



Universidad de Castilla-La Mancha  
Escuela Superior de Ingeniería Informática

**Trabajo Fin de Máster**  
Máster Universitario en Ingeniería Informática

**Adaptación de la metodología ágil  
a un proyecto de ciencia  
de datos**

*Rafael Muñoz González*

Junio, 2020





## TRABAJO FIN DE MÁSTER

Máster Universitario en Ingeniería Informática

# Adaptación de la metodología ágil a un proyecto de ciencia de datos

**Autor:** Rafael Muñoz González

**Tutor:** Pablo Bermejo López

**Co-Tutor:** Luis de la Ossa Jiménez

Junio, 2020



*A mis padres, mi hermano y mis amigos.  
Gracias a ellos soy la persona en la que me he convertido,  
y es a ellos hacia los que expreso mi más sincero  
agradecimiento.*



## **Declaración de autoría**

Yo, Rafael Muñoz González, con DNI 06287234T, declaro que soy el único autor del trabajo fin de máster titulado “Adaptación de la metodología ágil a un proyecto de ciencia de datos”, que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual, y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a 25 de Junio de 2020

Fdo.: Rafael Muñoz González



## Resumen

El presente Trabajo de Fin de Máster recoge la aplicación, impacto y adaptación de una metodología ágil en un proyecto de ciencia de datos, pues las metodologías convencionales utilizadas en el desarrollo software no son aplicables, necesitando una adaptación de las necesidades y tiempos.

Se ha tomado como base la metodología mostrada en el libro “*Agile Data Science 2.0*” [1], dividiendo el proyecto en 6 niveles bien diferenciados:

- Diseño de la arquitectura de la solución.
- Recolección y muestra de datos.
- Limpieza de datos y realización de agregaciones sobre los mismos para visualizar información que responde a preguntas interesantes por parte del usuario final.
- Diseño de la información, links y metadatos para que el usuario sea capaz de interactuar y explorar toda la información.
- Creación de modelos de predicción.
- Despliegue de los modelos y mejora continua.

## Summary

This Final Master's Paper covers the application, impact and adaptation of an agile methodology in a data science project, since the conventional methodologies used in software development are not applicable, requiring an adaptation of the needs and times.

The methodology shown in book *Agile Data Science 2.0* [1] has been taken as the basis of the project, dividing the project into 6 well-differentiated levels:

- Design of the architecture of the solution.
- Data collection and sampling.
- Data cleaning and aggregation to visualize information that answers interesting questions from the end user.
- Design of the information, links and metadata so that the user could interact and explore all the information.
- Creation of prediction models.
- Deployment of models and continuous improvement.

## Agradecimientos

Este Trabajo de Fin de Master realizado en la Universidad de Castilla La-Mancha es un esfuerzo en el que participaron diversas personas, ya sea directamente o indirectamente, opinando, corrigiendo, animando y acompañando en momentos de frustración y de felicidad.

Por todo esto, me gustaría agradecer a mis tutores todo el tiempo invertido en que este proyecto saliese adelante.

Además, también me gustaría agradecer la ayuda y conocimientos que me han aportado todos los profesores durante toda esta etapa de mi vida.

Por último agradezco a mi familia y amigos su apoyo incondicional en los momentos más difíciles.



# Índice general

---

<b>1 Introducción .....</b>	<b>1</b>
1.1 Motivación .....	2
1.2 Objetivos .....	3
1.3 Desarrollo del trabajo .....	4
1.4 Estructura de la memoria .....	5
<b>2 Aplicaciones de la ciencia de datos.....</b>	<b>7</b>
2.1 Inteligencia de negocio .....	7
2.2 Big Data .....	8
2.3 Modelos y técnicas de aprendizaje automático .....	9
2.4 Problemas de los proyectos de ciencia de datos.....	12
<b>3 Adaptación de la metodología ágil a aplicaciones basadas en ciencia de datos .....</b>	<b>13</b>
3.1 El desarrollo en cascada.....	13
3.2 El desarrollo ágil.....	14
3.3 Incrementos de valor en la ciencia de datos .....	18
3.4 Definición de los integrantes del equipo .....	19
3.5 Organización del trabajo .....	20
3.6 Herramientas de gestión .....	21
3.6.1 <i>Git-Hub</i> .....	21
3.6.2 <i>Zen-Hub</i> .....	21

<b>4 Sistema para la visualización.....</b>	<b>23</b>
4.1 Fase de preparación del trabajo .....	23
4.1.1 <i>Modelo de datos.</i> .....	23
4.1.2 <i>Modelo de visualización</i> .....	24
4.2 Integración de las tecnologías .....	28
4.3 Obtención de los datos .....	28
4.4 Limpieza, agregación y visualización de los datos .....	29
4.4.1 <i>Arquitectura del proyecto de ciencia de datos.</i> .....	34
4.5 Exploración de datos y creación del modelo de predicción .....	36
4.6 Despliegue del modelo de predicción .....	37
4.7 Mejora continua del modelo de predicción. ....	38
<b>5 Conclusiones y propuestas.....</b>	<b>39</b>
5.1 Conclusiones .....	39
5.2 Competencias cubiertas.....	39
5.3 Trabajo futuro .....	40
5.4 Opinión personal .....	41
<b>Referencia bibliográfica.....</b>	<b>41</b>

## Índice de figuras

---

2.1	Esquema de entrenamiento del modelo . . . . .	10
3.1	Sprints utilizando la metodología de desarrollo en cascada [1]. . . . .	13
3.2	Composición del equipo de ciencia de datos ágil . . . . .	17
3.3	Esquema de necesidades de un proyecto de ciencia de datos . . . . .	18
3.4	Proyecto de <i>Agile Data Science</i> en GitHub . . . . .	21
3.5	Tablero Kanban en ZenHub para el proyecto realizado . . . . .	22
3.6	Gráfico Burndown del proyecto realizado . . . . .	22
4.1	Página principal . . . . .	24
4.2	Página de cuadro de mando . . . . .	25
4.3	Página de información sobre los aviones . . . . .	25
4.4	Página de información sobre las aerolíneas . . . . .	26
4.5	Página de información sobre una aerolínea específica . . . . .	26
4.6	Página de información sobre un avión . . . . .	27
4.7	Página de predicciones . . . . .	27
4.8	Modelado de los datos . . . . .	29
4.9	Página principal de la aplicación . . . . .	30
4.10	Cuadro de mando para mostrar información al usuario . . . . .	30
4.11	Página de aerolíneas . . . . .	31
4.12	Página de información sobre una aerolínea . . . . .	31
4.13	Página sobre información de los aviones . . . . .	32
4.14	Página sobre información del avión . . . . .	32
4.15	Página de búsqueda de vuelos . . . . .	33
4.16	Página de predicciones de retrasos . . . . .	33
4.17	Diagrama funcional de la obtención de predicciones . . . . .	34
4.18	Diagrama de la arquitectura de la solución . . . . .	35
4.19	Ciclo de vida del modelo de predicción . . . . .	38



# 1. Introducción

---

La ciencia de datos es un campo interdisciplinario que involucra la estadística, la minería de datos, el aprendizaje automático, y la analítica predictiva, junto con otros procesos y sistemas para extraer conocimiento de los datos en sus diferentes formas: estructurados, semi-estructurados o no estructurados. El conocimiento que proporcionan los datos ayuda a tomar decisiones importantes que involucran, principalmente, los negocios y la calidad de vida.

La ciencia de datos ha sido objeto de un auge importante durante los últimos años. Este crecimiento se debe principalmente a dos fenómenos:

- **La evolución de la tecnología y de procesamiento:** la capacidad de almacenamiento aumenta año a año, permitiendo realizar tareas que antes no se podían realizar.
- **El aumento en la cantidad de datos:** proporcionados por sensores, negocios, redes sociales, etc.

Además, se prevé un incremento exponencial en el volumen de datos disponibles los próximos años, por lo que conocer la forma de tratarlos y generar valor con ellos es una habilidad que será cada vez más importante en el futuro.

Debido a esta nueva realidad, la mayoría de aplicaciones y servicios incluyen o se basan en aplicaciones centradas en los datos y, por tanto, la tecnología está proporcionando nuevos servicios que facilitan el uso de los datos a través de servicios escalables en la nube que permiten almacenar y procesar los datos reduciendo la inversión inicial para tener una infraestructura con la que trabajar.

---

## 1.1. Motivación

El ciclo de desarrollo de un servicio o aplicación basado en ciencia de datos tiene una serie de particularidades que modifican la forma de dirección.

En este tipo de proyectos la mayoría de veces el usuario final no tiene claro qué es lo que quiere al principio, y los requisitos pueden estar poco o mal definidos. Esto supone un gran problema, ya que se produce una gran cantidad de cambios en los requisitos de desarrollo. La implementación de dichos cambios supone una gran cantidad de tiempo, por lo que es importante elegir una metodología adecuada, que disminuya el riesgo de fracaso, y que se centre en la interacción con los usuarios con el fin de detectar problemas de desarrollo y definir los requisitos.

La implantación de una metodología ágil hace al usuario partícipe de la solución, por lo que cuando el proyecto esté finalizado el usuario sentirá que es parte de él y tendrá la sensación de que se han satisfecho sus necesidades, algo muy importante, pues es lo que indica que el proyecto ha tenido éxito.

En proyectos basados en ciencia de datos, se busca implementar una metodología ágil de igual forma que se hace en los proyectos de desarrollo software pero adaptándola a este contexto particular, ya que las metodologías ágiles que se utilizan para el desarrollo software no son aplicables cuando se trabaja en el ámbito de la ciencia de datos debido a: la no existencia de fechas de referencia para el lanzamiento de un artefacto; la gran cantidad de roles que componen el equipo; el tratamiento de la deuda técnica y cómo afecta esto a la agilidad del equipo, siendo necesario una adaptación a las necesidades y tiempos de las tareas relacionadas.

Por todo esto, el presente Trabajo de Fin de Máster trata sobre el desarrollo de un proyecto de ciencia de datos y se centra, sobretodo, en el problema del desarrollo de principio a fin utilizando una metodología ágil, integrando todas las tecnologías de la forma más eficiente y rápida, que permita entregar cuanto antes valor al usuario, iterando y mejorando el proyecto según sus necesidades.

## 1.2. Objetivos

El objetivo de este trabajo es crear un servicio que utiliza ciencia de datos, y demostrar que éste puede ser desarrollado siguiendo una metodología ágil. El servicio en sí trata de una página web que proporciona información sobre los vuelos, los retrasos que han tenido, y la predicción del retraso en los próximos vuelos.

La adaptación de la metodología a las fases del proyecto se ha realizado de la siguiente manera:

- **Diseño de la arquitectura de la solución.** Se establecen los requisitos que debe tener la aplicación, definiendo el modelo de los datos, el modo de visualización de la información y la arquitectura de la solución a las peticiones de predicciones. En este caso, la metodología ágil se aplica para refinar los requisitos de los datos con el usuario, definiendo cuales son los adecuados y proponiendo sugerencias y alternativas.
- **Integración de las tecnologías.** Se estudian las diferentes tecnologías que pueden ser aplicadas en el proyecto y se integran aquellas con mayor probabilidad de éxito. En este caso, la metodología ágil permite detectar problemas con algunas tecnologías a aplicar, como seguridad, política de negocio, etc.
- **Obtención de datos.** Mensualmente se obtienen los datos necesarios para almacenarlos siguiendo el modelo diseñado con anterioridad. En este caso, la metodología ágil se aplica para validar con el usuario si la información recogida es de interés o tiene el valor esperado.
- **Tratamiento y visualización de datos.** Se realiza un preprocesado de los datos para aumentar la calidad de los mismos y se desarrolla una forma de visualizarlos. En este caso, la metodología ágil se aplica para obtener la opinión del usuario sobre como visualizar la información, mostrándole los incrementos para demostrar que el proyecto va creciendo y aumentando el interés del usuario.
- **Desarrollo de modelo de predicción.** Se crea un modelo que pueda predecir los retrasos en los vuelos futuros. En este caso, la metodología ágil se aplica para que el usuario detecte problemas en el modelo de predicción, buscando qué es lo que realmente quiere predecir.
- **Despliegue del modelo de predicción.** Se orquestan todas las tecnologías integradas para que sigan la arquitectura de la solución a las peticiones de predicciones diseñada en la primera fase. En este caso, la metodología ágil se aplica para conocer la opinión del usuario con respecto a la solución de predicción, detectando si los tiempos de respuesta le parecen correctos al usuario y si la visualización de los resultados le parece adecuada.

---

## 1.3. Desarrollo del trabajo

El desarrollo del trabajo se ha basado en el estudio de la evolución de los proyectos de ciencia de datos, teniendo en cuenta las dificultades y problemas en llevarlos a cabo de una forma eficaz y centrada en el usuario final. Además, desarrollar nuevas funcionalidades con los datos implica el estudio de diseños de arquitecturas junto con el de las tecnologías implicadas.

