



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA DE SEGOVIA

**Grado en Ingeniería Informática
de Servicios y Aplicaciones**

**Análisis de marcas de productos basado en Twitter
utilizando técnicas de PNL y la teoría de grafos**

Alumno: Alejandro Barrio Mateos

**Tutores: José Vicente Álvarez Bravo
Silvia Duque Moro**

Fecha: 30 de junio de 2023

Análisis de marcas de productos basado en Twitter utilizando técnicas de PNL y la teoría de grafos

Alejandro Barrio Mateos

30 de junio de 2023

Índice general

Lista de figuras	v
Lista de tablas	vii
Lista de código	ix
Resumen	xiii
Abstract	xv
I Descripción del proyecto	1
1. Introducción	3
1.1. Planteamiento del problema	4
1.2. Objetivos del trabajo	5
1.2.1. Restricciones	6
1.3. Contexto empresarial	6
1.4. Estructura de la memoria	7
2. Planificación	9
2.1. Metodología de trabajo	9
2.1.1. Metodología de trabajo en <i>data science</i>	11
2.2. Planificación temporal	13
2.2.1. Sprint #1	13
2.2.2. Sprint #2	14
2.2.3. Sprint #3	15
2.2.4. Sprint #4	16
2.2.5. Sprint #5	16
2.3. Presupuestos	17
2.3.1. Aproximación a la estimación de costes	17
2.3.2. Recursos técnicos	19
2.3.3. Coste final	20
2.4. Balance temporal y económico	20

3. Contexto del trabajo	23
3.1. Entorno específico	23
3.1.1. Entorno específico de la empresa	23
3.1.2. Entorno específico del estudio	26
3.2. Entorno de negocio	28
3.2.1. Análisis de datos sociales	28
3.2.2. Casos de aplicación	29
3.3. Contexto científico-técnico	31
3.3.1. <i>Machine Learning</i>	31
3.3.2. Procesamiento del Lenguaje Natural	31
3.4. Estado del arte	32
3.4.1. <i>Web scraping</i> de Metacritic	33
3.4.2. <i>Topic modelling</i> de un conjunto de tweets	34
3.4.3. <i>Sentiment analysis</i> de reseñas de Amazon	35
II Desarrollo de la propuesta y resultados	37
4. Data extraction	39
4.1. APIs	39
4.1.1. Twitter	41
4.1.2. Twitter API	43
4.1.3. Caso de estudio elegido	44
4.2. Web scraping	46
4.2.1. Metacritic	47
4.2.2. Caso de estudio elegido	48
4.3. Estructura de los datos	50
4.3.1. Twitter	50
4.3.2. Metacritic	52
5. Preprocesado de datos	55
5.1. Data cleaning	56
5.2. Language filtering	56
5.3. Tokenization	57
5.4. Stop words	57
5.5. Stemming y lemmatization	58
6. Topic modelling	59
6.1. Exploratory Data Analysis	60
6.1.1. WordCloud	61
6.1.2. Resultados	62
6.2. Latent Dirichlet Allocation	64
6.2.1. N-gramas	64
6.2.2. TF-IDF	65

6.2.3.	LDA	66
6.2.4.	Implementación y resultados	68
6.3.	Biterm Topic Model	72
6.3.1.	Entrenamiento del modelo y métricas	73
6.3.2.	Resultados	75
7.	Sentiment analysis	77
7.1.	Modelos preentrenados	78
7.1.1.	BERT	78
7.1.2.	VADER	80
7.1.3.	TextBlob	80
7.2.	Modelos específicos	81
7.2.1.	Naive Bayes	81
7.2.2.	Entrenamiento del modelo	82
7.2.3.	Evaluación del modelo	83
7.3.	Resultados	85
7.3.1.	Tweets	85
7.3.2.	Reseñas de prensa especializada	87
7.3.3.	Reseñas de usuarios	88
8.	Conclusiones y trabajo futuro	91
8.1.	Conclusiones	91
8.1.1.	Conclusiones del estudio	91
8.1.2.	Conclusiones personales	93
8.2.	Trabajo futuro	93
III	Apéndices	95
A.	Manual de Instalación	97
B.	Contenido adjunto	99
Bibliografía		101
Webgrafía		103

Índice general

Índice de figuras

1.1.	Estadísticas de datos generados diariamente durante 2021	4
2.1.	Pasos para la extracción, procesado y análisis de datos	11
3.1.	Modelo de Porter	25
3.2.	De los datos sin procesar a la información semántica	28
3.3.	Puntuaciones de prensa en Metacritic	33
3.4.	Puntuaciones de usuarios en Metacritic	33
3.5.	Proporción de juegos según su clasificación por edades para cada consola .	34
3.6.	<i>Wordcloud</i> con las cuentas de usuario más mencionadas en los tweets . .	35
3.7.	Valoración de las reseñas en función de la longitud de las mismas . .	36
4.1.	<i>Dashboard</i> de una cuenta de desarrollador de Twitter	41
4.2.	Ejemplo de página de resumen de un videojuego	47
4.3.	Esquema del DOM de las reseñas de usuarios	49
4.4.	Formato y datos de un usuario	50
4.5.	Dataframe con las reseñas de usuarios de Fire Emblem Engage	52
5.1.	Reseñas de usuarios de Hi-Fi Rush en Xbox	56
6.1.	Esquema básico de funcionamiento de un proceso de <i>topic modelling</i> . . .	59
6.2.	Esquema real del desarrollo de un proyecto de <i>data science</i>	60
6.3.	<i>Word cloud</i> de los tweets de Nintendo	61
6.4.	<i>Word cloud</i> de los tweets de Xbox	62
6.5.	<i>Word cloud</i> de las reseñas de Forspoken en PS5	63
6.6.	Ejemplo básico de embebimiento para la detección de significados similares	65
6.7.	Representación gráfica de la generación a partir de LDA. El primer cuadro indica que el proceso se realiza sobre el conjutno de documentos disponible, mientras que el segundo hace lo propio para cada documento (elección de temas y palabras)	67
6.8.	Visualización del modelo entrenado con tweets sobre Nintendo, considerando 10 temas	69
6.9.	Modelo de Nintendo resaltando el cuarto tema y el término <i>walmart</i> . . .	69

Índice de figuras

6.10. Términos más relevantes en las reseñas de cada videojuego en las consolas de cada compañía (FE Engage para Switch, Hi-Fi Rush para Xbox Series X y Forspoken para PS5; respectivamente)	70
6.11. Gráfico de burbuja relativo a Xbox al resaltar el término <i>modded</i>	71
6.12. Representación gráfica de (a) LDA, (b) mezcla de unigramas y (c) BTM. Por claridad, no se representan los hiperparámetros fijados α y β	73
6.13. Consola de visualización de resultados del modelo relativo a los tweets que mencionan el videojuego Forspoken, seleccionando el tema 4 para analizar.	76
7.1. Idea intuitiva detrás del clasificador de Bayes y la bolsa de palabras	81
7.2. Matriz de confusión del modelo entrenado con reseñas de usuarios para predecir la categoría de una crítica	84
7.3. Sentimiento detectado según las métricas definidas para RoBERTa y VADER en los tweets que mencionan a Nintendo, Playstation y Xbox	85
7.4. Sentimiento detectado según las métricas definidas para RoBERTa y VADER en los tweets que mencionan a Fire Emblem Engage, Forspoken y Hi-Fi Rush	86
7.5. Muestra de algunos tweets y la clasificación otorgada por cada modelo preentrenado.	86
7.6. Comparativa entre la polaridad del texto y la nota dada por prensa especializada a Fire Emblem Engage (Switch), Forspoken (PS5) y Hi-Fi Rush (Xbox Series X)	87
7.7. Distribución de la subjetividad detectada en las críticas de prensa especializada de Fire Emblem Engage (Switch), Forspoken (PS5) y Hi-Fi Rush (Xbox Series X)	88
7.8. Comparativa entre la polaridad del texto y la nota dada por usuarios a Fire Emblem Engage (Switch), Forspoken (PS5) y Hi-Fi Rush (Xbox Series X) .	88
7.9. Gráficos de caja que indican la polaridad de los textos según el tipo de reseña	89
7.10. Distribución de la subjetividad detectada en las críticas de usuarios de Fire Emblem Engage (Switch), Forspoken (PS5) y Hi-Fi Rush (Xbox Series X) .	89
7.11. Comparativa entre los valores de polaridad y subjetividad detectados por TextBlob en las reseñas de usuarios de Fire Emblem Engage (Switch), Forspoken (PS5) y Hi-Fi Rush (Xbox Series X)	90
7.12. Comparativa entre la utilidad de las reseñas del videojuego Fire Emblem Engage y la polaridad detectada por VADER, así como la categoría en la que se encuadran	90
A.1. Cabecera de un archivo ejecutado por Jupyter Notebook	98

