
2.7.8	Revisión global y despliegue de la aplicación	Los archivos de configuración de render dieron problemas con los nombres y comandos de build	Lectura exhaustiva de la documentación de render y aplicar correcciones.

A continuación la cobertura del testing de la aplicación en la siguiente figura 4.4. Se han realizado test a los modelos y las vistas:

Name	Stmts	Miss	Cover
-----	-----	-----	-----
backend__init__.py	0	0	100%
backend\asgi.py	4	4	0%
backend\urls.py	3	0	100%
myapp__init__.py	0	0	100%
myapp\admin.py	1	0	100%
myapp\apps.py	4	0	100%
myapp\management__init__.py	0	0	100%
myapp\management\commands__init__.py	0	0	100%
myapp\management\commands\import_volcanoes.py	17	17	0%
myapp\models.py	16	2	88%
myapp\serializers.py	6	0	100%
myapp\tests.py	1	0	100%
myapp\urls.py	3	0	100%
myapp\views.py	18	0	100%
tests__init__.py	0	0	100%
tests\tests_models.py	18	0	100%
tests\tests_views.py	28	0	100%
-----	-----	-----	-----
TOTAL	119	23	81%

Figura 4.4: Cobertura de la App

4.1.3. Riesgos

Finalmente, mencionamos los riesgos durante el desarrollo de la web:

- RISK-005: Problemas de compatibilidad y rendimiento de la aplicación web en diferentes dispositivos y navegadores.

- La aplicación no tenía estilos responsive, en el móvil no se veían los botones de inicio de sesión.
- RISK-008: Falta de claridad en la definición de los requisitos y/o ambigüedad.
 - Algunas requisitos funcionales no especificaban de manera clara los criterios que cumplimentaban la funcionalidad. Se ha tomado la decisión de desarrollar la funcionalidad que cumpla los mínimos.

4.2. Manual de instalación

En esta sección se detalla el manual de instalación de ambos software.

4.2.1. Manual de instalación del software de experimentación

Antes de comenzar, asegúrese de tener instaladas las siguientes herramientas en su sistema:

- **Git**: para clonar el repositorio. [Página oficial](#).
- **Python 3.x**: el lenguaje de programación necesario. [Página oficial](#).
- **pip**: el instalador de paquetes de Python.

Tras revisar los requisitos previos continuamos con la instalación:

1. Ejecute el siguiente comando para clonar el repositorio desde GitHub en el directorio que desee:

```
git clone https://github.com/AitorRD/TFG-Vulcan-Prediction.git
```

2. Cree un entorno virtual utilizando **venv**:

```
python -m venv venv
```

3. Active el entorno virtual (Este paso no es necesario, pero se recomienda hacerlo):

- En Windows:

```
.\venv\Scripts\activate
```


- En macOS y Linux:

```
source venv/bin/activate
```

4. Ejecute el siguiente comando para instalar todas las dependencias listadas en el archivo `requirements.txt`:

```
pip install -r requirements.txt
```

5. Descargue los datos de la competición desde el sitio web de Kaggle.
6. Descomprima el archivo descargado en su sistema.
7. Cree el directorio necesario para almacenar los datos descomprimidos:

```
mkdir -p src/data/kaggle/input
```

8. Mueva los archivos descomprimidos al directorio creado:

```
mv ruta_del_archivo_descomprimido/* src/data/kaggle/input/
```

9. Asegúrese de que la estructura de directorios quede de la siguiente manera:

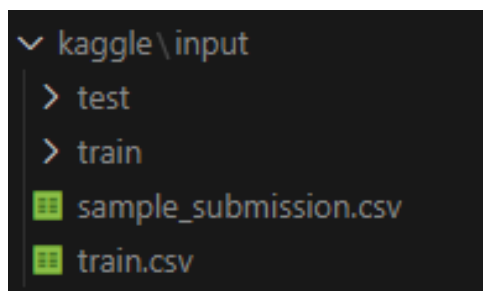


Figura 4.5: Estructura Datos de Kaggle

4.2.2. Manual de instalación de la aplicación web

Antes de comenzar, asegúrese de tener instaladas las siguientes herramientas en su sistema:

- **Git**: para clonar el repositorio. [Página oficial](#).
- **Python 3.x**: el lenguaje de programación necesario. [Página oficial](#).
- **pip**: el instalador de paquetes de Python.