El desarrollo del trabajo implica unir una gran cantidad de tecnologías para cada una de las fases de desarrollo:

1. **Fase de obtención, almacenamiento y tratamiento de datos.** Se obtienen y almacenan los datos necesarios para el desarrollo de la aplicación utilizando datos procedentes de ficheros y páginas web. Las tecnologías utilizadas son:
  - La obtención y tratamiento de los datos se ha realizado mediante Python [2] y la librería SparkSQL, de forma que para el tratamiento de los datos se ha seguido un patrón ELT (Extract, Load and Transform).
  - El almacenamiento de los datos se ha realizado con MongoDB[3], que es una base de datos no relacional.
2. **Fase de visualización.** Se muestran los datos almacenados en la aplicación web. Las tecnologías utilizadas son:
  - La parte de visualización de los datos se ha centrado en la creación de una página web mediante el framework de Flask[4] y Jinja[5], de forma que estará creada utilizando Python, JavaScript, Jquery y las estructuras que nos permite crear Jinja para el desarrollo más rápido de varias páginas dentro de la página web.
  - La búsqueda de datos a través de diversos parámetros se ha mejorado en rendimiento a través de la creación de índices en ElasticSearch [6], que aumentan la velocidad de acceso a los datos cuando se realizan consultas específicas.
3. **Fase de predicción.** Se añade a la aplicación web la funcionalidad de realizar predicciones sobre los retrasos de los vuelos futuros. Las tecnologías utilizadas son:
  - La predicción de eventos ha sido realizada mediante las librerías PySpark y SparkML. Estas librerías permiten parallelizar el entrenamiento de los modelos cuando existe una enorme cantidad de datos.
  - El entrenamiento regular del modelo de predicción se ha llevado a cabo utilizando Airflow que permite programar tareas en Python basándose en un grafo acíclico dirigido.
  - La obtención de predicciones “near-realtime” se ha llevado a cabo utilizando Kafka [7], ya que permite almacenar las peticiones de las predicciones y, posteriormente, enviarlas a SparkStreaming para ser resueltas.

- El procesamiento continuo de peticiones de predicción ha sido llevado a cabo mediante SparkStreaming[8], extensión del núcleo del Spark API, que permite el procesamiento escalable, de alto rendimiento y tolerante a fallos de flujos de datos en tiempo real. Los datos ingeridos por esta extensión pueden ser Kafka, Flume, Kinesis, o sockets TCP, y puede procesar dichos datos utilizando algoritmos complejos.

## 1.4. Estructura de la memoria

Esta memoria del Trabajo de Fin de Máster presenta los contenidos teóricos y, posteriormente, su puesta en práctica centrándose en cubrir las siguientes competencias establecidas:

- CE1 - Capacidad para la integración de tecnologías, aplicaciones, servicios y sistemas propios de la Ingeniería Informática, con carácter generalista, y en contextos más amplios y multidisciplinares.
- CE4 - Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.
- CE10 - Capacidad para comprender y poder aplicar conocimientos avanzados de computación de altas prestaciones y métodos numéricos o computacionales a problemas de ingeniería.
- CE12 - Capacidad para aplicar métodos matemáticos, estadísticos y de inteligencia artificial para modelar, diseñar y desarrollar aplicaciones, servicios, sistemas inteligentes y sistemas basados en el conocimiento.
- CE13 - Capacidad para utilizar y desarrollar metodologías, métodos, técnicas, programas de uso específico, normas y estándares de computación gráfica.
- CE14 - Capacidad para conceptualizar, diseñar, desarrollar y evaluar la interacción persona-ordenador de productos, sistemas, aplicaciones y servicios informáticos.

Estas competencias son tratadas en los diferentes capítulos de la memoria organizados de la siguiente forma:

- **Capítulo 2 - Aplicaciones basadas en ciencia de datos.** Ciencia de datos, aplicaciones, técnicas de aprendizaje automático, la problemática específica en el desarrollo de proyectos en este área y la necesidad de utilizar una metodología ágil.
- **Capítulo 3 - Adaptación de la metodología ágil a aplicaciones basadas en ciencia de datos.** Proceso de desarrollo en ciencia de datos y necesidad de establecer una metodología, estudiando las alternativas a elegir, centrándose en las dificultades y ventajas que supondría su aplicación en este proyecto. Por último se describe la metodología elegida.

- 
- **Capítulo 4 - Sistema para la visualización.** Pasos seguidos para la creación de un sistema de visualización de información. Se explican los pasos que se han seguido para el desarrollo de la arquitectura de solución; integración de las tecnologías implicadas; obtención y limpieza de los datos; creación de los paneles de control que sintetizan la información útil para el usuario; y la creación de un modelo de predicción para su posterior despliegue en un entorno de producción y mejora continua.
  - **Capítulo 5 - Conclusiones y propuestas.** Reflexión final sobre el presente Trabajo de Fin de Master. Se habla sobre la efectividad de la propuesta realizada y también de los trabajos futuros con respecto a este proyecto.

## 2. Aplicaciones de la ciencia de datos

---

Existe la idea de que la ciencia de datos [9] es una disciplina de reciente creación, pero en la realidad el trabajo en este área se remonta a la década de los sesenta como evolución de la estadística, la minería de datos, el aprendizaje automático y la analítica predictiva, involucrando métodos científicos, procesos y sistemas para extraer conocimiento o mejorar el entendimiento de los datos en sus diferentes formas, ya sean estructurados o no estructurados. Esta disciplina ofrece un conjunto cada vez más maduro de técnicas y principios orientados a la extracción de información útil implícita en un conjunto de datos.

Desde la aparición de la ciencia de datos han surgido nuevas aplicaciones que se sustentan en este área, desde los análisis descriptivos a los predictivos, siendo difícil encontrar un proyecto desarrollo software que no tenga alguna parte relacionada con este área, por lo que en este capítulo se mostrará todo lo relacionado con la misma y sus aplicaciones.

### 2.1. Inteligencia de negocio

Inteligencia de negocio [10] (Business Intelligence) es un término general que incluye los procesos y métodos para recopilar, almacenar y analizar datos de actividades u operaciones de negocios, ayudando a las empresas en las actividades relacionadas con la identificación de áreas en las que aumentar ganancias; análisis de comportamiento de clientes; comparaciones con la competencia; optimización de operaciones; predicción de éxito en emprendimientos nuevos; identificación de tendencias de mercado; e identificación de problemas en el negocio.

La Inteligencia de negocio está muy ligada a la ciencia de datos, pero se caracteriza por:

- **Ámbito de aplicación.** La inteligencia de negocio se centra mayormente en datos internos de la empresa, mientras que la ciencia de datos trata datos tanto internos como externos.
- **Tipo de datos utilizados.** La inteligencia de negocio trabaja con datos de tipo comercial como son las ventas, marketing, servicio hacia el cliente y además datos de los

---

empleados y la empresa. Por el contrario, la ciencia de datos se basa tanto documentos como estadísticas y elementos del social media como emails, audios, vídeos, fotografías etc.

- **Técnicas utilizadas.** La inteligencia de negocio se basa en técnicas empresariales e IT mientras que la ciencia de datos se basa en matemáticas, estadísticas y estrategias empresariales.

Las áreas de trabajo en las que se centra son:

- **Minería de datos:** proceso de extraer información de distintas fuentes.
- **Preparación de datos:** proceso encargado de realizar la transformación de los datos para mejorar la calidad de los pasos posteriores.
- **Pruebas comparativas y métricas de rendimiento:** proceso en el que se generan métricas para medir el rendimiento y comparar aspectos del negocio.
- **Análisis descriptivos:** proceso que busca dar respuesta a alguna pregunta formulada.
- **Análisis estadístico:** proceso para describir una situación recopilando y explorando grandes cantidades de datos para descubrir patrones y tendencias implícitas en los mismos.
- **Visualización de datos:** proceso encargado de mostrar la información de forma visual, normalmente en forma de gráficos que resuman los datos y proporcionen valor.
- **Generación de informes:** proceso en el que se genera un documento con la información importante relativa al negocio.

Esta rama del análisis de datos está teniendo una gran relevancia en estos últimos años debido al crecimiento de los datos generados por los procesos de los negocios de organizaciones de todos los tamaños. Todos esperan poder acceder a información nueva y usarla para fundamentar decisiones diarias y satisfacer su curiosidad sobre el estado en el que se encuentran y cuáles serán los próximos pasos.

## 2.2. Big Data

Se habla de Big Data[11] cuando se trabaja con volúmenes de datos de un gran tamaño y complejidad, especialmente porque proceden de diversas fuentes.

El conjunto de datos sobre el que se trabaja se considerará Big Data si cumplen las propiedades denominadas como las 5 'Vs':

- **Volumen:** Se deben procesar grandes volúmenes de datos, que pueden ser de valor desconocido, como feeds de datos de Twitter, datos que contienen los flujos de clicks de una página web, datos proporcionados por una aplicación de móvil o los datos generados por un equipo con sensores. Esto puede suponer decenas de terabytes o, incluso, cientos de petabytes.

- **Velocidad:** Se refiere tanto al ritmo con el que se reciben los datos como a la velocidad con la que se aplica alguna acción. Algunos productos necesitan funcionar y actuar a tiempo real.
- **Variedad:** Se refiere a los diversos tipos de datos disponibles. Los tipos de datos convencionales eran estructurados y podían organizarse claramente en una base de datos relacional. Con el auge del Big Data, los datos se presentan en nuevos tipos de datos no estructurados o semi-estructurados como el vídeo, el audio y el texto. Debido a este nuevo paradigma se requiere un pre-procesamiento adicional para poder obtener significado y generar los meta-datos.
- **Valor:** Se refiere a la importancia que tienen los datos, es decir, la oportunidad de sacarles el máximo partido.
- **Veracidad:** Aparece debido al gran volumen de datos generados, pues puede hacer que dudemos del grado de veracidad de todos ellos, ya que esto provoca que muchos de ellos lleguen incompletos o incorrectos.

Esto supone un cambio en la concepción de los sistemas, que deben ser replanificados y escalados, ya que el software de procesamiento de datos convencional sencillamente no puede gestionarlos. Sin embargo, estos volúmenes masivos de datos pueden utilizarse para abordar problemas empresariales que antes no hubiera sido posible solucionar.

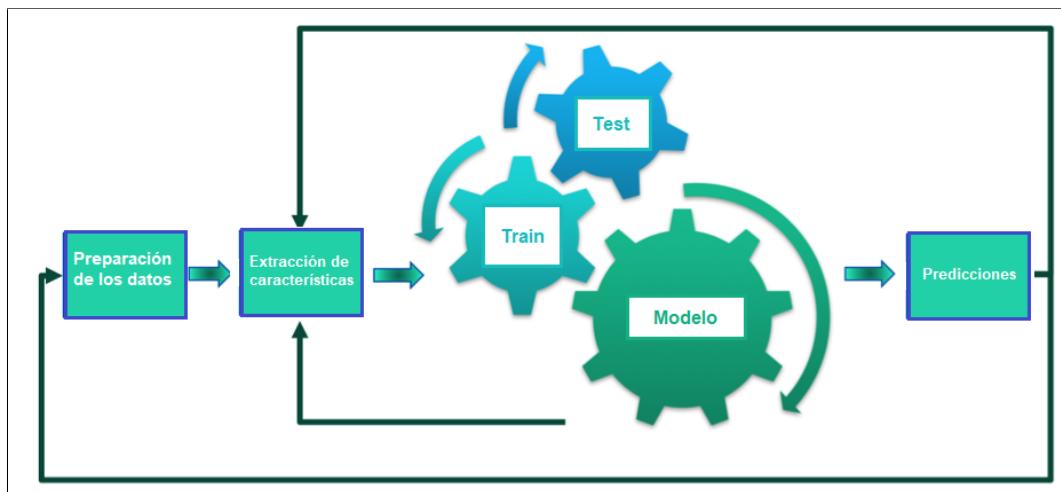
Este proyecto está basado en el Big Data puesto que requiere utilizar datos con alta volumetría y predicciones a una gran velocidad, concretamente “near-realtime”, utilizando varias fuentes de datos entre las que se destacan los datos obtenidos a partir del tratamiento de ficheros y textos en bruto de páginas web.

Además, se centrará en garantizar que los datos utilizados tengan valor y sean veraces, pues el fin es generar valor al cliente.

### 2.3. Modelos y técnicas de aprendizaje automático

El análisis predictivo permite aportar valor añadido a las aplicaciones basadas en ciencia de datos, ya que permite describir qué puede pasar en el futuro. Para realizar este análisis predictivo se necesita crear un buen modelo de predicción que pueda proporcionar una buena predicción.

Un modelo de predicción es un algoritmo que estima el valor de una o varias variables de salida en base a unos datos de entrada. Este modelo de predicción debe ser entrenado con datos históricos para que el modelo sea capaz de capturar la relación entre los datos de entrada y las variables objetivo de la predicción. Para que un modelo de predicción sea útil, es necesario que los datos de entrenamiento sean de buena calidad y sean tratados, limpia-dos y mejorados. Además, algo muy a tener en cuenta en las aplicaciones es la necesidad de recoger más información habitualmente para reentrelar el modelo, y que de esta forma se adapte a los cambios en los datos y mejore los resultados (Figura 2.1).



**Figura 2.1:** Esquema de entrenamiento del modelo

Existen muchas técnicas de aprendizaje automático. Un factor determinante para su elección es la envergadura de los datos, puesto que una buena solución dependerá de la técnica que se utilice:

- **Poca cantidad de datos:** técnicas de aprendizaje automático que no utilizan paralelismo de datos distribuidos en varias máquinas/nodos.
- **Gran cantidad de datos:** técnicas de aprendizaje automático que utilicen paralelismo de datos distribuyéndolos en varias máquinas/nodos.

Las técnicas de aprendizaje automático que no utilizan paralelismo distribuido en varias máquinas/nodos se basan en utilizar un ordenador local, utilizando un lenguaje de programación como Python y librerías como Scikit-Learn para la creación y entrenamiento de modelos de predicción. El problema con este tipo de técnicas es la limitación de los recursos, ya que el modelo a entrenar y los datos son almacenados en memoria y esto limita la capacidad de entrenamiento del modelo.