Índice de cuadros

2.1.	Primera historia de usuario	14
2.2.	Segunda historia de usuario	14
2.3.	Tercera historia de usuario	15
2.4.	Cuarta historia de usuario	16
2.5.	Quinta historia de usuario	17
2.6.	Aproximación a la estimación de los costes por sprint usando puntos de historia	19
2.7.	Recursos técnicos	20
2.8.	Diagrama de Gantt final con las fechas en que se desarrolló el proyecto . .	21

Índice de cuadros

Índice de código

4.1.	Ejemplo de configuración de cliente Tweepy	44
4.2.	Ejemplo de obtención de los tweets relativos a una query especificada	45
5.1.	Diferencia entre <i>stemming</i> y <i>lemmatization</i>	58
7.1.	Ejemplo de asignación de puntuaciones de sentimiento a un texto	78
7.2.	Puntuaciones y porcentajes (negativo, neutro y positivo; respectivamente) asignadas por RoBERTa a un tweet.	79
A.1.	Comandos para instalar todos los paquetes requeridos por el proyecto	98

Índice de código

*A mi abuelo Abelardo,
por haber sido mi compinche durante 22 maravillosos años;
a mis padres, José Antonio y Ascensión,
porque todo lo que soy hoy es gracias a vosotros;
y a todos mis amigos,
por estar ahí siempre que os necesito.*

Resumen

Hoy en día, los medios sociales tienen una gran influencia tanto en la sociedad como en los negocios, pues permiten a las marcas conocer la opinión de sus clientes respecto a sus productos y servicios, dándoles la oportunidad de mejorar sus estrategias de marketing y comunicación, entre otras. De este modo, los medios sociales se han convertido en una importante fuente de información y análisis.

En este contexto, el objetivo de este TFG es llevar a cabo un análisis de la información publicada en Twitter asociada a un subconjunto de marcas, realizando una comparativa de su presencia e impacto en redes sociales. Para ello, se pondrá el foco en el análisis de la información textual recogida tanto por redes sociales, como a través de páginas de reseñas de productos, haciendo un estudio de los temas a tratar así como del sentimiento asociado a dichos textos.

Palabras claves: *social media analytics, web scraping, Twitter, procesamiento de lenguaje natural, topic modelling, sentiment analysis.*

Abstract

Nowadays, social media have a great influence on both society and business, as they allow brands to know what their customers think about their products and services, giving them the opportunity to improve their marketing and communication strategies, among others. In this way, social media have become an important source of information and analysis.

In this context, the aim of this project is to carry out an analysis of the information published on Twitter associated with a subset of brands, making a comparison of their presence and impact on social networks. To do this, the focus will be on the analysis of textual information collected both by social networks and through product review pages, making a study of the topics to be addressed as well as the sentiment associated with these texts.

Palabras claves: social media analytics, web scraping, Twitter, natural language processing, topic modelling, sentiment analysis.

Parte I

Descripción del proyecto

Capítulo 1

Introducción

Internet se ha convertido en una herramienta de uso diario para todo el mundo. La mayoría de nuestra vida la desarrollamos en la red: desde buscar cómo hacer la cena, a comentar el último capítulo de nuestra serie favorita. Pero, ¿somos verdaderamente conscientes de la cantidad de datos que generamos a diario? En 2021, se estima que hubo alrededor de 4,66 millardos de usuarios activos en Internet por todo el mundo, generando cada día más de un billón de megabytes de datos, lo cual supone que aproximadamente el 60 % de la población mundial se encuentra conectada a la red [7]. Tal y como se observa en la figura 1.1, se estima que para 2024 la cantidad de datos presente en Internet alcanzará los 149 zettabytes.

Gran parte de dicha información proviene de los denominados como *social media* ó medios sociales que, aunque en primera instancia pueden evocar únicamente a las redes sociales, suponen una parte aún mayor de la red. Los **medios sociales** son las plataformas sobre las que interactúan y socializan las personas, formando comunidades en las que compartir ideas, noticias o intereses. En contraposición con los medios tradicionales, donde el contenido es generado por un gran emisor único, los medios de comunicación social permiten que sea la misma comunidad que consume el contenido el responsable de generarla. La interacción es un factor clave en los medios sociales [29].

Éstas características resultan cruciales en diversas plataformas, por lo que los medios sociales se usan para una amplia variedad de propósitos:

- Mantener microblogs y estar al día de la actualidad (por ejemplo, vía Twitter)
- Crear y compartir contenido multimedia (por ejemplo, vía YouTube)
- Encontrar respuesta a preguntas específicas (por ejemplo, vía Stack Overflow)
- Leer reseñas y opiniones, tanto de usuarios como de crítica especializada, de películas, series o videojuegos (por ejemplo, vía Metacritic)
- Estar en contacto con familiares y amigos (por ejemplo, vía Facebook)
- Conocer las valoraciones de otros usuarios que han acudido al mismo restaurante al que planeas ir tú (por ejemplo, vía Google Maps).



Figura 1.1: Estadísticas de datos generados diariamente durante 2021.

1.1. Planteamiento del problema

En vista de la ingente cantidad de datos que están a nuestra disposición, cabe pregunparse qué conclusiones podemos extraer de ellos. Esta pregunta surge especialmente en el ámbito empresarial, pues estos datos pueden suponer una fuente de información en tiempo real sobre las tendencias del mercado y el comportamiento de los consumidores, permitiendo a los equipos directivos tomar decisiones que se ajusten a la realidad de una manera más precisa. Esta corriente de pensamiento queda avalada por los diferentes estudios, que muestran que ya el 97,2% de las empresas está invirtiendo en proyectos de

análisis de datos e inteligencia artificial, de las cuales el 24 % emplean una toma de decisiones impulsada por los datos. Dicho auge del enfoque *data-driven* se debe sobre todo al hecho de que las compañías que emplean métodos para la estructuración y el análisis de datos han visto aumentados sus beneficios una media de un 8 % [42].