- **Node.js y npm:** necesarios para la instalación del frontend. Página oficial.

Tras revisar los requisitos previos, continuamos con la instalación:

1. Ejecute el siguiente comando para clonar el repositorio desde GitHub en el directorio que desee:

```
git clone https://github.com/AitorRD/Volcano-map.git
```

2. Instalación del Backend

- a) Navegue al directorio del backend:

```
cd Volcano-map/backend
```

- b) Cree un entorno virtual utilizando **venv**(Este paso no es necesario, pero se recomienda hacerlo):

```
python -m venv venv
```

- c) Active el entorno virtual:

- En Windows:

```
.\venv\Scripts\activate
```

- En macOS y Linux:

```
source venv/bin/activate
```

- d) Instale las dependencias del backend listadas en el archivo **requirements.txt**:

```
pip install -r requirements.txt
```

- e) Realice las migraciones de la base de datos:

```
python manage.py makemigrations  
python manage.py migrate
```

- f) Cree un superusuario para acceder al panel de administración de Django:

```
python manage.py createsuperuser
```

- g) Inicie el servidor de desarrollo de Django:

```
python manage.py runserver
```

3. Instalación del Frontend

- a) Navegue al directorio del frontend:

```
cd Volcano-map/frontend
```

- b) Instale las dependencias del frontend utilizando npm:

```
npm install
```

- c) Inicie el servidor de desarrollo del frontend:

```
npm run serve
```

4.3. Manual de usuario

En esta sección se detallan los manuales de usuario de ambos software.

4.3.1. Manual de usuario del software de experimentación

Este manual está diseñado para guiarte a través del uso del script `main.py`, el cual realiza el procesamiento de datos, optimización y entrenamiento de modelos predictivos para el tiempo hasta la erupción de volcanes. A continuación, se describen los pasos y opciones disponibles al ejecutar el script.

1. Antes de ejecutar `main.py`, asegúrate de que tienes todas las dependencias necesarias instaladas y que tu entorno está configurado correctamente.
2. Para ejecutar el script principal y seguir los pasos de entrenamiento o predicción, ejecute:

```
python main.py
```

3. A continuación, siga las indicaciones en la consola:

- a) **Modo de datos:** Especifique si desea ejecutar en modo de entrenamiento (`train`) o prueba (`test`):

```
Do you want to run the model in train or test mode? (train/
test):
```

- b) **Procesar los datos:** Indique si desea procesar los datos crudos:

```
Do you want to process the raw data? (y/n):
```

- c) **Modo de procesamiento de datos:** Seleccione el modo de procesamiento de datos (MANUALFEATURES/TSFRESH):

```
Enter data processing mode (MANUALFEATURES/TSFRESH):
```

- d) **Optimización del modelo:** Indique si desea optimizar el modelo:

```
Do you want to run the model optimization? (y/n):
```

- e) **Tipo de división de datos:** Especifique el tipo de división de datos (CROSSVAL/HOLDOUT):

```
Enter the split type you want to use (CROSSVAL/HOLDOUT):
```

- f) **Modelo a usar:** Introduzca el nombre del modelo que desea usar (KNN/DT/RF/ADABOOST/GBOOST):

```
Enter the model you want to use (KNN/DT/RF/ADABOOST/GBOOST)
:
```

4. Los resultados se guardarán en un archivo CSV en el directorio `src/predict/results/`.
5. El nombre del archivo incluirá el nombre del modelo, el tipo de división de datos y el modo de procesamiento de datos, por ejemplo: `results_knn_crossval_manualfeatures_train.csv`.
6. Ejemplo de Uso: A continuación, se muestra un ejemplo de cómo se vería una sesión típica de ejecución del script:

```
Do you want to run the model in train or test mode? (train/test): train
Do you want to process the raw data? (y/n): y
Enter data processing mode (MANUALFEATURES/TSFRESH): MANUALFEATURES
----- PROCESSING DATA -----
[Salida del proceso de datos]
----- DATA SAVED -----
```

```

Do you want to run the model optimization? (y/n): n
Enter the split type you want to use (CROSSVAL/HOLDOUT): CROSSVAL
Enter the model you want to use (KNN/DT/RF/ADABOOST/GBBOOST): RF
----- TRAINING MODEL -----
[Salida del entrenamiento del modelo]