Por otro lado, las técnicas de aprendizaje automático que utilizan paralelismo distribuido en varias máquinas/nodos aparecen cuando se necesitan más recursos, ya sea memoria o procesamiento. Entre estas técnicas podemos encontrar:

- **Spark[12]:** Framework de computación en cluster que aparece en 2010 como una evolución a Hadoop Map Reduce centrándose en la velocidad. Su potencia se basa en el uso de RDD o Resilient Distributed Dataset que es un conjunto de datos de solo lectura que se distribuyen a lo largo de un clúster de máquinas que se mantiene en un entorno tolerante a fallos. La velocidad sobre las operaciones de los RDD depende de si estas afectan a una o varias particiones. Las mejoras que se implementaron con respecto a Hadoop fueron las siguientes:

1. Spark tiene una mayor velocidad en el procesamiento de los datos debido a que los RDDs trabajan en memoria, mientras que en Hadoop los datos son tratados en disco.
  2. Spark tiene APIs más sencillas de utilizar.
  3. Spark tiene librerías que aumentan su usabilidad como Mlib para aprendizaje automático, GraphX para grafos, Spark Streaming para tiempo real y Spark SQL para la consulta utilizando el lenguaje SQL.
- **Dask[13]:** Framework de computación en cluster que aparece en el 2014 y está orientado a escalar rápidamente un proyecto local a un proyecto basado en clusters de forma que la mayoría del código no necesite ser modificado. Su potencia se basa en el uso de Dask Dataframes que es un gran DataFrame paralelo compuesto por muchos DataFrames de Pandas más pequeños que se distribuyen a lo largo de un clúster. Además, debido a que se basa en DataFrames de Pandas, permite migrar proyectos desarrollados con pandas y otras librerías a un proyecto basado en clusters.

La diferencia entre Spark y Dask radica en los siguientes aspectos:

**1. Paralelismo:**

- El paralelismo en Spark se basa en las primitivas de más alto nivel (map, reduce, groupby, join).
- El paralelismo en Dask es más personalizable que en Spark debido a la incorporación de APIs de bajo nivel que permiten construir paralelismo personalizado.

**2. Aprendizaje automático:**

- Spark se basa en la biblioteca de MLlib que permite realizar operaciones de aprendizaje automático basándose en el esquema map-reduce.
- Dask se basa en las bibliotecas existentes como Scikit-Learn y XGBoost, e interopera con ellas. Al ser más familiares puede ser un buen punto a favor para elegir esta opción.

**3. Datasets:**

- El Dataframe de Spark tiene su propia API y modelo de memoria. También implementa un gran subconjunto del lenguaje SQL. Spark incluye un optimizador de consultas de alto nivel para consultas complejas.
- El Dataframe de Dask reutiliza la API y el modelo de memoria de Pandas. No implementa ni SQL ni un optimizador de consultas. Es capaz de realizar accesos aleatorios, operaciones eficientes de series temporales y otras operaciones indexadas al estilo de Pandas.

---

#### **4. Arrays:**

- Spark no incluye soporte de arrays multidimensionales.
- Dask incluye soporte del modelo de arrays de Numpy.

#### **5. Streaming:**

- SparkStreaming esta muy bien integrado con otras APIs pero el procesamiento a tiempo real sigue un enfoque basado en mini lotes.
- Dask permite el procesamiento a tiempo real pero requiere más trabajo.

La opción que se ha elegido en este trabajo es la de utilizar Spark, ya que es la más escalable a la nube mediante Databricks[14] y porque se necesita la distribución de los datos para poder entrenar el modelo debido a las limitaciones del equipo.

## **2.4. Problemas de los proyectos de ciencia de datos**

Tras conocer lo que es la ciencia de datos, sus aplicaciones, los modelos de predicción y los tipos de técnicas para el aprendizaje, se debe estudiar la problemática que presenta el desarrollo de aplicaciones basadas en ciencia de datos con respecto a la metodología a utilizar.

El problema que se pretende resolver en este Trabajo de Fin de Máster es el de desarrollar un proyecto de ciencia de datos de forma exitosa centrándose en la metodología de desarrollo software más extendida actualmente. El inconveniente de las metodologías ágiles convencionales que se utilizan para el desarrollo software es que no son aplicables cuando se trabaja en el ámbito de la ciencia de datos, de forma que es necesaria una adaptación de las necesidades y tiempos de las tareas con respecto a los siguientes aspectos que serán tratados en profundidad en el próximo capítulo:

- No existen fechas de referencia para el lanzamiento de un artefacto ya que un artefacto en ciencia de datos son valores, gráficos o modelos predictivos cuya disponibilidad con calidad no puede ser planificada igual que una característica funcional del software.
- Aparecen gran cantidad de roles que componen el equipo.
- El tratamiento de la deuda técnica y como afecta esto a la agilidad del equipo.

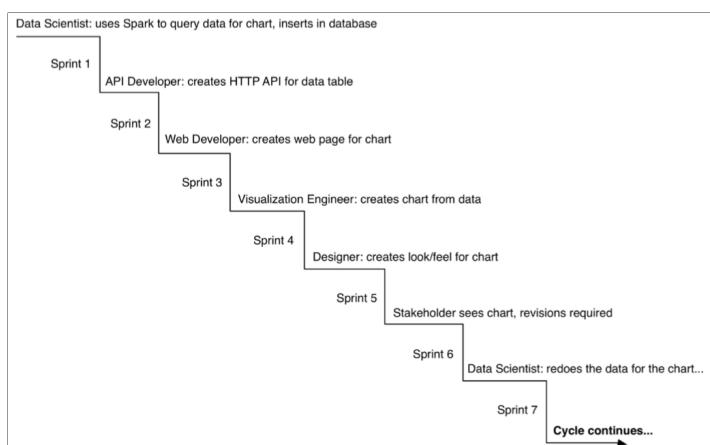
### 3. Adaptación de la metodología ágil a aplicaciones basadas en ciencia de datos

---

El desarrollo de proyectos gestionados mediante metodologías ha sufrido grandes cambios desde su propuesta en 1980. Desde ese momento han surgido progresivamente nuevas metodologías que han lidiado con problemas que tenían las anteriores. En el caso particular de este trabajo se verá la evolución de las metodologías hasta desarrollar una que lidie con los problemas propios de los proyectos de ciencia de datos.

#### 3.1. El desarrollo en cascada

La metodología de desarrollo en cascada se adoptó ampliamente en los años ochenta y noventa, y es la base de todos los demás modelos de ciclo de vida. Esta metodología ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.



**Figura 3.1:** Sprints utilizando la metodología de desarrollo en cascada [1].

---

El desarrollo de una aplicación basada en Big Data es mucho más complejo que el de una aplicación normal, por lo que se necesita un amplio conjunto de habilidades para crear aplicaciones que sean capaces de escalar. El método de desarrollo en cascada en estos proyectos puede hacerse tedioso ya que se necesitan muchos sprints para realizar una iteración (Figura 3.1). De esta forma, el desarrollo en cascada demuestra varios inconvenientes:

- En la vida real, un proyecto rara vez sigue una secuencia lineal, esto crea una mala implementación del modelo, lo cual hace que lo lleve al fracaso.
- El proceso de creación del software tarda mucho tiempo y hasta que el software no está completo es entregado.
- Una etapa determinada del proyecto no se puede llevar a cabo a menos de que se haya terminado la etapa anterior.
- Un científico de datos puede optimizar una solución sin cesar para dar las mejores predicciones posibles o para obtener todas las funcionalidades de un cuadro de mando previstas antes de compartir los resultados. De esta forma los interesados podrían perder el interés y la confianza en la finalización exitosa del proyecto.
- Podría existir ambigüedad en cuanto a qué predecir o qué fuentes de datos son de las correctas debido a la falta de comunicación entre las partes interesadas, los empresarios y los científicos de datos.

Por todos estos inconvenientes, nace la metodología ágil, haciendo el proceso de desarrollo más productivo y eficiente.

## 3.2. El desarrollo ágil

El desarrollo de un proyecto de software utilizando una metodología ágil permite adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

La metodología ágil resuelve algunos de los inconvenientes que presentaba la metodología en cascada. Sin embargo, las metodologías ágiles que se utilizan en el desarrollo software, como “Scrum”, presentan dificultades cuando se aplican a un desarrollo de un proyecto de ciencia de datos, pues tienen diferencias en el tratamiento de la deuda técnica y en los principios aplicados.

### **Deuda Técnica**

El uso de una metodología ágil para el desarrollo software trae consigo la contracción de deuda técnica para incrementar el valor de la solución de forma rápida. Esta deuda técnica es un concepto en el desarrollo de software que refleja el costo implícito del trabajo adicional por haber elegido una solución fácil, en lugar de utilizar un enfoque que llevaría más tiempo en su desarrollo e implementación. Contraer deuda técnica no es negativo, puesto que permite avanzar en algunos proyectos. El problema surge cuando se acumula deuda técnica y esta no es pagada, pues según vaya aumentando el desconocimiento o entropía del software, más trabajo adicional se tendrá que realizar. Las dificultades de utilizar un desarrollo ágil en un proyecto de ciencia de datos surgen debido a que la deuda técnica se trata de forma diferente a como sería en un proyecto de desarrollo software.

En el desarrollo software, la deuda técnica debe ser tratada en todas las funcionalidades desarrolladas, en cambio, en el desarrollo de un proyecto de ciencia de datos, cuando se desarrollan varios modelos de predicción, solamente se trata la deuda técnica en aquellos que dan mejores resultados, ya que se necesita entender completamente la solución a aplicar para mejorarla y comprobar que tiene suficiente calidad para mantenerla y reutilizarla.

Por otro lado, en el desarrollo software, para lidiar con la deuda técnica se suele utilizar el conocimiento de otros investigadores, mientras que en un proyecto de ciencia de datos, el conocimiento de otros investigadores suele ser usado para crear prototipos que posteriormente serán descartados o mínimamente usados.

### **Principios de un equipo de ciencia de datos**

La agilidad proporcionada por una metodología ágil se basa en los principios del equipo, como la definición de fechas de referencia para lanzar un artefacto y la segmentación óptima de roles para mejorar la cadena de trabajo de un proyecto. El inconveniente de utilizar una metodología ágil estándar en un proyecto de ciencia de datos aparece debido a que los principios de un equipo de ciencia de datos y los de un equipo de desarrollo software no se basan en los mismos aspectos.

En un proyecto de desarrollo software siempre existen fechas que se utilizan como referencia para que un artefacto sea lanzado, mientras que en un proyecto de ciencia de datos no hay fechas predeterminadas debido a que no se conoce cuando se va a obtener un requisito o artefacto. A cambio, lo que se obtiene es una verdadera visibilidad del trabajo del equipo hacia los objetivos del negocio, haciendo ver lo que el equipo está haciendo en esos instantes.

Con esta visibilidad del trabajo que realiza el equipo de ciencia de datos, otros procesos de negocio pueden estar alineados con el proyecto. De esta forma, los usuarios, pueden saber hacia donde se está moviendo el proyecto y coordinar los esfuerzos.

---

A su vez, los roles de un equipo de ciencia de datos no son exactamente los mismos que los de un equipo de desarrollo software. Concretamente los roles de un equipo de desarrollo software que trabaje con ciencia de datos son los siguientes:

- **Customers:** Usan el producto. Se debe crear valor para ellos repetidamente ya que su interés determina el éxito del producto.
- **Business Development:** Perfil estratégico que se encarga de aumentar los ingresos del negocio explorando nuevas opciones.
- **Marketers:** Hablan con el cliente y determinan que mercados seguir. Determinan la perspectiva que se debe seguir en un proyecto de ciencia de datos.
- **Product managers:** Toman la perspectiva de cada rol, las combina y construye un consenso sobre la visión y la dirección que debe seguir el producto.
- **User experience designers:** Responsables de encajar el diseño de la visualización del dato para que siga la perspectiva del cliente.
- **Interaction designers:** Diseñan la interacción entre los modelos de datos para dar mayor valor al usuario.
- **Web developers:** Crean la aplicación web para mostrar los datos en el navegador.
- **Engineers:** Construyen los sistemas para entregar los datos a las aplicaciones.
- **Data Scientist:** Exploran y transforman los datos de muchas formas para crear y publicar nuevas características, combinándolos de diversas fuentes para crear valor. Se encargan, además, de crear su visualización con los investigadores, ingenieros, desarrolladores webs y diseñadores.
- **Applied researchers:** Resuelven los problemas que los científicos de datos no pueden tratar y que se interponen en el camino de la entrega del valor. Requieren un enfoque y tiempo para resolverlos, además de métodos novedosos de estadística y aprendizaje automático.
- **Platform or data engineers:** Resuelven problemas en la infraestructura distribuida que permite a la metodología ágil de ciencia de datos realizarse sin inconvenientes. Además, se encargan de implementar planes y proyectos para mantener y mejorar la usabilidad de los datos para los investigadores, científicos de datos e ingenieros.
- **Quality assurance engineers:** Se encargan de automatizar las pruebas de los sistemas de predicción de principio a fin para asegurar que se hagan predicciones precisas y confiables.
- **Operations/DevOps engineers:** Se encargan de asegurar la instalación y el funcionamiento sin problemas de la infraestructura de datos de producción. Automatizan el despliegue y aparecen cuando las cosas tienen un funcionamiento inadecuado.

Como se ha podido ver, aparecen nuevos roles cuando un equipo de desarrollo trabaja con ciencia de datos con respecto a otro equipo que solamente desarrolla software. Estos nuevos roles son “*Platform or data engineers*”, “*Data Scientist*” y “*Applied researchers*” que

---

### *3. Adaptación de la metodología ágil a aplicaciones basadas en ciencia de datos*

---

se encargarán de todo lo relacionado con la explotación de los datos y la infraestructura que está por debajo.

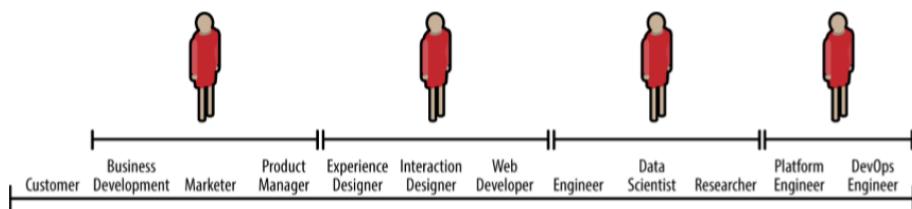
Todos los roles de un equipo que trabajan con ciencia de datos muestran el amplio conjunto de aptitudes necesarias para elaborar un producto final. Este gran conjunto de aptitudes representa tanto una oportunidad como un problema.