Este interés queda también reflejado en el mercado laboral, donde se pueden ver decenas de miles de ofertas de puestos de trabajo que requieren de habilidades de análisis de datos, disponibles a lo largo de todo el mundo [43]. En concreto, en el informe del año pasado sobre competencias informáticas elaborado por la plataforma de selección y entrevistas a desarrolladores DevSkiller, se registró un aumento del 295 % en el número de tareas relacionadas con la ciencia de datos que los reclutadores laborales establecían para los candidatos en el proceso de entrevistas durante el año 2021. Este incremento de demanda ya fue predicho por autoridades como la Royal Society del Reino Unido, que en 2019 advertía que la demanda de científicos e ingenieros de datos se había triplicado en los últimos cinco años, dejando a las empresas "desesperadas por encontrar profesionales para desbloquear el potencial de las nuevas tecnologías", como el aprendizaje automático y la inteligencia artificial [32].

No obstante, el aumento de la demanda en dichos puestos no ha crecido acorde a las capacidades del mercado, si no más bien al contrario. Esto ha generado la conocida como **brecha de habilidades de datos**, pues los profesionales del sector no han tenido la oportunidad de formarse lo suficientemente rápido como para lograr adquirir las competencias demandadas. Esto tiene importantes consecuencias. Para 2030, la escasez de mano de obra en el sector de la tecnología, los medios y las telecomunicaciones podría hacer que Estados Unidos perdiere aproximadamente 162.000 millones de dólares, según un informe de 2018 de Korn Ferry. A nivel mundial, la brecha de habilidades digitales podría hacer que 14 países del G20 perdieran 11,5 billones de dólares de crecimiento acumulado del producto interior bruto, según estimaciones de Salesforce [25].

La finalidad última de este trabajo consiste en dar respuesta a ambos desafíos a la vez, generando una serie de cuadernos interactivos de Jupyter para la extracción, cribado y análisis de información relativa a una serie de compañías junto a la opinión de los usuarios con respecto a los lanzamientos más recientes de dichas empresas. De esta forma, quedará a disposición del estudiantado material didáctico que sirva como punto de partida a futuros trabajadores del campo del análisis de datos sociales, así como un estudio efectivo del cual las empresas implicadas puedan extraer conclusiones inmediatas del calado de sus productos en la comunidad.

1.2. Objetivos del trabajo

El proyecto busca resolver la problemática expuesta anteriormente, centrándose en datos sociales textuales relativos a un subconjunto de compañías y a sus productos lanzados más recientemente. En concreto, se propone la consecución de los siguientes objetivos:

- **OBJ-1:** Generar un conjunto de cuadernos interactivos que sirvan de introducción a la extracción de datos sociales y su posterior análisis.

- **OBJ-2:** Identificación de los principales temas asociados a la marca mediante *topic modelling*.
- **OBJ-3:** Detección de la opinión pública mostrada por la comunidad con respecto a los lanzamientos más recientes de cada empresa.
- **OBJ-4:** Correlación de estos datos con las ventas y acogida por parte del público.

1.2.1. Restricciones

No obstante, se debe tener también en cuenta las diferentes restricciones encontradas durante el desarrollo del proyecto:

- **REST-1:** El trabajo se ha realizado en un equipo que tiene Windows 10 como sistema operativos, usando la versión de Python 3.0 y pip 22.2.2, lo cual provoca que algunas implementaciones algo antiguas de librerías hayan causado problemas, obligando a recurrir a versiones más recientes (como es el caso del cambio de librería usada para implementar el modelo bitem).
- **REST-2:** El alcance y la duración del proyecto deben enmarcarse dentro de los estándares establecidos por la guía docente del Trabajo Fin de Grado (12 ETCS).
- **REST-3:** Durante el proceso de planificación y documentación del trabajo, se produjo la compra de Twitter por parte de Elon Musk. Una de sus primeras decisiones fue la de convertir la API de Twitter en un servicio de pago a partir del 9 de febrero de 2023, lo que precipitó el proceso de obtención de datos.[34]

1.3. Contexto empresarial

Cabe destacar que este Trabajo Fin de Grado brota a partir de la colaboración entre la Universidad de Valladolid y la empresa NielsenIQ, centrada en el ámbito del análisis de datos relativos al comportamiento y consumo de los usuarios.

NielsenIQ nace como empresa dentro del conglomerado de medios neerlandés-estadounidense con sede en Nueva York conocido como Nielsen Holdings. Actualmente se sitúa como líder mundial en medición de audiencias, análisis de datos referidos a productos y servicios, y proveedor de información de *marketing* [1]. Combinando el uso de complejas herramientas de análisis y soluciones de marketing integradas, Nielsen ofrece a sus clientes una visión global tanto de su mercado como de sus clientes. Con varias sedes distribuidas a lo largo del mundo, la empresa opera en más de 100 países con un equipo global dedicado a ayudar a sus clientes de manera efectiva, descubriendo oportunidades que previamente permanecían ocultas [2].

En concreto, NielsenIQ se enfoca en brindar la más completa visión del comportamiento de los consumidores de manera imparcial a nivel mundial, centrado en las compañías de bienes de consumo y minoristas líderes de todo el mundo. Para ello, hace uso de un

conjunto de datos integrales midiendo todas las transacciones por igual, brindando a los clientes una visión prospectiva del comportamiento de los consumidores. Su filosofía abierta sobre integración de datos permite que exista una amplia base sobre la que trabajar e investigar.[3]

A partir de marzo de 2021, NielsenIQ fue adquirida por el inversor de capital privado Advent International, cambiando su estatus al de empresa independiente. Este cambio de paradigma ha permitido a NielsenIQ acelerar su transformación y fortalecer su posición como líder del mercado del análisis [47].

1.4. Estructura de la memoria

El trabajo se articula siguiendo el siguiente esquema:

- **Capítulo 1.** Introducción. Descripción general del problema a tratar, así como un primer esbozo del entorno empresarial en el cuál se enmarca el proyecto. Delimitación de objetivos, identificación de restricciones y un esquema de la estructura del documento.
- **Capítulo 2.** Planificación. Metodología de trabajo a seguir, fijando las tareas a desarrollar enmarcadas dentro del contexto temporal previsto. Se presenta también un balance económico y temporal, ligado a los costes derivados de la consecución del proyecto, comprobando si la realidad se ha ajustado a la planificación prevista.
- **Capítulo 3.** Contexto del trabajo. Dónde se enmarca el proyecto dentro de la realidad actual, tanto en el entorno general como específico, presentando la terminología básica de la ciencia de datos enfocada al procesamiento del lenguaje natural. Asimismo, se exponen algunos trabajos ya existentes que realizan estudios similares al de este proyecto.
- **Capítulo 4.** *Data extraction.* Presentación de las dos técnicas más habituales para la recopilación de información escrita en la web: el uso de APIs y la técnica conocida como *web scraping*. Se muestra la implementación de las mismas, haciendo un análisis de las estructuras de datos obtenidas.
- **Capítulo 5.** Preprocesado de datos. En esta sección se muestran las técnicas más comunes para cribar y limpiar la información obtenida a través de la web, aplicando estos procesos a la propia información que se ha recolectado en el apartado previo y así tener un conjunto final de datos útil.
- **Capítulo 6.** *Topic modeling.* A lo largo de este apartado se realiza un análisis de los diferentes temas que se abordan en los textos obtenidos, primero mediante el análisis preliminar que ofrece la técnica del *wordcloud*, para después profundizar en los diferentes temas gracias a modelos de detección de tópicos: LDA y BTM.