```

4.3.2. Manual de usuario de la aplicación web

1. Abrir la Aplicación

- a) Abre tu navegador web.
- b) Ingresa la URL para la Aplicación de Predicción de Volcanes: <http://localhost:8080> o si quieres usar la aplicación desplegada: <https://volcano-map-frontend.onrender.com/>
- c) La página de inicio se cargará, mostrando el mapa y el encabezado.



Figura 4.6: Pantalla Principal

2. Registrar un Nuevo Usuario

- a) Haz clic en el botón *Register* ubicado en el encabezado.
- b) Completa los campos requeridos: nombre de usuario, correo electrónico, contraseña.



REGISTRATION PAGE

Username

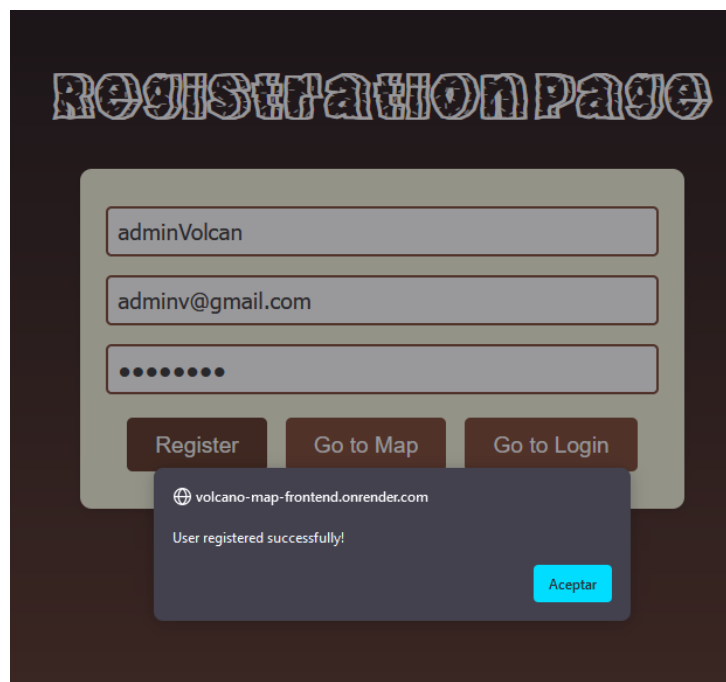
Email

Password

Register Go to Map Go to Login

Figura 4.7: Vista del formulario de registro

- c) Haz clic en el botón *Register* para enviar el formulario.



REGISTRATION PAGE

adminVolcan

adminv@gmail.com

.....

Register Go to Map Go to Login

volcano-map-frontend.onrender.com

User registered successfully!

Aceptar

Figura 4.8: Rellenar formulario de registro

- d) Tras un registro exitoso, serás redirigido a la página de inicio de sesión.

3. Iniciar Sesión

- a) Haz clic en el botón *Login* ubicado en el encabezado o si acabas de registrarte serás redirigido directamente a esta vista.

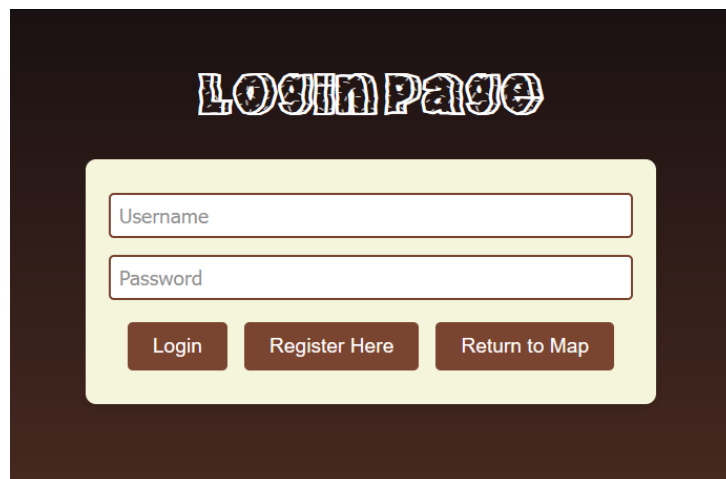


Figura 4.9: Vista del formulario de login

- b)* Ingresa tu nombre de usuario y contraseña.
- c)* Haz clic en el botón *Login* para iniciar sesión y esperar el mensaje de inicio exitoso.