En un equipo de desarrollo software, por lo general, cuando se necesita tal conjunto de aptitudes, lo que se busca es que estas habilidades puedan ser llevadas a cabo por expertos en cada función trabajando en equipo, de forma que los problemas puedan descomponerse en partes y ser atacados directamente. Al utilizar este enfoque en un equipo de ciencia de datos, crece el tamaño del equipo para mejorar las habilidades en estas áreas. Esto conlleva un problema con la comunicación, pues las reuniones con 12 personas serían poco productivas. Una posible solución sería dividir este equipo en múltiples departamentos, estableciendo entregas entre ellos, pero esto provocaría una pérdida tanto en agilidad como en cohesión.

Al problema de la cantidad de habilidades necesarias, y la comunicación, se le sumaría tener más de una visión del desarrollo del producto. Por todo ello, para permanecer ágiles se recomienda:

- Elegir personas con un conocimiento más general en lugar de especialistas.
- Crear equipos de tamaño reducido.
- Usar herramientas y plataformas de alto nivel: computación en nube, sistemas distribuidos y plataformas como servicio (PaaS).
- Compartir de forma continua e iterativa el trabajo intermedio, incluso cuando ese trabajo esté incompleto.

Con estos cambios, el proyecto de ciencia de datos ágil se compondría de un pequeño equipo de personas con un conocimiento más general que utiliza herramientas escalables de alto nivel para refinar los datos de manera iterativa, añadiendo cada vez más valor en sus entregas tal y como se puede observar en la Figura 3.2.

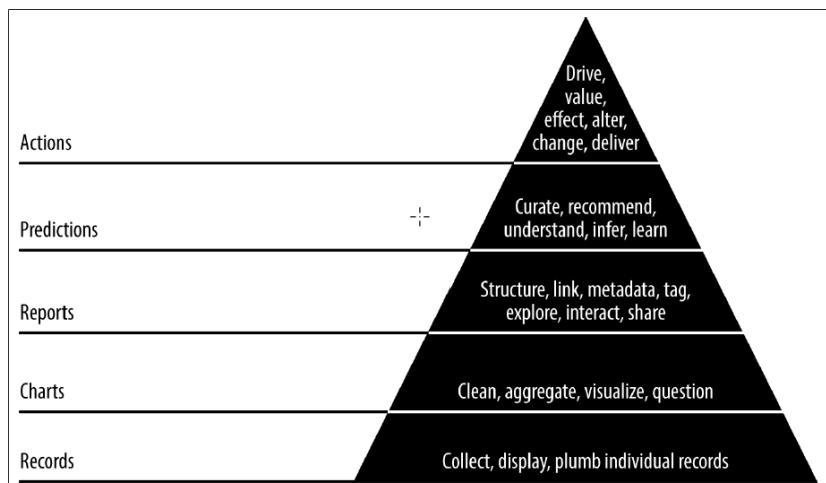


**Figura 3.2:** Composición del equipo de ciencia de datos ágil

### 3.3. Incrementos de valor en la ciencia de datos

La adaptación de una metodología ágil a la ciencia de datos busca entregar valor al usuario siguiendo la escalada de una pirámide de necesidades basada en la de Maslow[15] pero enfocada al desarrollo de un proyecto de ciencia de datos (Figura 3.3). Esta pirámide expresaría el incremento de valor creado al refinar los datos en bruto obteniendo tablas, gráficos, informes y predicciones para añadir nuevas acciones o mejorar las existentes siguiendo estos niveles:

- El primer nivel de la pirámide de valor de datos (registros) se refiere a crear un flujo de datos encargado de recoger los datos desde el origen hasta almacenarlos para mostrarlos en la aplicación.
- El segundo nivel añade la capa de gráficos y tablas, comenzando de esta forma el refinamiento y el análisis.
- En el tercer nivel, se añade la capa de informes que permite una exploración inmersiva de los datos, permitiendo razonar sobre ellos y llegar a conocerlos.
- El cuarto nivel añade la capa de predicciones que puede añadir mucho valor. En este nivel, la creación de buenos modelos de predicción necesitan aplicar ingeniería de características, aumentando la eficacia predictiva de los algoritmos de aprendizaje mediante la creación de características de los datos sin procesar que facilitan el proceso de aprendizaje. Esta capa se apoya fuertemente en los niveles inferiores.
- El nivel final es donde la inteligencia artificial toma lugar, la capa de predicción es desplegada para que el usuario la utilice. Esta capa de predicción debe ir mejorando para adaptarse a los cambios de los datos y aportar el máximo valor. Si este nivel de la pirámide se descuida, el valor proporcionado por la capa de predicción podría perderse.



**Figura 3.3:** Esquema de necesidades de un proyecto de ciencia de datos

### *3. Adaptación de la metodología ágil a aplicaciones basadas en ciencia de datos*

---

La pirámide de valor de datos proporciona pautas a seguir en el trabajo pero se debe tener en cuenta que esta pirámide no es una regla fija a seguir. Es posible subir a un nivel más alto o retroceder a uno más bajo. Además, la pirámide proporciona una visibilidad de si se ha incurrido en una deuda técnica al añadir un conjunto de datos directamente a un modelo predictivo y no se ha hecho transparente y accesible añadiéndolo al modelo de datos de la aplicación en los niveles inferiores.

Otro de los principios de un proyecto de ciencia de datos es el de aprovechar la computación en nube, los sistemas distribuidos y la plataforma como servicio (PaaS) para llevar un desarrollo ágil y eliminar los inconvenientes al lidiar con la infraestructura. Posteriormente, se usarían estas tecnologías para publicar de forma iterativa los resultados de la investigación.

## **3.4. Definición de los integrantes del equipo**

Tras haber definido cómo debe ser un equipo y aprender cómo dirigir un proyecto de ciencia de datos para que su resultado sea exitoso, se ha decidido seguir una metodología basada en Scrum pero con las siguientes adaptaciones que permiten adaptar el escenario real al contexto del desarrollo de un trabajo de fin de máster.

Los roles de los integrantes del equipo han sido:

- **“Product Owners”**: Los directores del Trabajo de Fin de Máster. Buscan maximizar el valor que se proporciona al usuario (tribunal del trabajo de fin de máster).
- **“Scrum Masters”**: Los directores del Trabajo de Fin de Máster. Facilitan los eventos y quitan impedimentos que pueda tener el equipo de trabajo.
- **“Development Team”** El estudiante. Encargado de desarrollar todas las fases del proyecto.

La metodología utilizada para el desarrollo del proyecto, al estar basado en una metodología ágil como el framework ‘Scrum’, se ha centrado en desarrollar características de forma rápida y ágil, entregando valor a través de sprints (bloques de tiempo de hasta un mes de duración). Previo a esto, se han establecido los requisitos del proyecto y se han almacenado en un “Product Backlog”. Además, para dar transparencia del trabajo realizado se ha utilizado un panel Kanban con la herramienta ZenHub.

Para el desarrollo ágil, el sprint se ha llevado a cabo en un evento en el que participan todos los miembros del equipo Scrum , constando de los siguientes pasos:

- **Revisión del sprint.** Se revisa el incremento realizado en el sprint anterior de la siguiente manera:
  1. Los “Product Owners” indican qué elementos del “Product Backlog” han sido terminados y cuales no.

- 
2. El Equipo de Desarrollo comenta los problemas que aparecieron y cómo se resolvieron.
  3. El “Development Team” hace una demostración del trabajo finalizado y responde preguntas sobre el incremento realizado al proyecto.
  4. Los “Product Owners” proyectan próximos objetivos.
  5. El grupo completo colabora acerca de qué hacer a continuación, corrigiendo en caso de ser necesario el “Product Backlog”.
- **Retrospectiva del sprint.** Se inspecciona cómo fue el sprint en cuanto a procesos y herramientas, buscando mejorar el proceso de desarrollo del proyecto.
  - **Planificación del sprint.** Se establece un objetivo para el próximo sprint y se planifica el trabajo a realizar durante él, almacenándose en un “Sprint Backlog”.

### 3.5. Organización del trabajo

En la sesión inicial donde se han establecido los requisitos del proyecto, la organización planteada ha sido la siguiente:

- **Inicio del proyecto:** Se planteará el proyecto con su planificación, definiendo de forma preliminar requisitos del sistema y todos los aspectos técnicos del proyecto.
- **Estudio del estado del arte:** Se centrará en el estudio de la base teórica sobre la que se sustenta el trabajo.
- **Integración de las tecnologías:** Se establecerán y configurarán todas las tecnologías en un entorno de trabajo, pudiendo desarrollarlo minimizando las dependencias entre ellas.
- **Obtención de los datos:** Se recolectarán los datos provenientes de varios orígenes de datos.
- **Limpieza, agregación y visualización de los datos:** Se transformarán los datos obtenidos para realizar un análisis descriptivo del negocio utilizando visualizaciones que aportan valor al usuario final.
- **Exploración del dato:** Se identificará la forma del dato del proyecto, detectando patrones, relaciones entre características de los datos.
- **Creación del modelo de predicción:** Se diseñará y creará un análisis predictivo a la solución.
- **Despliegue del modelo de predicción:** Se llevará el modelo de análisis predictivo a la solución que utiliza el usuario final.
- **Mejora continua del modelo de predicción:** Se encargará del mantenimiento y mejora del modelo de predicción.
- **Presentación y defensa del Trabajo de Fin de Máster:** Se presentarán los resultados del proyecto.

## 3.6. Herramientas de gestión

La aplicación de una metodología ágil necesita una buena organización y un proceso claramente definido. Para apoyar el desarrollo de la metodología ágil se han decidido utilizar herramientas como Git-Hub y Zen-Hub.

### 3.6.1. Git-Hub

GitHub [16] es un servicio online de control de versiones distribuido. GitHub permite controlar, alojar y revisar código, mejorando la gestión de proyectos y construyendo software de forma colaborativa. Esta herramienta se ha elegido para realizar el control de versiones y mostrar avances continuos del proyecto [17].

Como se puede ver en la Figura 3.4, cuando se realiza un cambio se añade un comentario a los archivos que han sido modificados, dando visibilidad en los ficheros de las actualizaciones que va sufriendo el proyecto.

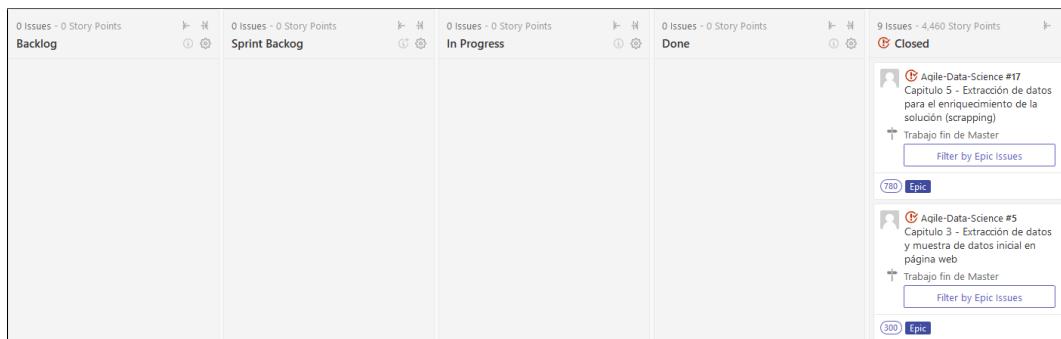
 Rafamg96	Script información aviones añadido	Latest commit 40c4478 27 days ago
 Archivos utiles	Script información aviones añadido	27 days ago
 data	Capítulo 5	6 months ago
 models	Final del proyecto	27 days ago
 web	Final del proyecto	27 days ago
 Comandos a ejecutar	Final del proyecto	27 days ago
 README.md	mostrar datos en página web	6 months ago

**Figura 3.4:** Proyecto de Agile Data Science en GitHub

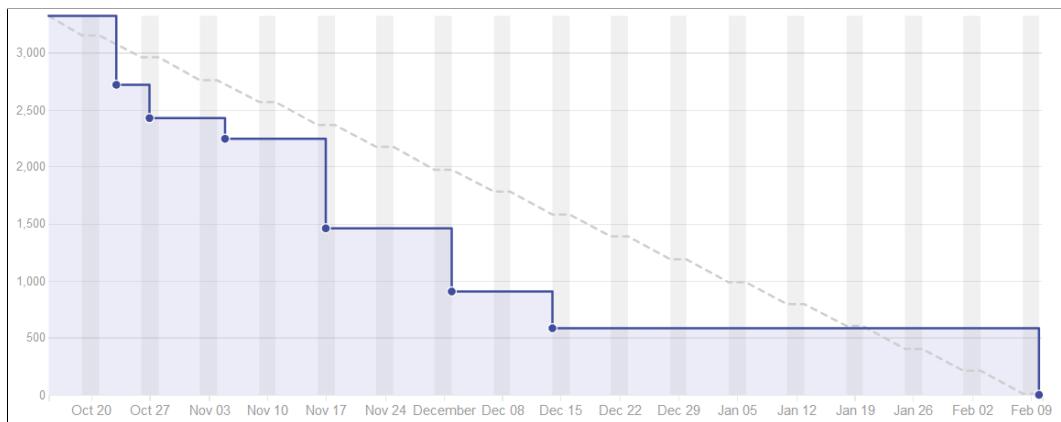
### 3.6.2. Zen-Hub

Como complemento a GitHub, y con el objetivo de realizar un seguimiento del tiempo invertido y del trabajo realizado en este proyecto existen herramientas como Zen-Hub. Zen-Hub ayuda a seguir una metodología de trabajo ágil, proporcionando un tablero Kanban que se asocia con GitHub (Figura 3.5) y gráficos asociados a dicho tablero como el “Burndown” que proporciona una información del trabajo restante en cada iteración.