- **Capítulo 7.** *Sentiment analysis.* En último lugar, se introduce una de las técnicas más potentes relativas al análisis de textos producidos en Internet: el análisis de sentimientos. Se aplicarán modelos preentrenados para extraer dichas conclusiones, presentando incluso un modelo propio para detectar el sentimiento general de reseñas.
- **Capítulo 8.** Conclusiones y trabajo futuro. A modo de colofón, se realiza una comparativa final entre las empresas y sus productos a través de la información recogida a lo largo del estudio. Se presentan también posibles mejoras que podrían realizarse en futuras iteraciones del trabajo.
- **Apéndice A.** Manual de instalación del entorno de ejecución que permite visualizar e interactuar con los cuadernos de Jupyter.
- **Apéndice B.** Contenido adjunto a la memoria.
- **Bibliografía.**

Capítulo 2

Planificación

2.1. Metodología de trabajo

La metodología elegida para el desarrollo del proyecto es Scrum [57]. **Scrum** es un marco de gestión de proyectos de metodología ágil que permite a equipos estructurar y gestionar el trabajo mediante un conjunto de valores, principios y prácticas. Scrum destaca por permitir abordar proyectos centrándose en el empirismo y el pensamiento *lean* -aquel que reduce el despilfarro y se centra en lo esencial-, siendo una heurística idónea para desarrollos como el propuesto en este trabajo, en el cual el equipo de desarrollo (en este caso el estudiante) carece de una experiencia plena, la cual se irá incrementando a lo largo del proyecto gracias al aprendizaje continuo y la adaptación a factores fluctuantes. Esta adaptabilidad de Scrum a las necesidades cambiantes resulta también ideal para realizar la planificación del proyecto [21].

Scrum se cimenta en la división del equipo de desarrollo en diferentes **roles**. No obstante, dado que nos encontramos ante un desarrollo individual, será necesario hacer una adaptación de éstos a nuestra situación:

- **Desarrollador.** Rol encarnado por el estudiante que realiza el proyecto. Su misión es desarrollar el Trabajo de Fin de Grado, tanto la implementación del mismo como la memoria que se entregará como resultado final.
- **Product owner.** Encargado de maximizar el valor del producto resultante generado tras cada iteración, así como de dar una primera idea del *Product Backlog*. Este papel lo desempeña la empresa que encarga al estudiante la realización del TFG, representada por la tutora de prácticas de la empresa.
- **Scrum Master.** Encargado de hacer que se sigan las pautas del marco de trabajo Scrum. Este papel está compartido tanto por los tutores de la universidad y la empresa, como por parte del propio estudiante, último responsable de la adecuación de sus esfuerzos a la metodología seleccionada.

A parte de los roles, Scrum toma como base para la estructuración de las tareas los **eventos**, que dotan al desarrollo de transparencia para así realizar labores de inspección y mejora de los mismos a lo largo del desarrollo. Distinguiremos los siguientes eventos para nuestro proyecto:

- **Sprint.** Cada *sprint* puede considerarse como un proyecto a pequeña escala. Durante el tiempo de duración de un *sprint*, pueden realizarse cambios en el *Product Backlog* o incluso redefinir el alcance, siempre y cuando se mantenga el compromiso con la calidad y dichas modificaciones no pongan en peligro el objetivo último del *sprint*.
Para este proyecto en concreto, se ha estimado que serán necesarios 5 sprints de 2 semanas cada uno [50], con una carga de 60 horas por *sprint*.
- **Planificación de sprint.** Primer evento realizado antes de comenzar un *sprint*. Reunión entre el estudiante y uno de los tutores, en la que se fijan los objetivos a alcanzar en el *sprint*, las Historias de usuario del *Product Backlog* a desarrollar y el plan para llevarlo todo a cabo. El compendio de todo se denomina *Sprint Backlog*. La primera sesión de planificación se realizó de manera individual, pero a partir del segundo sprint se unificó con el evento de Reunión tutorizada para agilizar los procesos de planificación, a causa de la breve duración de los *sprints*.
- **Daily (Scrum).** Reunión diaria enfocada a analizar los avances hechos en el desarrollo y, en caso de que fuese necesario, adaptar el *Sprint Backlog* al plan de trabajo marcado. Al tratarse de un trabajo llevado a cabo por una sola persona, dicha reunión consiste en la revisión de los objetivos marcados y modificarlos en caso de que procediese.
- **Reunión tutorizada.** Evento destinado a evaluar los progresos conseguidos durante el sprint, fijando posibles puntos de mejora para la siguiente iteración del producto entregado. Estas mejoras se analizan también en dichas reuniones para optimizar tiempo, pues se realizan entre el estudiante y la tutora de la empresa o el tutor de la universidad. Este evento vendría a unificar los conceptos de Scrum conocidos como *Sprint Review* y *Sprint Retrospective*.

En último lugar encontramos los **artefactos**, los cuales representan trabajo o valor. Su diseño está enfocado a la transparencia, para así poder alcanzar el estándar de progreso que cada uno de ellos modela:

- **Product Backlog.** Lista ordenada de lo que debe mejorarse del producto final. Sus componentes pueden refinarse e incluso descomponerse en otros más precisos y tangibles, momento en el cual se añaden al *Sprint Backlog* de la planificación. Su fin último es asegurar que se lleva a cabo el **objetivo de producto**, meta a largo plazo que define el estado futuro ideal del proyecto y que debe cumplirse antes de poder asignar uno nuevo.

- **Sprint Backlog.** Planificación hecha por y para desarrolladores. Se compone del objetivo del sprint, una lista de los elementos del *Product Backlog* específicos del sprint en cuestión, así como el plan para llevar a cabo el incremento (porqué, qué y cómo; respectivamente). Permite al desarrollador tener claro cuál es el **objetivo del sprint**, elemento contra el cual se compromete el *Sprint Backlog*. Es un objetivo concreto que se define durante la planificación.
- **Incremento.** Paso específico encaminado a la consecución del objetivo de producto. Cada incremento se suma a los ya realizados previamente, aportando también valor al ser un elemento utilizable. Un incremento culmina cuando se alcanza el denominado *Definition of Done* (DoD) ó Definición de hecho, estado en el cual se han alcanzado los estándares de calidad fijados.

Para el desarrollo de este proyecto, fue común la entrega de incrementos a la empresa que proporcionó el Trabajo de Fin de Grado, pues supone uno de los *stakeholders* principales del trabajo.

2.1.1. Metodología de trabajo en *data science*

La metodología Scrum resulta válida para el desarrollo de proyectos independientemente de su naturaleza. Sin embargo, dado que este trabajo se centra específicamente en el desarrollo de un proyecto de *data science*, será conveniente estudiar las fases prototípicas de las que se conforman proyectos de esta índole [17].