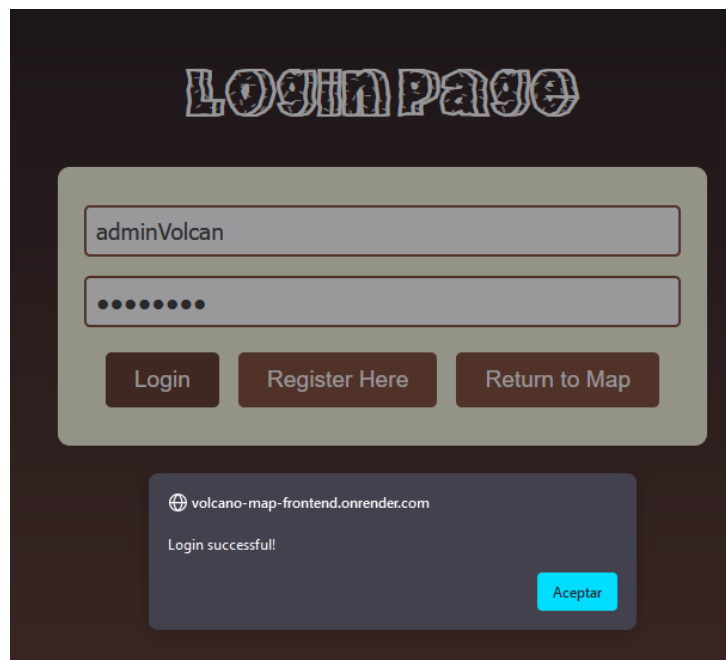


Figura 4.10: Rellenar formulario de login

- d)* Tras un inicio de sesión exitoso, serás redirigido a la página de principal.

4. Ver tu Perfil

- a)* Después de iniciar sesión, haz clic en el botón *Profile* ubicado en el encabezado.

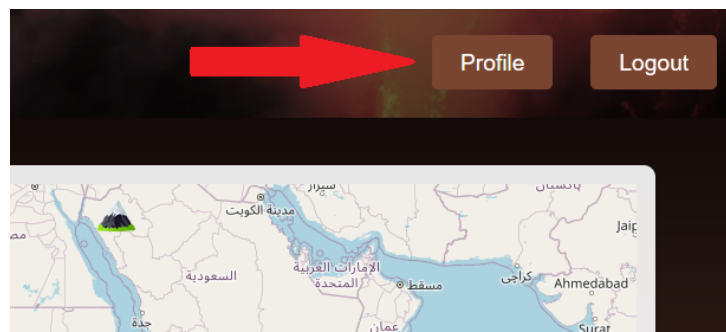


Figura 4.11: Encabezado tras login

b) Se mostrarán los detalles de tu perfil de usuario.

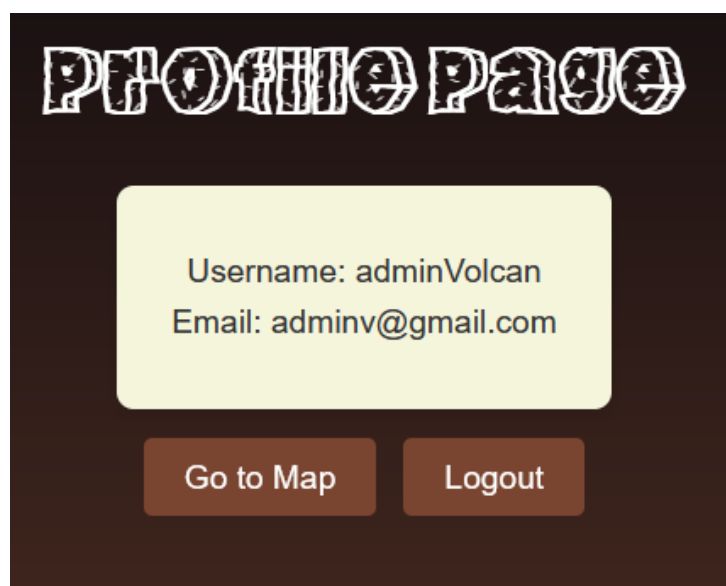


Figura 4.12: Vista de perfil de usuario

5. Ver Detalles del Volcán

a) Si arrastras el ratón hacia cualquier volcán aparecerá un popup con sus datos.