En este proyecto se puede ver como siempre se ha realizado el trabajo de forma más rápida a lo ideal según el gráfico “Burndown” mostrado en la Figura 3.6, por lo que el proyecto se ha ido desarrollando tal y como se esperaba. Aún así, se puede ver que en Diciembre el trabajo quemado fue nulo debido a un parón en el desarrollo del trabajo.



**Figura 3.5:** Tablero Kanban en ZenHub para el proyecto realizado



**Figura 3.6:** Gráfico Burndown del proyecto realizado

Tras realizar el estudio de la metodología a seguir, cómo aplicarla al contexto de un trabajo de fin de máster y que herramientas apoyan a la implementación de la metodología en el proyecto, ya se puede comenzar con el desarrollo del proyecto siguiendo la metodología definida.

## 4. Sistema para la visualización

---

Para el desarrollo de un sistema de visualización que proporcione información al usuario se necesitan seguir varias fases en las que se itera varias veces para satisfacer las necesidades del usuario: una fase de preparación del trabajo y diseño de la arquitectura de la solución; una fase para implementar las tecnologías implicadas; una fase para la obtención de los datos y la creación de gráficos; y otra fase para desarrollar un modelo de predicción, para su puesta en producción y su mejora continua.

### 4.1. Fase de preparación del trabajo

Antes de comenzar a trabajar en un proyecto es importante definir los modelos de datos que se van a utilizar. Además, tras la definición de los modelos de datos es necesario saber cómo se mostrará la información al usuario y la arquitectura de la solución.

#### 4.1.1. Modelo de datos

Para el planteamiento de la arquitectura es importante decidir qué datos se van a necesitar, y para eso es necesario tener presente la solución final.

La solución final consiste en dar información útil sobre los vuelos en Estados Unidos, para ello, definimos el modelo de los datos como se muestra a continuación:

- **Vuelos:** Contiene todos los datos sobre los vuelos, desde el origen y destino hasta los diferentes retrasos que sufre en cada etapa (seguridad, tiempo atmosférico, etc).
- **Aeropuertos:** contiene toda la información sobre los diferentes aeropuertos que encontramos en Estados Unidos. Encontramos información como la ciudad del aeropuerto, estado, latitud y longitud.
- **Aerolíneas:** contiene toda la información sobre las diferentes Aerolíneas que trabajan en Estados Unidos.
- **Aviones:** contiene información sobre los aviones, como el año de fabricación, dueño, motor, etc.

#### 4.1.2. Modelo de visualización

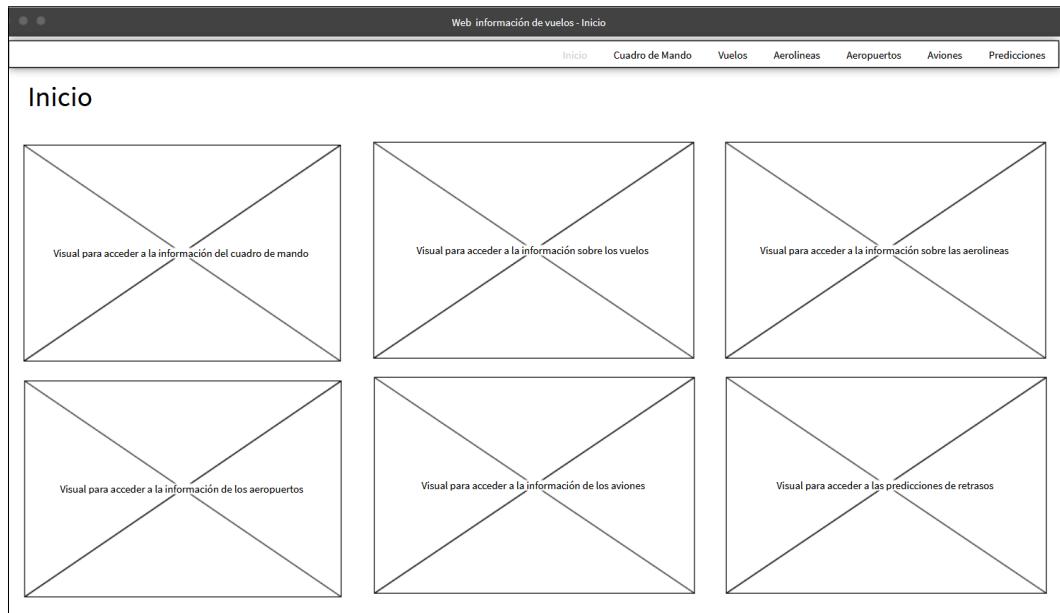
El modelo de visualización indica de qué forma el proyecto de ciencia de datos da valor al usuario. Existen dos opciones importantes para la visualización de datos, que son las más utilizadas: mostrar información al usuario mediante una página web y mostrar información al usuario mediante un cuadro de mando.

La creación de cuadros de mando para los usuarios les permite interactuar con los datos de forma dinámica y responder preguntas del negocio. Su desventaja reside en que están más enfocados al análisis descriptivo.

Por otro lado, la creación de una página web permite añadir más características enfocadas a los análisis predictivos a tiempo real, con la desventaja de que el proceso de la creación de esta visualización es más lento y no se puede iterar tanto con el usuario para realizar un diseño enfocado al mismo. En este proyecto nos hemos centrado en este tipo de visualización debido a que se busca añadir un modulo de predicción a tiempo real.

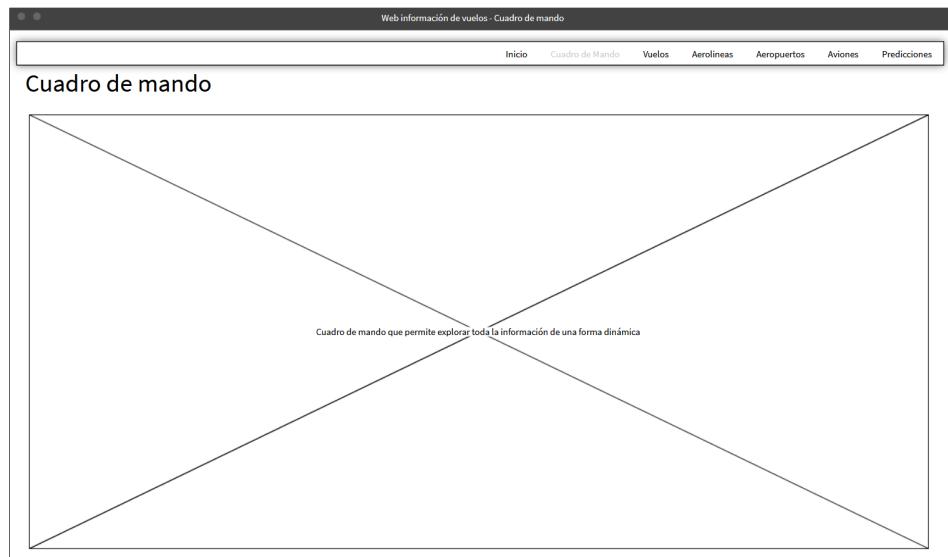
Para la creación de la página web, se han diseñado los siguientes prototipos:

- **Página principal:** En esta página (Figura 4.1) se muestran las posibles opciones del usuario para solicitar información y se proporciona al usuario un menú para que navegue entre las visualizaciones de los datos históricas y futuras.



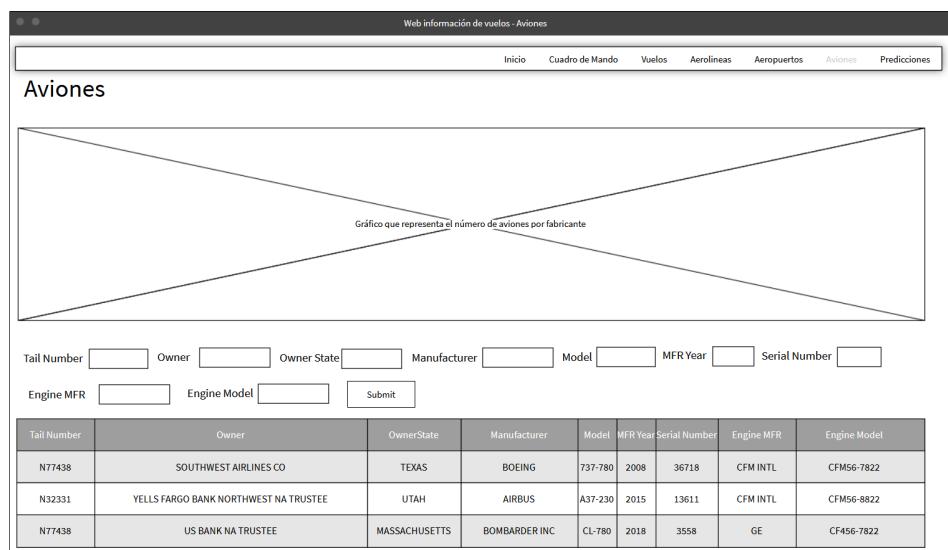
**Figura 4.1:** Página principal

- **Página panel de control:** En esta página (Figura 4.2) se muestra un cuadro de mando que da información útil al usuario, permitiendo la interacción con el mismo.



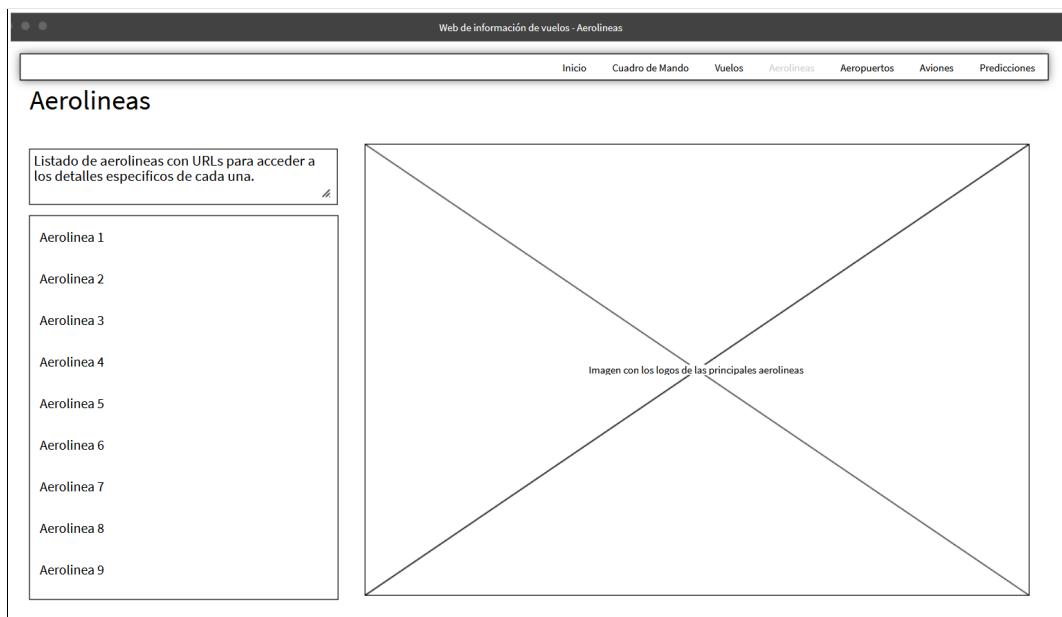
**Figura 4.2:** Página de cuadro de mando

- **Página de búsqueda de aviones:** Esta página (Figura 4.3) contiene la información relacionada con los aviones, mostrando una tabla con todos los aviones registrados, permitiendo buscar aviones específicos y acceder a la página de los mismos.



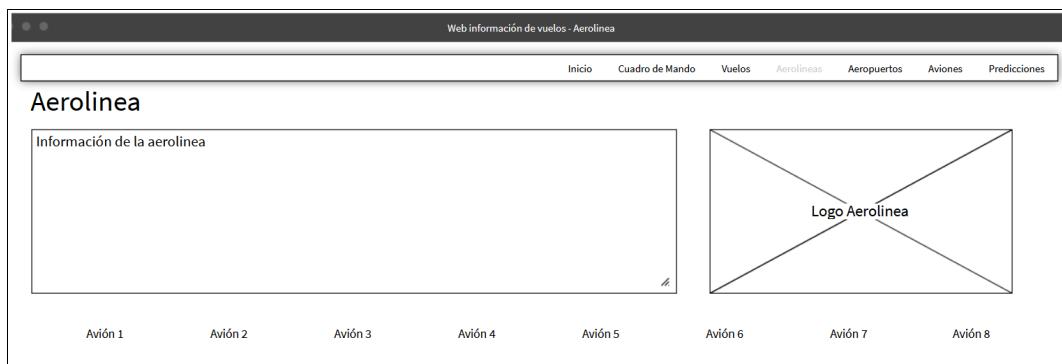
**Figura 4.3:** Página de información sobre los aviones

- 
- **Página sobre las aerolíneas:** En esta página (Figura 4.4) se muestran todas las aerolíneas en formato lista y permite navegar a las páginas específicas de cada una de ellas.



**Figura 4.4:** Página de información sobre las aerolíneas

- **Página de aerolínea:** Esta página (Figura 4.5) contiene información sobre una aerolínea específica y los diferentes aviones que tiene a su disposición, permitiendo acceder a los detalles de los mismos.



**Figura 4.5:** Página de información sobre una aerolínea específica

- **Página del Avión:** Esta página (Figura 4.6) contiene información sobre un avión específico como el año de fabricación, motor, etc. Además se muestra la información de los vuelos realizados por dicho avión.