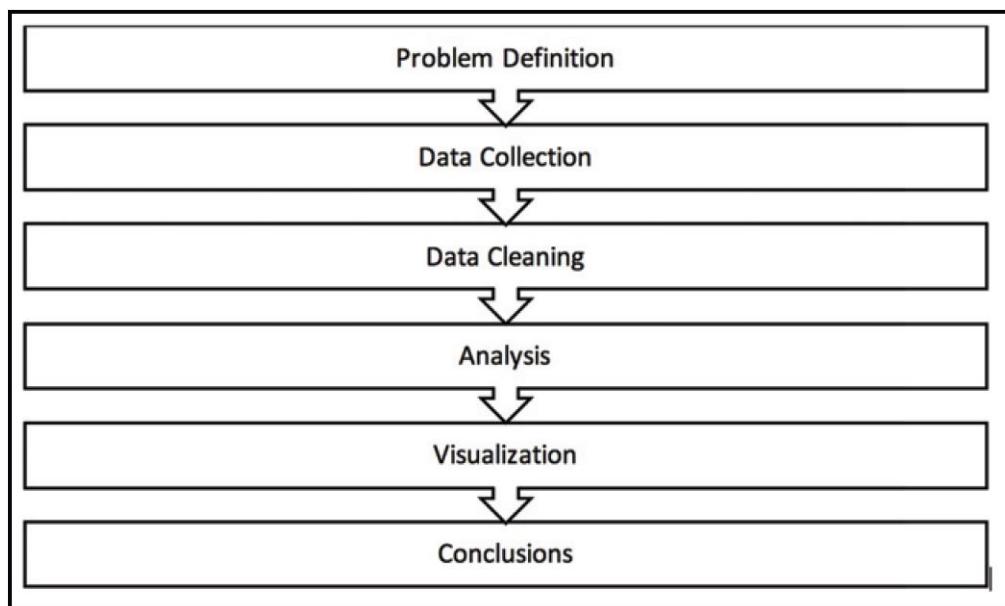


Figura 2.1: Pasos para la extracción, procesado y análisis de datos

1. **Definición del problema:** Un entendimiento adecuado del problema nos permitirá elegir las fuentes adecuadas de información, así como los métodos adecuados para analizar dichos datos y las conclusiones que esperamos extraer. Esto resulta crucial, pues si no se define un objetivo claro a la hora de realizar un estudio, podemos acabar con datos contaminados (por ejemplo, anuncios generados por bots) que no proporcionen ningún tipo de conclusión relevante.
2. **Recolección de datos:** Una vez elegido el ámbito de estudio en el que se centrará el proyecto, será necesario adquirir los datos que se vayan a analizar. En función de la fuente de la que se recolecten los datos, esta labor se puede llevar a cabo mediante el uso de una API que facilite su extracción automatizada (como es el caso de Twitter y su API); o bien mediante un procedimiento algo más complejo conocido como *scraping*, destinado a obtener información de medios que no proporcionan dichas facilidades (como es el caso de blogs o foros).
3. **Limpieza de datos:** Tras almacenar todos los datos que podrían ser considerados como relevantes, será necesario realizar una criba para deshacernos de todos aquellos que no nos resulten de interés. Algunos ejemplos de este caso son el uso de un filtro de idiomas o descartar datos duplicados. El objetivo último de esta fase consiste en obtener un conjunto de datos limpios con el que poder empezar a trabajar.
4. **Análisis:** En el proceso de análisis, tendremos que definir qué tipo de estudio realizaremos, así como cuál es la mejor estructura para nuestros datos. La elección del método de análisis depende enteramente del objetivo final del estudio, pues puede variar desde el uso de técnicas estadísticas básicas hasta la necesidad de emplear modelos de *machine learning*.
5. **Visualización de los resultados** La mejor manera de entender los resultados obtenidos durante la fase de análisis siempre suele ser a través de representaciones gráficas, las cuales resultan aclaratorias tanto para el analista (que puede descubrir nuevas características en los datos que motiven un segundo análisis de la información desde otra perspectiva), como para el cliente que haya encargado el estudio.
6. **Conclusiones** A tenor de la información obtenida durante el análisis y la visualización de los mismos, llega el momento de extraer conclusiones para finalizar el estudio y hacer una presentación final de toda la información obtenida.

A lo largo del proceso de desarrollo, así como en este mismo documento, se ha intentado estructurar la planificación, diseño y ejecución del proyecto ateniéndose al orden establecido por este esquema.

2.2. Planificación temporal

El evento de Planificación de *sprint* requiere de una fase de estimación, en la cual se calcule el tiempo aproximado que requerirá la consecución de las diversas tareas que componen el *sprint*. Para realizar dicha asignación existen una gran variedad de métodos, aunque el que se ha utilizado en este desarrollo es el conocido como *Planning Poker*.

El **Planning Poker** [4] es una técnica de estimación para proyectos ágiles, orientada a la búsqueda de consenso entre las diferentes partes involucradas en el desarrollo del mismo para valorar el esfuerzo necesario de llevar a cabo la consecución de una historia de usuario. Los miembros del equipo asignarán a cada historia un valor de la sucesión de Fibonacci (1, 2, 3, 5, 8...), eligiéndose como valor final aquel más votado por el equipo. Cabe destacar que dicha puntuación obtenida nunca podrá ser convertida en unidades de tiempo, pues lo que se representa con dicho valor es:

- Cantidad de trabajo a realizar
- Complejidad de Historia o tarea
- Riesgos o incertidumbres que pueden presentarse a lo largo del desarrollo

Para este proyecto, a pesar de tratarse de un desarrollo individual, el empleo de esta metodología para la definición de puntos de historia ha permitido que la planificación y estructuración de las tareas se realice de manera sistemática, definiendo de mejor manera el esfuerzo requerido para cada *sprint* a la hora de ubicarlo en el tiempo. Al trabajar con *sprints* cortos, cada uno está compuesto por una única historia de usuario, descompuesta en diversas tareas. En total, se han obtenido un total de 5 historias de usuario que engloban un conjunto de 85 puntos de historia, teniendo cada uno aproximadamente 17 puntos de historia asignados, lo cual garantizaría una distribución equilibrada del trabajo entre los diferentes *sprints*.

Además, la planificación dinámica de los *sprints* ha permitido un desarrollo con mayor grado de libertad, ajustándose tanto a las necesidades del proyecto (como fue la recogida de datos por parte de Twitter debido a las restricciones de pago futuras de los servidores), como a las de los diferentes actores involucrados en el proceso de desarrollo (como es el conflicto con otras responsabilidades del estudiante -exámenes de asignaturas del grado así como otro TFG y prácticas de empresa- para poder seguir con la implementación del trabajo).

2.2.1. Sprint #1

Durante el primer *sprint* se llevó a cabo la ejecución de las tareas que componen la primera historia de usuario [2.1], centrada en la adquisición de nociones básicas dentro del contexto del trabajo y buscando delimitar el enfoque original del proyecto, pues tras un primer análisis se dirimió que un enfoque de objetivos más específico permitiría desarrollar un producto de mayor calidad en el tiempo dado.

Capítulo 2. Planificación

Identificador de la tarea	Nombre de la tarea	Puntos de historia	07/11/2022	08/11/2022	09/11/2022	10/11/2022	11/11/2022	12/11/2022	13/11/2022	14/11/2022	15/11/2022	16/11/2022	17/11/2022	18/11/2022	19/11/2022	20/11/2022
H1	Documentación inicial	12														
T1.1	Introducción al análisis de datos sociales	2														
T1.2	Introducción al Procesamiento del Lenguaje Natural	3														
T1.3	Introducción a la teoría de grafos	2														
T1.4	Introducción a topic modelling	3														
T1.5	Introducción al análisis de sentimientos	2														
R1	Delimitación de scope															

Cuadro 2.1: Primera historia de usuario

2.2.2. Sprint #2

Identificador de la tarea	Nombre de la tarea	Puntos de historia	02/02/2023	03/02/2023	04/02/2023	05/02/2023	06/02/2023	07/02/2023	08/02/2023	09/02/2023	10/02/2023	11/02/2023	12/02/2023	13/02/2023	14/02/2023	15/02/2023
H2	Data extraction	15														
T2.1	Introducción a la API de Twitter	3														
T2.2	Extracción de datos de la API de Twitter	5														
R2	Evaluación y próximos pasos															
T2.3	Introducción al web scraping	2														
T2.4	Extracción de datos de Metacritic	3														
T2.5	Análisis y estructuración de los datos	2														

Cuadro 2.2: Segunda historia de usuario

En un principio, la segunda historia de usuario [2.2] se iba a componer únicamente de tareas relacionadas con la extracción de datos provenientes de Twitter, las cuales se realizarían durante un sprint que tendría como punto de partida el día 27 de marzo, para así no coincidir con otras actividades que dificultasen la tarea por parte del estudiante. No obstante, la transición de la API a un modelo orientado al pago por suscripciones [34] obligó a realizar esta fase de manera algo más precipitada, por lo que para completar los datos obtenidos mediante esta vía, se decidió también recopilar datos mediante técnicas de *web scraping*.