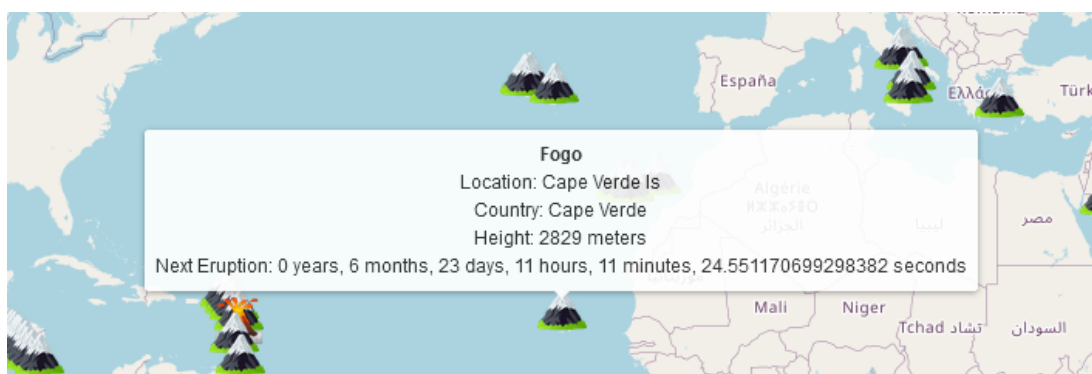


Figura 4.13: Popup de volcán

- b) Haz clic en cualquier marcador de volcán en el mapa, si quieres ver los datos detalladamente.
- c) Se mostrará una ventana emergente con detalles sobre el volcán, incluyendo nombre, ubicación, país, altura y próximo tiempo de erupción.

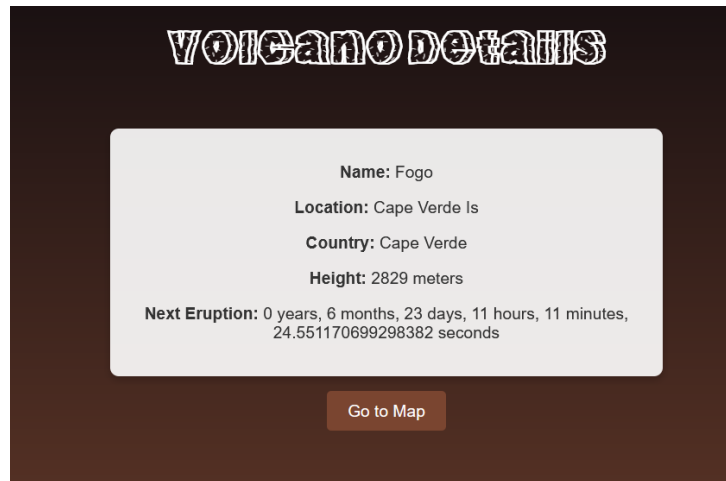


Figura 4.14: Vista de detalles del volcán

6. Cerrar Sesión

- a) Haz clic en el botón *Logout* ubicado en el encabezado.

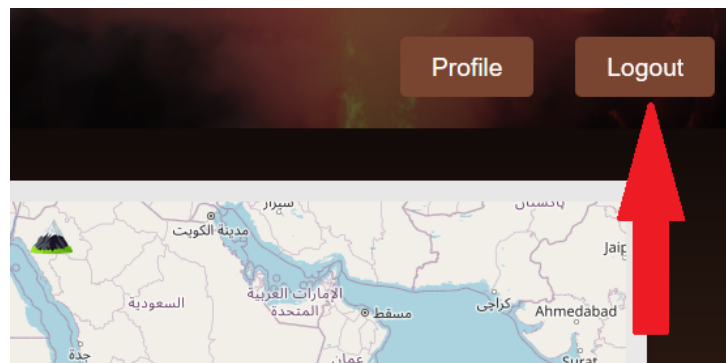


Figura 4.15: Botón de Logout

- b) Se cerrará tu sesión y serás redirigido a la página de inicio.