Serial Number	Manufacturer	Model	MFR Year	Owner	Owner State	Engine Manufacturer	Engine Model
44564	BOEING	737-924ER	2014	UNITED AIRLINES INC	ILLINOIS	CFM INTL	CFM56-7B27E

Carrier	Date	Flight Number	Origin	Destination
UA	2015-07-22	1002	ORD	SAN
UA	2015-05-10	1003	ORD	SFO
UA	2015-05-10	1004	SFO	LAX

**Figura 4.6:** Página de información sobre un avión

- **Página de predicción de retraso del avión:** En esta página (Figura 4.7) se solicitan datos al usuario sobre el vuelo del que se quiere conocer la predicción del retraso y se devuelve el resultado.

**Figura 4.7:** Página de predicciones

---

## 4.2. Integración de las tecnologías

Para el desarrollo del proyecto, tras definir la información a mostrar junto al modelo de visualización, se debe configurar e integrar las siguientes tecnologías en un pipeline y pila de herramientas para desarrollar la solución:

- **Python:** Lenguaje de programación que nos facilita el tratamiento de datos utilizando librerías como Pandas o Spark.
- **Jupyter Notebooks:** Entorno de trabajo interactivo que permite desarrollar código en Python de manera dinámica a la vez que integrar en un mismo documento tanto bloques de código como texto, gráficas o imágenes. Es un SaaS utilizado ampliamente en análisis numérico, estadística y Machine Learning, entre otros campos de la informática y las matemáticas.
- **MongoDB:** Base de datos distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube. Se utilizará esta base de datos como almacenamiento de los documentos tratados.
- **Flask con Jinja:** Tecnologías que permiten la creación de aplicaciones web de forma ágil. Flask es un framework minimalista escrito en Python que permite mostrar la información obtenida a partir del análisis de los datos y predicciones. Jinja complementa a Flask dando facilidades para la creación de múltiples páginas utilizando una misma base.
- **ElasticSearch:** Motor open source de analítica y análisis distribuido para todos los tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y desestructurados. Se utilizará Elasticsearch debido a su capacidad de indexar muchos tipos de contenido aumentando la velocidad de las búsquedas en el sitio web.

## 4.3. Obtención de los datos

Tras integrar las tecnologías y conocer los requisitos de los datos para el desarrollo del proyecto, se han seleccionado las siguientes fuentes de datos:

- **Vuelos:** Se han obtenido a través de la página web de United States Department of Transportation[18].
- **Aeropuertos:** Realizando un tratamiento de datos sobre la información de los vuelos, los aeropuertos origen y destino obtenidos se han enriquecido con datos como la latitud, longitud, ciudad, estado, etc.
- **Aerolíneas:** Se han obtenido a través de la página web de Open Flights [19] en formato csv.
- **Aviones:** Realizando un tratamiento de datos sobre los vuelos, se ha obtenido el número de aviones que han volado. Posteriormente, realizando scrapping a la página web [20], se ha obtenido más información sobre cada uno de ellos.

Posteriormente, se debe diseñar un modelo de datos que use la aplicación, tal y como se muestra en la Figura 4.8.

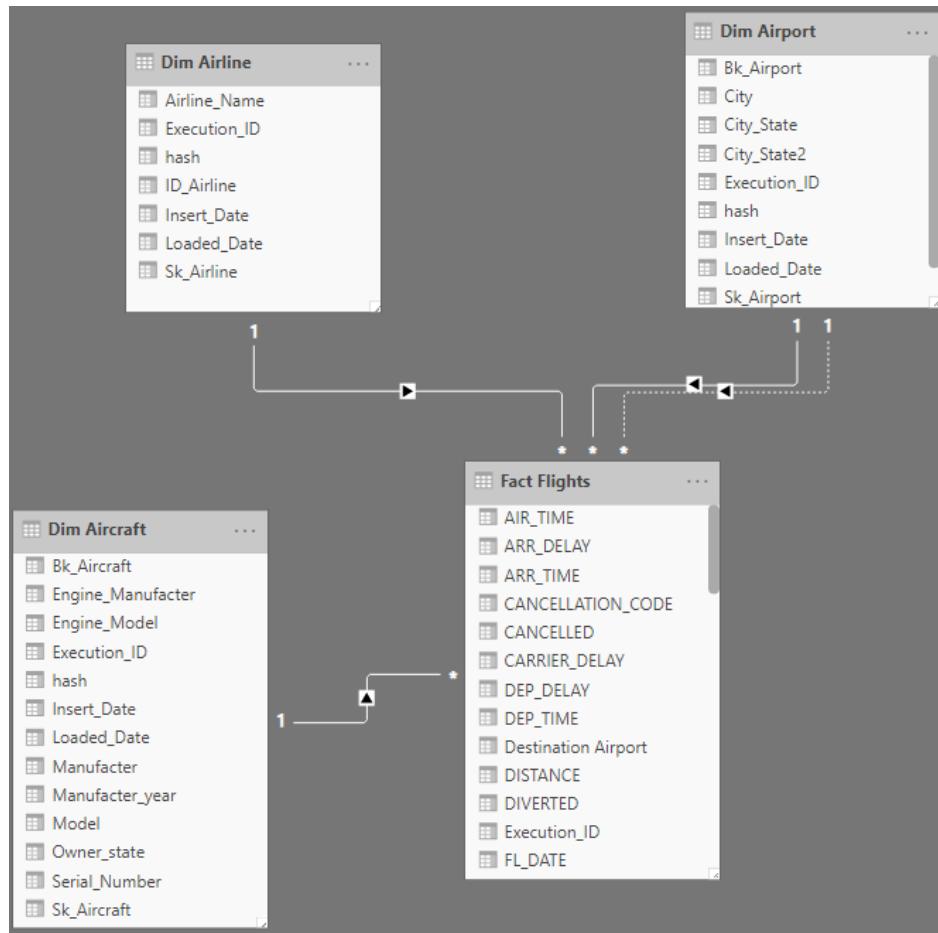


Figura 4.8: Modelado de los datos

#### 4.4. Limpieza, agregación y visualización de los datos

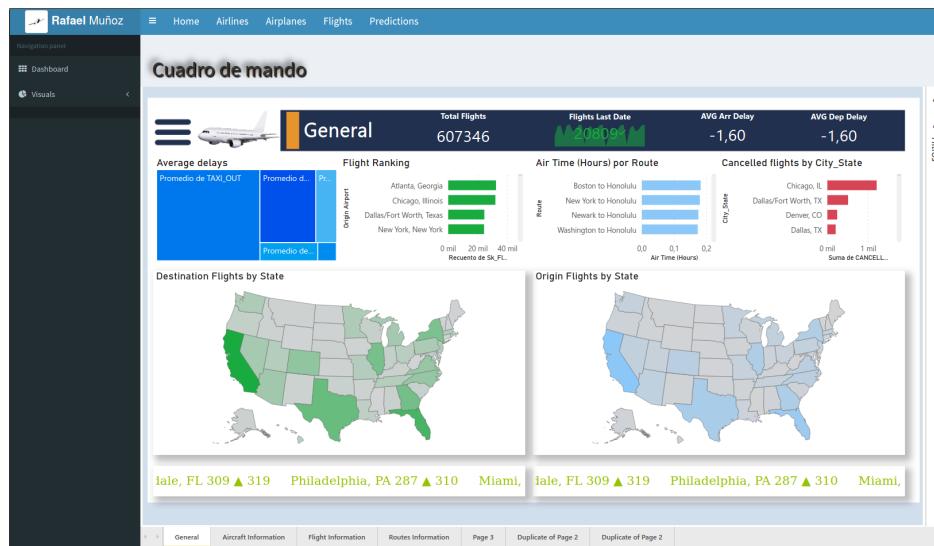
Tras obtener la información de cada una de las estructuras del modelo de datos ha sido necesaria una limpieza de los mismos borrando valores vacíos. Posteriormente, se han diseñado las visualizaciones que serían interesantes para cada apartado de la página detectando si se necesitan datos agregados:

- **Página principal:** Esta página (Figura 4.9) muestra un menú que permite navegar entre las demás páginas de la aplicación.



**Figura 4.9:** Página principal de la aplicación

- **Página del cuadro de mando:** En esta página se mostrará un panel de visuales (Figura 4.10) ofreciendo un cuadro de mando integral al usuario.



**Figura 4.10:** Cuadro de mando para mostrar información al usuario

- **Página sobre las Aerolíneas:** Se mostrará un página (Figura 4.11) con todas las aerolíneas en formato tabla, de forma que no se necesitan datos agregados sino datos detallados.

The screenshot shows a dashboard interface titled "Airlines". The navigation bar at the top includes "Home", "Airlines", "Airplanes", "Flights", and "Predictions". On the left, there is a "Navigation panel" with "Dashboard" and "Visuals" options. The main content area displays a grid of airline codes: F9, HA, AS, VX; DL, AA, OO, B6; EV, US, NK, UA. Below the grid, there is a logo for "UCLM" and some footer text: "Trabajo realizado para el Máster Universitario de Ingeniería Informática Universidad de Castilla-La Mancha Albacete +34 640 25 09 90 rafamg96@gmail.com".

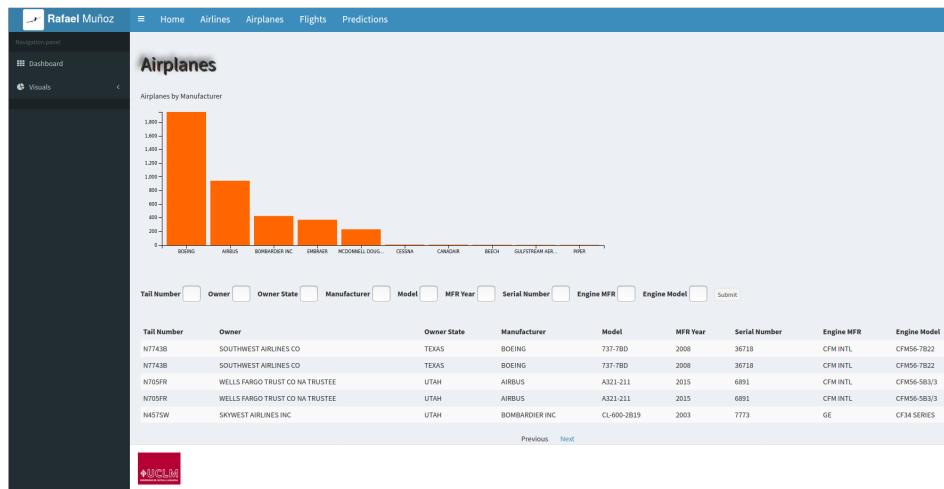
**Figura 4.11:** Página de aerolíneas

- **Página de Aerolínea:** En esta página, se mostrará un visual (Figura 4.4) con toda la información sobre la aerolínea y los diferentes aviones que tiene a su disposición. Se necesitan los datos detallados que relacionan la aerolínea y los aviones.

The screenshot shows a detailed page for Frontier Airlines. The navigation bar at the top includes "Home", "Airlines", "Airplanes", "Flights", and "Predictions". The main content area features the "FRONTIER AIRLINES" logo. Below it, there is descriptive text about Frontier Airlines being an ultra-low-cost carrier based in Denver, Colorado, operating flights to over 100 destinations. A section titled "Fleet: 63 Planes" lists numerous aircraft tail numbers: N261FR, N203FR, N204FR, N205FR, N206FR, N207FR, N208FR, N209FR, N210FR, N211FR, N213FR, N214FR, N216FR, N218FR, N219FR, N220FR, N221FR, N223FR, N227FR, N228FR, N229FR, N230FR, N232FR, N701FR, N702FR, N704FR, N705FR, N902FR, N905FR, N906FR, N910FR, N912FR, N918FR, N923FR, N920FR, N921FR, N922FR, N923FR, N924FR, N925FR, N926FR, N927FR, N928FR, N929FR, N931FR, N932FR, N933FR, N934FR, N935FR, N936FR, N938FR, N939FR, N941FR, N943FR, N947FR, N948FR, N949FR, N951FR, N952FR. At the bottom, there is a footer with the "UCLM" logo and the same contact information as in Figure 4.11.

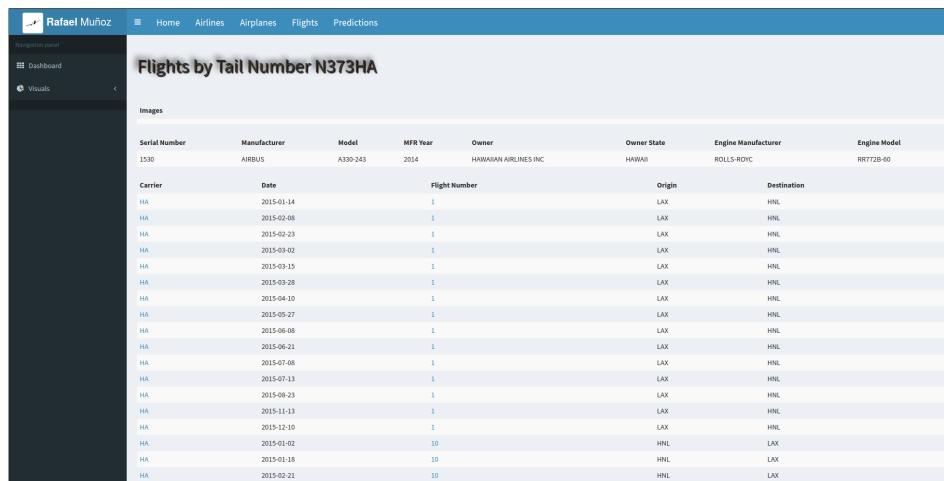
**Figura 4.12:** Página de información sobre una aerolínea

- **Página de los aviones:** En esta página (Figura 4.13), se mostrará un visual en formato de tabla con toda la información de los aviones, por lo que se necesitan datos detallados. Por otro lado se mostrará un visual con la información agregada del número de aviones creados por empresa manufacturera.