2.2.3. Sprint #3

El tercer *sprint*, en el cual se lleva a cabo la tercera historia de usuario [2.3], está enfocado en el cribado de la información obtenida en la etapa previa, buscando dotar a los datos de la forma propicia para realizar tanto un análisis preliminar de los mismos, como un escrutinio más concienzudo en base a los temas que se tratan en ellos.

Identificador de la tarea	Nombre de la tarea	Puntos de historia	10/04/2023	11/04/2023	12/04/2023	13/04/2023	14/04/2023	15/04/2023	16/04/2023	17/04/2023	18/04/2023	19/04/2023	20/04/2023	21/04/2023	22/04/2023	23/04/2023
H3	Data cleaning y topic modelling	18														
T3.1	Data cleaning	3														
T3.2	Análisis exploratorio de los datos	1														
T3.3	Topic modelling usando LDA	5														
R3	Análisis de resultados															
T3.4	Stemming y lemmatization	2														
T3.5	Pulido del modelo LDA	2														
T3.6	Topic modelling usando biterm	5														

Cuadro 2.3: Tercera historia de usuario

2.2.4. Sprint #4

La cuarta historia de usuario [2.4], desarrollada a lo largo del cuarto *sprint*, está enfocada al pulido del trabajo previo, así como a poner en práctica técnicas de análisis de sentimientos para la obtención de conclusiones definitivas sobre los productos evaluados y las empresas que los ofertan. El desfase de comienzo de este *sprint* con respecto al anterior se debe a que el *sprint* previo tuvo que demorarse al coincidir con la realización de exámenes parciales por parte del estudiante, de lo cual se hablará más en profundidad en el balance temporal realizado al final del capítulo.

Identificador de la tarea	Nombre de la tarea	Puntos de historia	04/05/2023	05/05/2023	06/05/2023	07/05/2023	08/05/2023	09/05/2023	10/05/2023	11/05/2023	12/05/2023	13/05/2023	14/05/2023	15/05/2023	16/05/2023	17/05/2023
H4	Sentiment analysis	17														
T4.1	Análisis de tweets mediante modelos preentrenados	3														
T4.2	Análisis de reseñas mediante modelos preentrenados	5														
T4.3	Análisis de reseñas mediante modelos propios	5														
T4.4	Reestructuración de los cuadernos de Jupyter	2														
T4.5	Conclusiones	2														
R4	Presentación de la implementación final al tutor universitario															
R5	Presentación de la implementación final a la empresa															

Cuadro 2.4: Cuarta historia de usuario

2.2.5. Sprint #5

En último lugar, en el quinto *sprint* se lleva a cabo la ejecución de la quinta historia de usuario [2.5], en la que se redacta la memoria final del trabajo para así poder presentarla al cliente final. A pesar de que se obvian dentro de la planificación, se mantiene una comunicación constante con el tutor para que aporte retroalimentación y así puedan irse corrigiendo los capítulos redactados del documento.

2.3. Presupuestos

Identificador de la tarea	Nombre de la tarea	Puntos de historia	17/05/2023	18/05/2023	19/05/2023	20/05/2023	21/05/2023	22/05/2023	23/05/2023	24/05/2023	25/05/2023	26/05/2023	27/05/2023	28/05/2023	29/05/2023	30/05/2023	31/05/2023	01/06/2023
H5	Redacción de la memoria	23																
T5.1	Introducción	2																
T5.2	Planificación	3																
T5.3	Contexto del trabajo	5																
T5.4	Data extraction	2																
T5.5	Data cleaning	1																
T5.6	Topic modeling	5																
T5.7	Sentiment analysis	3																
T5.8	Conclusiones	1																
T5.9	Apéndices	1																
R6	Presentación de la memoria final al cliente																	

Cuadro 2.5: Quinta historia de usuario

2.3. Presupuestos

Para calcular el coste de realización del proyecto debemos distinguir principalmente dos apartados: los recursos humanos (el coste de contratar y pagar a los trabajadores encargados de desarrollar el trabajo) y los recursos técnicos (los cuales comprenden el conjunto de herramientas hardware y software utilizadas para la consecución final del Trabajo Fin de Grado).

2.3.1. Aproximación a la estimación de costes

Si bien, tal y como se ha mencionado anteriormente, no se puede realizar una conversión directa entre el número de puntos de historia y las horas de trabajo correspondientes a la consecución de la misma, sí que es posible realizar una aproximación a ello. No obstante, dicho acercamiento debe realizarse teniendo siempre en cuenta los preceptos de la metodología ágil aplicado a la estimación de costes:

- Los costes deben resultar claros y transparentes para todos los integrantes del equipo
- Los costes deben ser calculados de manera automática y frecuente, por lo que resultaría conveniente disponer de una fórmula que facilitase dicho cálculo
- El cálculo de costes ha de ser comprensible y replicable

En base a estos objetivos, es posible definir una fórmula que permita estimar el coste de cada sprint, tal y como se muestra en [58], a partir de los siguientes elementos:

- *Horas* → Horas totales dedicadas al *sprint* por el equipo
- *Salario* → Salario medio de los trabajadores en € por hora
- *Velocidad* → Velocidad media del equipo de desarrollo
- *PH* → Puntos de historia totales del *sprint*

Obtenemos entonces la siguiente fórmula para calcular el coste estimado de cada *sprint*:

$$\text{Coste} = \frac{\text{Horas} \cdot \text{Salario}}{\text{Velocidad}} \cdot \text{PH}$$

Esta fórmula es aplicable a equipos de desarrollo conformados por varias personas, pero es posible adaptarla al marco de este proyecto unipersonal si consideramos los diferentes roles que tendrá que adoptar el estudiante a lo largo del desarrollo. En concreto, estos roles serán dos:

- **Científico de datos.** Encargado de la formulación de las preguntas y la extracción de datos que vayan a darles solución [39]. Por lo tanto, estará involucrado tanto en la fase de desarrollo como en la de recogida de información. En España, su sueldo medio es de 35000€ anuales [61], que se convierten en 2917€ al mes prorrateando en 12 meses (en vez de dividirlo en 14 pagas anuales) y supone un salario de 18.23€ la hora (considerando una semana laboral de 8 horas al día y 5 días por semana; 40 horas a la semana y 160 horas al mes).
- **Analista de datos.** Analiza los datos obtenidos para obtener respuestas a las preguntas formuladas por el científico de datos, dotando a la información de un formato adecuado para la extracción de conclusiones de negocio. En nuestro caso, se centrará en la presentación de resultados así como la redacción de la memoria final. Su salario medio en España ronda los 29000€ anuales [60], lo que supone 2417€ al mes o 15.10€ la hora.

Así, para el cálculo de horas trabajadas tomaremos la duración de cada *sprint* (2 semanas trabajando 5 horas al día; 70 horas) y únicamente un miembro en el equipo de desarrollo, cuyo salario será la media de ambos roles asumidos (16.67€ la hora) para simplificar los cálculos. Además, como velocidad media del equipo de desarrollo escogeremos la media de los puntos de historia de todas las definidas (17 puntos de historia por sprint). Esto nos permitirá estimar el coste de cada sprint, así como del desarrollo total del proyecto. Para este último debemos considerar también el gasto derivado de inscribir al empleado en la Seguridad Social, lo cual supone un 30 % de su salario bruto [78]. Podemos observar los resultados finales en [2.6].