4.4. Despliegue

El despliegue de la aplicación se realizó utilizando la plataforma Render. Se utilizó el plan gratuito por lo que no es necesario añadirlo a los costes:

- El backend estará desplegado en esta url: <https://volcano-map.onrender.com>

- EL frontend estará desplegado en esta url: <https://volcano-map-frontend.onrender.com>

La API para los registros e inicio de sesión al ser el plan gratuito se queda congelado unos segundos por eso la aplicación no da respuestas hasta que deja de congelarse.

Parte IV

Cierre

CAPÍTULO 5

Cierre

En esta última parte se exponen los resultados, lecciones y conclusiones del trabajo en su totalidad.

5.1. Lecciones Aprendidas

En esta sección enumero las lecciones aprendidas durante la planificación y desarrollo de este TFG.

Desglosando lo general: He descubierto y comprobado en el desarrollo de este trabajo lo importante de tener una idea global o de alto nivel del proyecto completo y poco a poco, desgranar cada parte y ámbito especificando cada vez más las secciones consiguiendo asimismo, mejorar en conjunto no solo todo el documento sino también al software desarrollado.

La planificación es esencial: El hecho de que invirtiera mucho tiempo en la planificación, creando las contingencias necesarias y siguiendo la lección anterior “de lo general a lo específico”, me ha permitido poder responder a adversidades durante el desarrollo, como pudo ser la replanificación a mediados del cuatrimestre y organizar el desarrollo de manera cómoda pudiendo, además, realizar a un seguimiento exhaustivo, manteniendo las comunicaciones en todo momento con el tutor.

Mantener el foco: Este es el proyecto con más plazo a planificar y desarrollar que he realizado en mi carrera, lo cual hacía que en muchas ocasiones dejara a un lado las tareas

prioritarias o principales. En este trabajo he aprendido a ser más constante y focalizado en cada actividad y he aprendido lo esencial que es la secuenciación de las actividades.

Versatilidad de Python y sus librerías: Durante la carrera me estancué en el framework de Django. Al desarrollar este trabajo he descubierto no solo lo versátil que es Python, sino la gran cantidad de librerías y opciones para hacer la misma función, cada una con sus especificaciones. Tanto Pandas como Sklearn han sido dos grandes aliados en este trabajo y la experiencia adquirida con ambas librerías me ha ayudado a progresar como ingeniero del software.

5.2. Conclusiones

Las conclusiones que podemos obtener tras todo el estudio son las siguientes:

- El modelo GradientBoosting es sin duda el que mejor rendimiento tiene, pues si haberlo optimizado en un primer momento ya daba resultados bastantes interesantes, pero tras la optimización bayesiana era sin duda el mejor modelo. Otro modelo que me sorprendió durante la experimentación fue el árbol de decisiones, puesto que siempre obtenía buenas métricas mejorando incluso al AdaBoost siendo que el propio AdaBoost es un modelo bastante más potente. Puede que se deba a un mal ajuste del AdaBoost y por eso su rendimiento sea menor del esperado, teniendo en cuenta la gran diferencia entre este y el GradientBoost.
- La optimización bayesiana fue un gran salto de mejora para los modelos en los que, en general, todos los modelos mejoraron, exceptuando el KNN que de por sí no tiene mucha cabida a la optimización.
- La validación cruzada es un modelo de validación ideal y sobrepasa por bastante al modelo de validación por retención, por ello desde la implementación de la validación cruzada se han hecho algunas comparativas y las métricas obtenidas por validación cruzada siempre eran mejores, independientemente del modelo o dataframe escogido.
- El mayor salto y mayor mejora en toda la experimentación fue crear el dataset de la tercera iteración 3.4 puesto que contenía la mayor cantidad de información manteniendo la temporalidad y a la vista de los resultados está que usando el mejor modelo optimizado con ese dataframe tiene por bastante las mejores métricas obtenidas en la investigación.