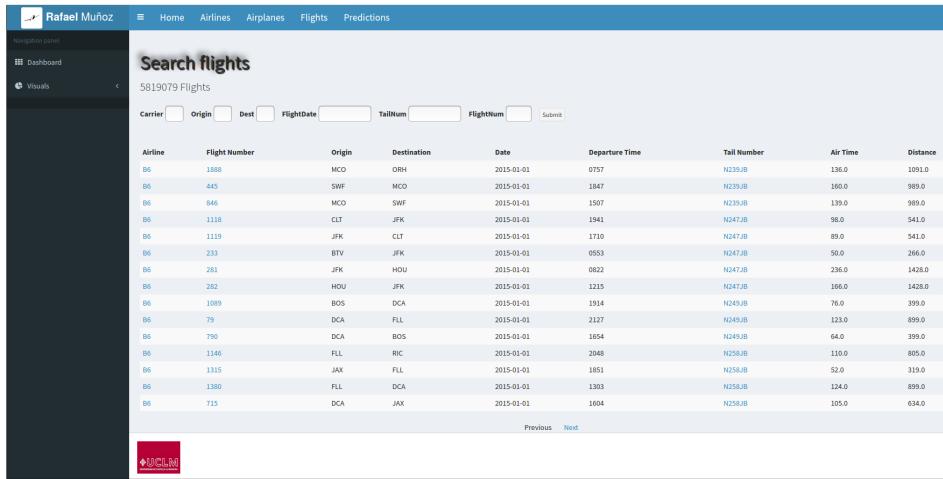
**Figura 4.13:** Página sobre información de los aviones

- **Página del avión:** En esta página (Figura 4.14), se mostrarán los visuales con toda la información sobre el avión y sus vuelos, por lo que se necesitan datos detallados de cada uno de ellos.



**Figura 4.14:** Página sobre información del avión

- **Página de búsqueda de vuelos:** Se mostrará un visual (Figura 4.16) con todas las aerolíneas en formato tabla, de forma que no se necesitan datos agregados sino datos detallados.

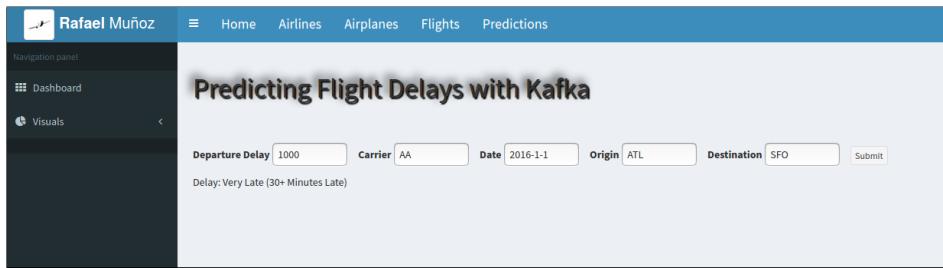


The screenshot shows a web-based application interface titled "Rafael Muñoz". The main header includes "Home", "Airlines", "Airplanes", "Flights", and "Predictions". A navigation panel on the left lists "Dashboard" and "Visuals". The central area is titled "Search flights" and displays a table of 5819079 flights. The table has columns: Airline, Flight Number, Origin, Destination, Date, Departure Time, Tail Number, Air Time, and Distance. The first few rows of data are as follows:

Airline	Flight Number	Origin	Destination	Date	Departure Time	Tail Number	Air Time	Distance
B6	1888	MCO	ORH	2015-01-01	0757	N239JB	136.0	1091.0
B6	445	SWF	MCO	2015-01-01	1847	N239JB	160.0	989.0
B6	846	MCO	SWF	2015-01-01	1507	N239JB	139.0	989.0
B6	1118	CLT	JFK	2015-01-01	1941	N247JB	98.0	541.0
B6	1119	JFK	CLT	2015-01-01	1710	N247JB	89.0	541.0
B6	233	BTV	JFK	2015-01-01	0553	N247JB	50.0	266.0
B6	281	JFK	HOU	2015-01-01	0822	N247JB	236.0	1428.0
B6	282	HOU	JFK	2015-01-01	1215	N247JB	166.0	1428.0
B6	1089	BOS	DCA	2015-01-01	1914	N249JB	76.0	399.0
B6	79	DCA	FLL	2015-01-01	2127	N249JB	123.0	899.0
B6	790	DCA	BOS	2015-01-01	1654	N249JB	64.0	399.0
B6	1146	FLL	RIC	2015-01-01	2048	N258JB	110.0	805.0
B6	1315	JAX	FLL	2015-01-01	1851	N258JB	52.0	319.0
B6	1380	FLL	DCA	2015-01-01	1303	N258JB	124.0	899.0
B6	715	DCA	JAX	2015-01-01	1604	N258JB	105.0	634.0

Figura 4.15: Página de búsqueda de vuelos

- **Página de predicción de retrasos de un vuelo:** Se solicitará al usuario los del vuelo del que se quiere predecir el vuelo y se devolverá un resultado.



The screenshot shows a web-based application interface titled "Rafael Muñoz". The main header includes "Home", "Airlines", "Airplanes", "Flights", and "Predictions". A navigation panel on the left lists "Dashboard" and "Visuals". The central area is titled "Predicting Flight Delays with Kafka" and contains a form with fields: "Departure Delay" (set to 1000), "Carrier" (set to AA), "Date" (set to 2016-1-1), "Origin" (set to ATL), "Destination" (set to SFO), and a "Submit" button. Below the form, a message reads: "Delay: Very Late (30+ Minutes Late)".

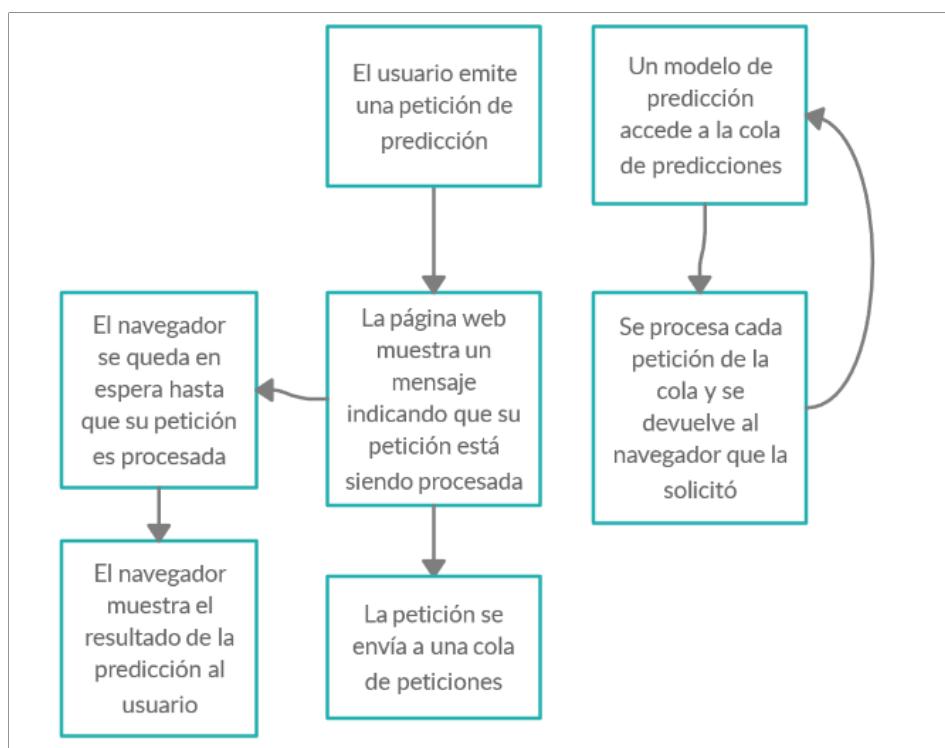
Figura 4.16: Página de predicciones de retrasos

Tras diseñar las visualizaciones necesarias para el proyecto, se procede a crear las visualizaciones que necesiten datos detallados, pues son las más sencillas ya que no se necesita tratamiento de los datos, y posteriormente, transformar estos datos en agregados según el requerimiento de cada uno de los visuales.

#### 4.4.1. Arquitectura del proyecto de ciencia de datos

Tras crear la visualización de los datos que proporcionan el análisis descriptivo al usuario, se debe definir la forma en la que el usuario va a realizar sus predicciones de la forma más sencilla posible, aumentando la usabilidad y accesibilidad de la solución. La arquitectura básica de funcionamiento debe seguir los siguientes pasos, tal y como se muestra en la Figura 4.17:

1. El usuario realiza una petición.
2. La página web muestra un mensaje de que su solicitud se está procesando.
3. La página web envía la petición a una cola de mensajes donde se almacenan las peticiones.
4. Un proceso comprueba cada 5 segundos si existe alguna petición en la cola de mensajes y devuelve las predicciones de cada una de las peticiones a la página web.
5. La página web muestra el resultado de la predicción al usuario.

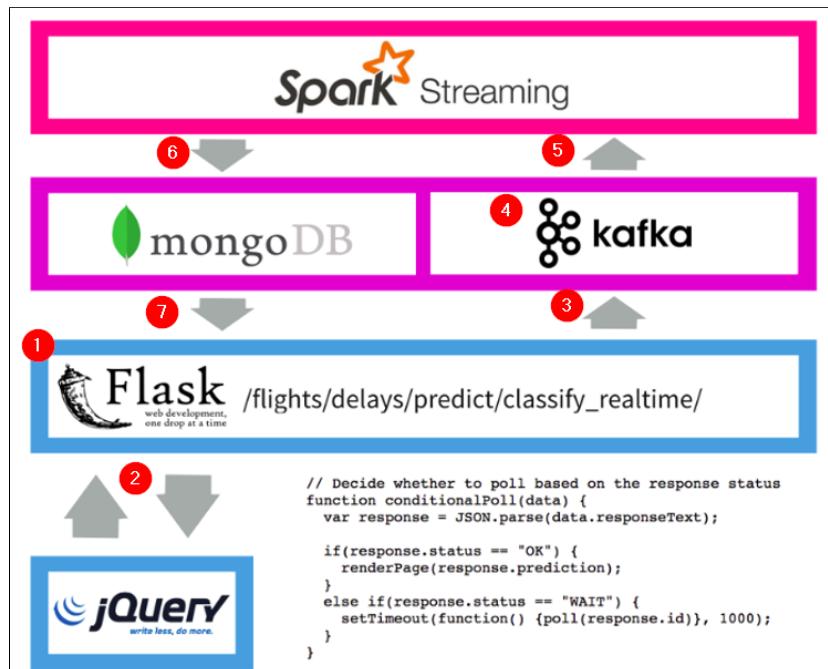


**Figura 4.17:** Diagrama funcional de la obtención de predicciones

Tras definir la forma en la que el usuario utilizará el modelo de predicción, se debe definir cómo trabajarán las diferentes tecnologías en el proyecto para lograr ese resultado.

El diagrama de arquitectura mostrado en la Figura 4.18 indica la forma en la que las tecnologías interactuarán entre ellas donde el funcionamiento sería el siguiente:

1. El usuario emitiría una petición de predicción en la página web montada con el framework Flask.
2. El módulo de Flask realiza una llamada al módulo de Jquery para mostrar en la página web que la predicción se está calculando y, posteriormente, se queda esperando el resultado de la predicción para mostrarla al usuario, esta espera suele tardar entre 1 y 5 segundos.
3. Flask realiza una llamada a Kafka para añadir a una cola de eventos las peticiones de predicción.
4. Kafka almacena la petición en una cola de mensajes.
5. SparkStreaming consulta cada 5 segundos la cola de peticiones de predicción de Kafka y genera las predicciones.
6. Spark Streaming almacena la predicción en la base de datos MongoDB.
7. Spark Streaming envía la predicción resuelta a Flask para mostrársela al usuario mediante JQuery.



**Figura 4.18:** Diagrama de la arquitectura de la solución

Como se ha visto en el diagrama de arquitectura de la figura 4.18, se necesitan nuevas tecnologías que integrar en la solución y que son el siguiente paso a implementar:

- 
- **Spark MLlib:** Librería para desarrollar proyectos de Machine Learning de forma paralela. Se utilizará debido a la gran cantidad de datos de los que se dispone.
  - **SparkStreaming:** Extensión del núcleo del Spark API, que permite el procesamiento escalable, de alto rendimiento y tolerante a fallos de flujos de datos en tiempo real. Los datos ingeridos por esta extensión pueden ser Kafka, Flume, Kinesis, o sockets TCP, y puede procesar dichos datos utilizando algoritmos complejos.
  - **Apache Kafka:** Plataforma unificada, de alto rendimiento y baja latencia para la manipulación en tiempo real de fuentes de datos. Se podría decir que es una cola de mensajes bajo el patrón publicación-suscripción masivamente escalable, nacida como un registro de transacciones distribuidas. La utilizaremos principalmente para enviar peticiones de predicción a nuestro módulo de predicción en tiempo real.
  - **Jquery:** Librería en JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Se utilizará esta tecnología para crear peticiones de predicciones y, una vez obtenida, mostrarlo en la página web [21].

## 4.5. Exploración de datos y creación del modelo de predicción

Siguiendo la pirámide de valor de los proyectos de ciencia de datos, tras extraer y mostrar información en la aplicación, uno de los últimos incrementos en la forma de entregar valor al usuario es el desarrollo de modelos predictivos. Para ello, se utilizarán los datos históricos para entrenar el modelo, y previamente se deben explorar los datos para limpiarlos de anomalías y otros inconvenientes que puedan hacer que el entrenamiento del modelo de lugar a predicciones erróneas.

Posteriormente, para crear un buen modelo predictivo que estime futuros retrasos en los vuelos se deben diseñar y crear varios modelos candidatos, y posteriormente realizar pruebas para evaluar cada uno de ellos, comprobando cuál tiene un rendimiento mejor en base a la matriz de confusión y la tasa de acierto. Para ello, los modelos se validan siguiendo el método “train/test split”, según el cual los datos históricos se dividen en un 80 % para entrenar al modelo y un 20% para evaluarlo, teniendo en cuenta al crear las particiones que los datos que se quieren predecir son posteriores a los históricos, por lo que los de entrenamiento deben ser anteriores a los que se utilizarán para la evaluación.