Sprint	Horas	Salario (€/hora)	Velocidad (PH/sprint)	PH	Coste estimado
Sprint 1	70	16,67 €	17	12	823,69 €
Sprint 2	70	16,67 €	17	15	1.029,62 €
Sprint 3	70	16,67 €	17	18	1.235,54 €
Sprint 4	70	16,67 €	17	17	1.166,90 €
Sprint 5	70	16,67 €	17	23	1.578,75 €
Coste total					5.834,50 €
Coste total + Seguridad Social					7.584,85 €

Cuadro 2.6: Aproximación a la estimación de los costes por sprint usando puntos de historia

Debe tenerse en cuenta que estos valores son una mera estimación de costes previa al desarrollo del proyecto. Para poder ver los gastos finales reales, basta referirse a la sección 2.4, en la que se calculan los costes efectivos una vez concluido el desarrollo del trabajo.

2.3.2. Recursos técnicos

El trabajo ha sido desarrollado en un ordenador portátil Acer Aspire 3, con procesador AMD Ryzen 5, 16GB de RAM y disco duro SSD de 1TB. Este dispositivo de gama media se estima que tiene unos 4 años de vida útil y costó 550€, por lo que debemos calcular su amortización a lo largo de la duración del proyecto (unos 3 meses, al ser 5 sprints de 2 semanas cada uno). Aparte de este dispositivo, también se ha usado un disco duro externo de 1TB para guardar una copia de respaldo tanto de los datos descargados como del propio trabajo en sí. Dicho componente tuvo un coste de 50€ y se estima que su vida útil ronda los 8 años de media.

También se ha precisado de conexión a Internet estable para poder consultar fuentes de información, realizar la descarga de los datos a analizar o mantener reuniones con los tutores del proyecto; por lo que es necesario incluir su coste dentro del presupuesto. La tarifa elegida proporciona 500MB de fibra a un precio de 30,95€ al mes.

Con respecto a las herramientas software, tanto Jupyter Notebook como Overleaf ofrecen licencias gratuitas para trabajar con ellos. Además, Microsoft Office y el sistema operativo Windows 10 se incluyen dentro de la compra del ordenador portátil, por lo que el coste reflejado de todo se contará como 0€ en [2.7].

Como último apunte, a pesar de que el proyecto se pudo desarrollar antes del cambio de modelo de negocio de la API de Twitter, se ha considerado cuál habría sido el coste en caso de haber realizado el desarrollo una vez se produjo el cambio. Al haber descargado casi un millón de tweets para poder analizarlos, habría hecho falta usar un mes de suscripción Pro a la API, la cual permite obtener hasta un millón de tweets cada mes por una suscripción mensual de 5000\$ [71] .

Recurso	Coste	Porcentaje de uso	Total
Ordenador portátil	550,00 €	6,25%	34,38 €
Disco externo	50,00 €	3,13%	1,56 €
Conexión a Internet	30,95 €	300,00%	92,85 €
Twitter API Pro	4.661,95 €	100,00%	4.661,95 €
Jupyter Notebook	0,00 €		0,00 €
Overleaf	0,00 €		0,00 €
Microsoft Office	0,00 €		0,00 €
Windows 10	0,00 €		0,00 €
Coste total			4.790,74 €

Cuadro 2.7: Recursos técnicos

2.3.3. Coste final

Una vez analizados ambos costes, se estima que el proyecto tendría un coste de desarrollo de 7713,64€, que se podría llegar a elevar hasta los 12375,59€ en caso de tener que haber utilizado la API de pago de Twitter.

2.4. Balance temporal y económico

La realización del primer *sprint* se culminó antes de lo previsto, fundamentalmente por el hecho de que la tarea de introducción a teoría de grafos se omitió, ya que la inclusión de estas consideraciones habría alargado en demasiada la duración del desarrollo y habría supuesto un alcance demasiado ambicioso del proyecto con respecto a la carga lectiva que tiene un Trabajo Fin de Grado. El sprint finalizó con la reunión del 17 de noviembre, tras 40 horas de trabajo.

Si bien las tareas relacionadas con la extracción de datos de Twitter concluyeron en el tiempo estipulado, motivado principalmente por la restricción temporal subyacente al cese de acceso gratuito de dicho recurso una vez pasada la fecha límite, tras la reunión de evaluación se pospuso la realización de las tareas relacionadas con el web scraping hasta mediados de marzo. Esto se debió a que dicha etapa coincidió con la finalización de las prácticas de empresa del estudiante y el inicio del nuevo cuatrimestre universitario. Dichas labores fueron llevadas a cabo del 17 de marzo al 24 de ese mismo mes. El total de horas dedicada a este *sprint* fueron 70.

De nuevo, las tareas previas a la reunión del tercer *sprint* se llevaron a cabo en el tiempo previsto, aunque tanto la reunión como las labores siguientes se postergaron hasta el 3 de mayo, al coincidir con el período de exámenes parciales de la universidad para el estudiante. El *sprint* concluyó el 8 de mayo, después de una dedicación de 75 horas de trabajo.

2.4. Balance temporal y económico

Inmediatamente después de culminar el *sprint* previo, dio comienzo el cuarto *sprint* buscando equilibrar el desajuste temporal sufrido y así poder ceñirse a los plazos originalmente estipulados. Dicho esfuerzo tuvo sus frutos y logró finalizarse el *sprint* en el plazo previsto, tras 65 horas de trabajo.

La mayor problemática surgió con respecto al último *sprint*, pues no fue posible ceñirse al objetivo marcado, al comenzar los exámenes finales de la universidad y tener que entregar las respectivas prácticas finales de dichas materias. Esta casuística, unido a otros motivos de índole personal, obligó a que la redacción de la memoria se hiciese a intervalos irregulares culminándose el 30 de junio, contabilizando un total de 90 horas trabajadas.

[2.8] muestra el diagrama de Gantt final, con las fechas definitivas de desarrollo de cada una de las tareas que componen las diversas historias de usuario ya mencionadas.

Cuadro 2.8: Diagrama de Gantt final con las fechas en que se desarrolló el proyecto

Tras el balance temporal, podemos **ajustar la presupuestación económica realizada usando como referencia el total de horas trabajadas por parte del estudiante**, en vez de utilizar la estimación dada previamente. El total de horas de trabajo fueron 340, lo cual se adecúa a la carga lectiva del Trabajo de Fin de Grado (entre 300 y 360 horas al ser 12 ECTS). Considerando que el salario del trabajador son los ya mencionados 16,67€ la hora, esto supone un total de 5667,80€ a lo que hay que sumar el coste de la Seguridad Social: un total de 7368,14€, muy similar a la estimación dada, lo cual muestra la robustez y precisión del método de estimación elegido. En conclusión, el presupuesto final de realización del trabajo han sido 7496,93€, pudiendo haber alcanzado los 12158,88€ en caso de haber necesitado usar la API de pago de Twitter.

Capítulo 3

Contexto del trabajo

3.1. Entorno específico

El **entorno** de una empresa es el conjunto de factores externos a una organización que, aún teniendo una influencia significativa en su estrategia, la organización no puede controlar. Dentro de éste distinguimos el **entorno general**, el cual es común a cualquier organización y viene determinado tanto por la sociedad como por la naturaleza y características del sistema socioeconómico; del **entorno específico**, el cual es propio de la tarea o actividad característica de la organización [53].

Tal y como se comentó en la introducción, este trabajo es fruto de la colaboración con la empresa NielsenIQ, por lo que resultará interesante analizar el entorno específico de la empresa para ser conscientes de las debilidades y fortalezas del estudio encargado con respecto al resto del mercado.