5.3. Trabajo futuro

En cuanto a mejoras al software de experimentación:

- Mejorar en todos los ámbitos de la investigación para ascender de posición en la clasificación de Kaggle.
- Implementar algún modelo de Deep Learning o algún tipo de red neuronal.
- Utilizar una clase más exhaustiva como `EfficientFCParameters` para obtener un mayor y/o mejor número de características de los datos en crudo.

En cuanto a funcionalidades nuevas a la aplicación web, podemos añadir:

- El filtro de búsqueda de volcanes que se tuvo que recortar del alcance. El filtro podría hacerse tanto por nombre, como por altura, localización o tiempo de erupción.
- Lista de volcanes favoritos que se tuvo que recortar del alcance. Cada usuario debería tener una lista asociada a su correo o nombre de usuario y añadir mediante algún botón en los detalles del volcán ese volcán a su lista de favoritos y así tenerlos más accesibles.
- Una alarma personalizada para el usuario donde le llegue notificación al usuario cuando el volcán esté cerca de erupcionar.
- El reloj del tiempo de erupción sea a tiempo real, es decir, que se crear algún tipo de contador para cada volcán.
- Utilizar otra plataforma de despliegue o un plan de pago del propio Render para que la aplicación desplegada no sea tan lenta con las peticiones.

Parte V

Anexo

Reunión Inicial

- **Fecha:** 18/12/2023 - 11:00
 - **Duración:** 1 hora
 - **Medio:** Presencial
 - **Asistentes:** Aitor Rodríguez Dueñas, Manuel Jesús Jiménez Navarro y María del Mar Martínez Ballesteros
-

Reunión Post Estudio Previo

- **Fecha:** 22/01/2024 - 11:00
 - **Duración:** 45 minutos
 - **Medio:** Presencial
 - **Asistentes:** Aitor Rodríguez Dueñas, Manuel Jesús Jiménez Navarro y María del Mar Martínez Ballesteros
-

Reunión Revisión 1º Iteración

- **Fecha:** 21/02/2024 - 10:40
 - **Duración:** 1 hora
 - **Medio:** Presencial
 - **Asistentes:** Aitor Rodríguez Dueñas, Manuel Jesús Jiménez Navarro y María del Mar Martínez Ballesteros
-

Reunión 2º Iteración - Correcciones

- **Fecha:** 4/03/2024 - 12:30

- **Duración:** 30 minutos
 - **Medio:** Presencial
 - **Asistentes:** Aitor Rodríguez Dueñas, Manuel Jesús Jiménez Navarro y María del Mar Martínez Ballesteros
-

Reunión Revisión 2º Iteración

- **Fecha:** 18/03/2024 - 12:30
 - **Duración:** 1 hora
 - **Medio:** Presencial
 - **Asistentes:** Aitor Rodríguez Dueñas, Manuel Jesús Jiménez Navarro y María del Mar Martínez Ballesteros
-

Reunión 3º Iteración - Correcciones

- **Fecha:** 08/04/2024 - 12:30
 - **Duración:** 30 minutos
 - **Medio:** Presencial
 - **Asistentes:** Aitor Rodríguez Dueñas, Manuel Jesús Jiménez Navarro y María del Mar Martínez Ballesteros
-

Reunión Revisión 3º Iteración

- **Fecha:** 26/04/2024 - 10:30
- **Duración:** 1 hora
- **Medio:** Online

- **Asistentes:** Aitor Rodríguez Dueñas, Manuel Jesús Jiménez Navarro y María del Mar Martínez Ballesteros

Reunión Revisión 4º Iteración

- **Fecha:** 10/05/2024 - 10:00
- **Duración:** 1 hora
- **Medio:** Presencial
- **Asistentes:** Aitor Rodríguez Dueñas, Manuel Jesús Jiménez Navarro y María del Mar Martínez Ballesteros

Reunión Revisión 5º Iteración

- **Fecha:** 27/05/2024 - 12:30
- **Duración:** 1 hora
- **Medio:** Online
- **Asistentes:** Aitor Rodríguez Dueñas, Manuel Jesús Jiménez Navarro y María del Mar Martínez Ballesteros