Una vez validados los modelos de predicción, se eligen los mejores y se perfeccionan modificando los hiper-parámetros. Tras ello, se opta por el que tenga un rendimiento superior y cumpla los requisitos para su puesta en producción.

## 4.6. Despliegue del modelo de predicción

Para el despliegue del modelo seleccionado en producción, éste se debe almacenar previamente en un fichero utilizando la librería Pickle de Python, para posteriormente, implantarlo en la web a la que tendrá acceso el usuario.

Para la utilización del modelo se debe pensar cómo se realizarán las predicciones. En este proyecto se ha optado por crear un apartado en la página web que solicita datos al usuario sobre el vuelo del que se quiere predecir el retraso.

Las 3 posibles soluciones que se podrían implementar para el uso de un predictor con sus virtudes y defectos son las siguientes:

- **Predicción en tiempo real utilizando un servicio web y Scikit-Learn:** Es la solución más sencilla. Se crea un servicio web que llame al modelo de predicción, creado con la librería Scikit-Learn, dando la predicción instantáneamente con el inconveniente de que no permite escalar horizontalmente. Esta solución, en entornos con Big Data, no sería una opción viable.
- **Predicción en lotes (batch) utilizando SparkML:** Esta solución almacena las peticiones de predicción de los usuarios en una colección de la base de datos, de forma que se programa la ejecución del modelo de predicción para que cada cierto tiempo se resuelvan las peticiones de los usuarios y se almacenen en la colección de la base de datos. Posteriormente, cuando un usuario realice una petición, se detecta si ya ha sido realizada y en el caso de que así fuese, se mostraría. En otro caso se almacenaría en la colección de la base de datos para predecir.
- **Predicción en tiempo real utilizando Kafka, SparkML y SparkStreaming:** Esta solución es la más completa y compleja. El usuario realiza una petición utilizando la página web y ésta se envía a un módulo de Kafka y se almacena en una cola de predicciones. Posteriormente, el usuario recibe un mensaje donde se le indica que la petición está siendo procesada y, mediante Jquery, se quedaría esperando una respuesta. Por otro lado, un componente de SparkStreaming realiza continuamente solicitudes a Kafka para recibir las peticiones de predicción, procesarlas y enviarlas a la página web, de forma que JQuery mostraría el resultado. Esta es la solución que ofrece las predicciones casi a tiempo real en un entorno Big Data como el que se presenta en este proyecto.

En este proyecto se ha elegido utilizar la tercera opción, puesto que se busca realizar una predicción a tiempo real y tenemos muchos datos como para poder almacenarlos en memoria para el entrenamiento. La arquitectura de esta solución se puede ver en la Figura 4.18.

## 4.7. Mejora continua del modelo de predicción

Tras haber desplegado el modelo de predicción en producción, se puede y debe seguir mejorando. Se debe entrenar frecuentemente para que se adapte a los cambios de los datos según pase el tiempo siguiendo el esquema mostrado en la Figura 4.19.

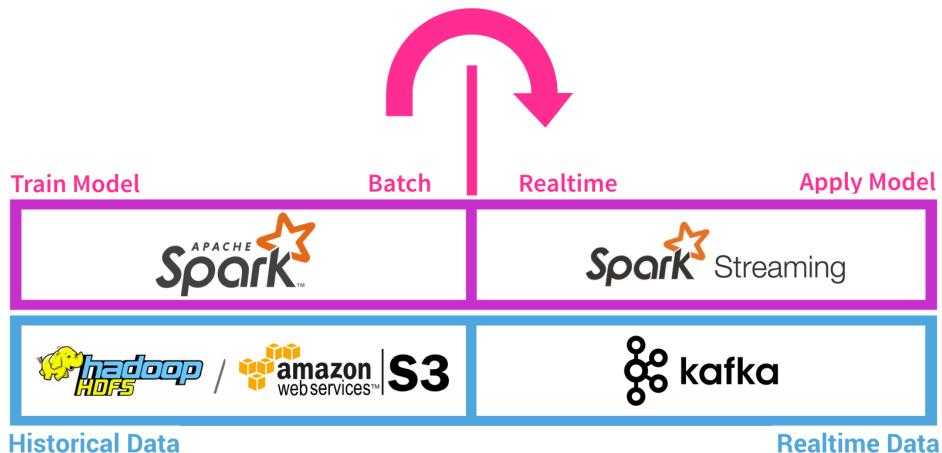


Figura 4.19: Ciclo de vida del modelo de predicción

Para ello, se pueden utilizar algunas librerías como Airflow que permiten establecer actividades programadas siguiendo una estructura de grafos cíclicos dirigidos, de forma que se puedan ejecutar diferentes acciones seguidas de otras en el lenguaje de programación Python.

En este proyecto se utiliza Airflow para programar el entrenamiento del modelo y que se ejecute diariamente con los datos históricos. De esta forma el modelo estará actualizado y realizará las predicciones más exactas.

Resumiendo, en este capítulo se ha tratado el desarrollo de un sistema de visualización que proporciona un análisis descriptivo de los datos mediante: el diseño de los datos; el prototipado del sistema de visualización; la integración de las tecnologías; la obtención de datos desde diversas fuentes; su limpieza y creación de visualizaciones. Posteriormente, se ha tratado la implementación del análisis predictivo a la solución mediante: el diseño de una arquitectura de funcionamiento para la obtención de predicciones y la creación del modelo de predicción, su despliegue y mejora continua del mismo.

## 5. Conclusiones y propuestas

---

### 5.1. Conclusiones

El trabajo realizado se ha centrado en el estudio, planteamiento y desarrollo de varias soluciones para un proyecto de ciencia de datos siguiendo una metodología ágil. Se ha comprobado que existen diversas soluciones, cada una enfocada a un tipo concreto de problema a resolver, por lo que es importante definir su arquitectura en base al volumen de los datos y la velocidad de las predicciones.

Se ha aprendido una gran cantidad de conceptos relativos a la ciencia de datos, concretamente cómo lidiar con un problema de gran volumen de datos utilizando Spark.

Se ha aprendido a implementar una gran cantidad de tecnologías que interactúan unas con otras para crear una arquitectura compleja y óptima para el problema que nos concierne.

### 5.2. Competencias cubiertas

Se destaca la utilización de técnicas de las asignaturas de “Desarrollo de Sistemas Inteligentes”, “Dirección IT y Gestión de la Innovación”, “Servicios de Computación de Altas Prestaciones y Disponibilidad” y “Planificación y gestión de infraestructuras TIC”. Concretamente, se ha utilizado la asignatura “Desarrollo de Sistemas Inteligentes” para la creación de modelos de Machine Learning aprendidos en clase, como la regresión lineal o Random Forest, siendo estos conocimientos ampliados mediante el entrenamiento de modelos de forma distribuida mediante la computación en cluster de Spark. Además, se ha utilizado Python y Jupyter Notebook como entorno de trabajo para el tratamiento de datos.

Por otro lado, las metodologías estudiadas en la asignatura de “Dirección IT y Gestión de la Innovación” han sido la base del proyecto, pues estas metodologías han sido evaluadas en

---

un proyecto de ciencia de datos y se ha visto la necesidad de modificar las bases de las mismas aportando novedad a un área específica de la informática: en un proyecto software el cliente espera la entrega de valor a través de nuevas funcionalidades. En cambio, en un proyecto de ciencia de datos el cliente recibe valor mediante entrega de datos, los cuales se ven transformados iterativamente en diferentes formas y mejoras de calidad: registros, tablas, gráficos, predicciones y, finalmente, toma de decisiones apoyada por inteligencia artificial.

La asignatura “Servicios de Computación de Altas Prestaciones y Disponibilidad” ha tenido cabida en el trabajo debido a la necesidad del paralelismo para la creación de un modelo de predicción por la gran cantidad de datos.

La asignatura ‘Planificación y gestión de infraestructuras TIC’ ha aportado la visión del diseño de la arquitectura solución para la resolución de un problema específico junto con posibles mejoras que se tratarán a futuro.

Este trabajo también ha estado enfocado al diseño de una arquitectura para la resolución de un problema específico para lo cual ha sido necesario el conocimiento obtenido de forma autodidacta apoyado en otras asignaturas de forma transversal.

### 5.3. Trabajo futuro

Para mejorar el trabajo realizado, se han pensado en las siguientes propuestas para el futuro.

Tras diseñar el modelo de los datos y realizar experimentos sobre las características a utilizar en el modelo de aprendizaje automático, se procedería a cambiar el modo de almacenamiento de los datos para que se almacenen en una base de datos relacional ya que encajaría más con el problema actual donde todos los datos son claramente estructurados y no tienen atributos cambiantes.

Por otro lado, para automatización de las tareas de extracción y limpieza de nuevos datos, sería conveniente añadir un procedimiento que sea capaz de almacenar incrementalmente en la base de datos los datos nuevos una vez al mes. Para ello se crearía un script en el lenguaje de programación Python que obtuviese el nuevo archivo de vuelos de dicho mes, posteriormente se detectarían sobre ellos los aviones, aeropuertos, aerolíneas y empresas encargadas de la creación de nuevos aviones que no existen en las tablas correspondientes de la base de datos y se procedería a enriquecer los datos para insertarlos correctamente en la base de datos. De esta forma se obtendría una solución escalable y mantenible en el tiempo. Una herramienta para realizar esto sería Integration Services de Microsoft.

Una vez que se hubiese creado la solución continua se buscaría crear un cuadro de mando que permita al usuario dar visibilidad de los datos del negocio de forma dinámica utilizando una herramienta para la visualización de este tipo como Microsoft Power BI.

Por último, se estudiará cómo desplegarlo en la nube eliminando problemas con la integración de las tecnologías, proporcionando más disponibilidad y reduciendo costes de mantenimiento.

## 5.4. Opinión personal

Este trabajo me ha permitido unir muchos de los conceptos que se han visto por separado en el Grado de Ingeniería Informática y que han sido unidos cursando el Máster de Ingeniería Informática. El Máster de Ingeniería Informática ha ayudado en gran medida a realizar este trabajo que ha dado la última pincelada de cómo todas las asignaturas estudiadas pueden estar relacionadas en mayor o menor medida, viendo cómo el diseño de una arquitectura y una metodología puede ser muy importante en el desarrollo de un proyecto de ciencia de datos.



## Bibliografía

---

- [1] Jurney Russell. *Agile Data Science 2.0 - Building Full-Stack Data Analytics applications with Spark*. 2017. ISBN: 9781491960110.
- [2] Python Software Fundation. *Python*. URL: <https://www.python.org/>.
- [3] Inc. MongoDB. *MongoDB*. URL: <https://www.mongodb.com/es>.
- [4] Pallets Projects. *Flask*. URL: <https://palletsprojects.com/p/flask/>.
- [5] Pallets Projects. *Jinja*. URL: <https://palletsprojects.com/p/jinja/>.
- [6] Elasticsearch B.V. ¿Qué es Elasticsearch? URL: <https://www.elastic.co/es/what-is/elasticsearch>.
- [7] Apache Software Foundation. *Introduction to Kafka*. URL: <https://kafka.apache.org/intro>.
- [8] Apache Software Foundation. *Spark Streaming Programming Guide*. URL: <https://spark.apache.org/docs/latest/streaming-programming-guide.html>.
- [9] UC Berkeley School of Information. *What is Data Science?* URL: <https://datascience.berkeley.edu/about/what-is-data-science/>.
- [10] empresa del grupo Salesforce Tableau Software LLC. ¿Qué es la inteligencia de negocios y por qué es importante? URL: <https://www.tableau.com/es-es/learn/articles/business-intelligence>.
- [11] Universidad Complutense de Madrid. ¿Qué es el Big Data? URL: <https://www.masterbigdataucm.com/que-es-big-data/>.
- [12] Apache Software Fundation. *Apache Spark*. URL: <https://spark.apache.org/>.
- [13] Inc Anaconda. *Comparison Spark vs Dask*. URL: <https://docs.dask.org/en/latest/spark.html>.
- [14] Databricks. *Databricks*. URL: <https://databricks.com/>.
- [15] Economipedia. *Pirámide de Maslow*. URL: <https://economipedia.com/definiciones/piramide-de-maslow.html>.

- 
- [16] GitHub. *GitHub*. URL: <https://github.com/>.
  - [17] Rafael Muñoz González. *Agile Data Science GitHub project*. URL: <https://github.com/Rafamg96/Agile-Data-Science/>.
  - [18] United States Department of Transportation. *Reporting Carrier On-Time Performance (1987-present)*. URL: [https://www.transtats.bts.gov/DL\\_SelectFields.asp?Table\\_ID=236&DB\\_Short\\_Name=On-Time](https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time).
  - [19] Open Flights. *Airport, airline and route data*. URL: <https://openflights.org/data.html>.
  - [20] Federal Aviation Administration. *Federal Aviation Administration Aircraft Registry*. URL: [https://registry.faa.gov/aircraftinquiry/Aircraft\\_Inquiry.aspx](https://registry.faa.gov/aircraftinquiry/Aircraft_Inquiry.aspx).
  - [21] Jquery Foundation. *Jquery*. URL: <https://jquery.com/>.

## Contenido del enlace de descarga de contenidos

---

El contenido del este enlace de descarga de contenidos sustituye al CD debido a la situación actual , este enlace acompaña a la memoria y podemos encontrar los siguientes recursos:

- Memoria del trabajo en formato PDF.
- Código fuente del trabajo dentro del directorio Código fuente.
- Páginas Web que han servido de bibliografía. Las podemos encontrar dentro del directorio Bibliografia/Enlaces Web.