Bibliografía

- [1] Istituto nazionale di geofisica e vulcanologia. Recuperado de: <https://www.ingv.it/>.
- [2] Instituto geográfico nacional - noticias e informe mensual de vigilancia volcánica de 2021. Recuperado de: https://www.ign.es/web/resources/volcanologia/html/CA_noticias_2021.html#20210920.
- [3] Predict volcanic eruptions — ingv - osservatorio etneo. Recuperado de: <https://www.kaggle.com/competitions/predict-volcanic-eruptions-ingv-oe>.
- [4] Repositorio de github del software de experimentación. Recuperado de: <https://github.com/AitorRD/TFG-Vulcan-Prediction>.
- [5] Project Management Institute. *La guía de los fundamentos para la dirección de proyectos (Guía del PMBOK)*. Project Management Institute, 6 edition, 2017.
- [6] Precio de la luz - informe ocu. Recuperado de: <https://www.ocu.org/vivienda-y-energia/gas-luz/informe/precio-luz>.
- [7] Página oficial de github. Recuperado de: <https://github.com/>.
- [8] Página oficial de python. Recuperado de: <https://www.python.org/>.
- [9] Página oficial sklearn. Recuperado de: <https://scikit-learn.org/stable/>.
- [10] Página oficial de seaborn. Recuperado de: <https://seaborn.pydata.org/>.
- [11] Página oficial de numpy. Recuperado de: <https://numpy.org/>.
- [12] Página oficial de pandas. Recuperado de: <https://pandas.pydata.org/>.
- [13] Página oficial de vue. Recuperado de: <https://vuejs.org/>.
- [14] Página oficial de django. Recuperado de: <https://www.djangoproject.com/>.
- [15] Página oficial de render. Recuperado de: <https://render.com/>.

- [16] Jesús García y José M. Molina, Antonio Berlanga, Miguel A. Patricio, Álvaro L. Bustamante, and Washington R. Padilla. *Ciencia de datos: Técnicas analíticas y aprendizaje estadístico*. PUBLICACIONES ALTARIA, S.L., 2018. Revisado por Sonia Vives y Carlos Martínez.
- [17] Curso kaggle - feature engineering. Recuperado de: <https://www.kaggle.com/learn/feature-engineering>.
- [18] Curso kaggle - python. Recuperado de: <https://www.kaggle.com/learn/python>.
- [19] Curso kaggle - pandas. Recuperado de: <https://www.kaggle.com/learn/pandas>.
- [20] Curso kaggle - intro to machine learning. Recuperado de: <https://www.kaggle.com/learn/intro-to-machine-learning>.
- [21] Curso kaggle - intermediate machine learning. Recuperado de: <https://www.kaggle.com/learn/intermediate-machine-learning>.
- [22] Curso kaggle - data visualization. Recuperado de: <https://www.kaggle.com/learn/data-visualization>.
- [23] Dragos D. Margineantu and Thomas G. Dietterich. Pruning adaptive boosting. Recuperado de: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b01224d0b698696d4d2d20d135adc29ba881db3b>.
- [24] Eric Brochu y Vlad M. Cora and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, December 2010. Accessed: 2024-06-22.
- [25] Página oficial de tsfresh. Recuperado de: <https://tsfresh.readthedocs.io/en/latest/>.
- [26] Página oficial de tsfresh - feature extraction. Recuperado de: https://tsfresh.readthedocs.io/en/latest/text/feature_extraction_settings.html.
- [27] Repositorio de bayes opt. Recuperado de: <https://github.com/bayesian-optimization/BayesianOptimization>.
- [28] Página oficial de sklearn - feature selection. Recuperado de: https://scikit-learn.org/stable/modules/feature_selection.html.
- [29] Pagina oficial de angular. Recuperado de: <https://docs.angular.lat/docs>.
- [30] Pagina de kaggle - volcano eruptions. Recuperado de: <https://www.kaggle.com/datasets/jessemostipak/volcano-eruptions?select=volcano.csv>.
- [31] Mdn web docs - introducción a django. Recuperado de: <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>.

Aitor Rodríguez Dueñas
Sevilla, 2024