3.1.1. Entorno específico de la empresa

NielsenIQ es una empresa enmarcada en el sector de la investigación de mercado y el análisis de datos, ofreciendo a las empresas la oportunidad de analizar en detalle los comportamientos de los consumidores y puedan adecuar sus estrategias de mercado en base a dichas tendencias.

Para poder entender cuál es la situación de NielsenIQ dentro de la industria del análisis de datos y las investigaciones de mercado, aplicaremos el **modelo de fuerzas competitivas de Porter** [3.1] para así comprender su posición de mejor manera:

1. **Competencia:** NielsenIQ opera en un mercado altamente competitivo y en constante evolución, con diversas empresas que ofrecen servicios similares. Entre los competidores directos y potenciales contra los que NielsenIQ debe enfrentarse en términos de tecnología, metodologías de investigación y cobertura geográfica se encuentran [48]:

- **Syndigo.** Empresa que ofrece a sus clientes una plataforma integrada para la recolección, manejo y análisis de sus propios datos, No obstante, NielsenIQ

tiene ventaja competitiva con respecto a ellos gracias a un mejor servicio de atención a sus clientes, así como el reconocimiento de entidades autorizadas de la industria como la FDA (*Food and Drug Administration*) ó la AHA (*American Hospital Association*).

- **The Data Council.** Empresa que ofrece a minoristas, fundamentalmente aquellos que venden bienes de consumo empaquetados, información clara, precisa, independiente y completa sobre los productos presentes en las cadenas de suministros. Sin embargo, de nuevo NielsenIQ muestra su superioridad en el mercado al ofrecer a sus clientes un servicio de mayor calidad, con conjuntos de datos más consistentes, así como mayores alternativas de escalabilidad en función de las necesidades del cliente.
- **Spins.** Proveedor de información, análisis y consultoría sobre el consumidor minorista para las industrias de productos naturales, orgánicos y especializados. Al igual que con los otros competidores ya mencionados, la experiencia de usuario ofrecida por NielsenIQ y las mejores alternativas de escalabilidad colocan a NielsenIQ en una posición de ventaja.

A pesar de que NielsenIQ ostenta una posición privilegiada en el sector del análisis de datos, ofrecido especialmente a comercios minoristas, la aparición de nuevas tecnologías y la constante evolución de los medios de análisis obliga a la empresa a mejorar sin cesar para mantener sus estándares de calidad y no perder su posición de privilegio en el sector.

2. **Proveedores:** La materia prima con la que trabaja NielsenIQ son los datos, por lo que es necesario disponer de una amplia gama de proveedores para garantizar la calidad y disponibilidad de los mismos. Algunos de estos proveedores son:

- **Proveedores de datos minoristas.** Información sobre las ventas de productos en diferentes categorías y ubicaciones geográficas, obtenida a partir de acuerdos y asociaciones estratégicas.
- **Proveedores de datos de audiencia.** Información sobre la audiencia y el consumo de medios en diferentes canales como televisión, radio o medios digitales. A partir de estos datos de visualización e interacción, NielsenIQ puede realizar las mediciones y análisis pertinentes.
- **Proveedores de datos demográficos.** Información sobre las características demográficas, socioeconómicas y geográficas de la población, que permite a la empresa entender mejor el comportamiento de los consumidores.

Además, la empresa también necesita **proveedores de tecnología y software** para poder desarrollar y mantener sus plataformas de análisis y visualización de datos; así como obtener herramientas que le permitan recopilar, analizar y presentar los datos de manera eficiente y efectiva.

3. **Clientes:** NielsenIQ trabaja con una amplia variedad de clientes, destacando el servicio ofrecido a empresas de bienes de consumo y minoristas, para que éstos puedan analizar el desempeño de sus productos y comprender el comportamiento de los consumidores, permitiéndoles adecuar sus decisiones estratégicas a la realidad del contexto actual. Además, NielsenIQ también da servicio a empresas de medios y publicidad, para que puedan evaluar el alcance y efectividad de sus campañas; así como a otras agencias de investigación y consultoría que brindan servicio a sus propios clientes, dándoles la oportunidad de respaldar sus propias investigaciones, análisis y estrategias comerciales.
4. **Sustitutos:** Además de las alternativas dentro del propio sector ya mencionadas, los clientes que solicitan los servicios de NielsenIQ pueden optar por soluciones de datos internas a través de la creación de un departamento específico de análisis dentro de la propia empresa. No obstante, esto requeriría una alta inversión tanto en infraestructuras como en personal, aunque ni siquiera dicha inversión garantizaría soluciones de la calidad que ofrecen empresas centradas únicamente en esta clase de investigaciones.



Figura 3.1: Modelo de Porter

El sector del análisis de datos puede enmarcarse ya como un **sector estratégico consolidado**, pues supone un factor clave a la hora de planear la estrategia a futuro de cualquier empresa. También es un **sector fragmentado**, pues la industria de la investigación de mercado es altamente competitiva; así como **emergente**, a causa de los rápidos avances tecnológicos que experimenta el sector (como el análisis de *big data* o la inteligencia artificial).

3.1.2. Entorno específico del estudio

Como ya se explicó en la metodología de trabajo propia de *data science*, el primer paso consiste en definir de manera clara y precisa el tema del estudio a realizar. Para este trabajo, se estudiará la presencia e impacto en redes sociales de las tres principales compañías de videojuegos en la actualidad, las cuales ofrecen tanto hardware como software: Nintendo, PlayStation y Xbox. En concreto, el tema elegido facilita la comparativa a realizar pues, cuando se recabaron los datos de la red social Twitter (primera semana de febrero), las tres compañías acababan de sacar tres nuevos juegos desarrollados por estudios propios de las compañías: Fire Emblem Engage, Forspoken y Hi-Fi Rush; respectivamente. Se podrá entonces comparar la recepción de los juegos por parte del público generalista y la prensa especializada, recabando información de páginas de reseñas e impresiones en redes sociales.

Antes de realizar el estudio, parece oportuno dar una panorámica de la situación actual de la industria del videojuego, analizando la posición de las tres compañías en el mercado. Si bien en sus inicios el sector se encontraba bastante fragmentado, con varias compañías pugnando por hacerse con el favor del público y eligiesen su consola por encima de las de la competencia, hoy en día grandes nombres como Sega o Atari se han visto relegadas a adoptar un papel más secundario y centrarse únicamente en el desarrollo de software. Dejando a un lado el innegable auge del PC como plataforma de juego o consolas derivadas de dicho modelo como es la propia Steam Deck, cuando un consumidor se plantea adquirir una nueva consola de videojuegos, tiene tres alternativas principales en la actual generación:

- **Nintendo Switch:** Es la séptima consola de sobremesa que desarrolla la compañía japonesa. Tras la gran decepción que supuso la Wii U, su anterior consola, Switch aboga por un modelo híbrido que permite a los jugadores combinar el juego clásico de sobremesa con las prestaciones de una consola portátil. A pesar de las limitaciones de rendimiento que conlleva, la consola ha tenido un amplio calado, erigiéndose como la favorita del público y destinándose especialmente para el juego con familia y amigos. Desde su lanzamiento en marzo de 2017, se ha convertido ya en la tercera consola más vendida de la historia, superando a otros éxitos de la propia compañía como Game Boy o Wii [41].
- **PlayStation 5:** Tal y como su nombre indica, es el quinto modelo dentro de la familia PlayStation, marca propiedad de la empresa Sony Interactive Entertainment. La compañía japonesa lanzó al mercado la consola en noviembre de 2020, pero la pandemia mundial y la escasez de componentes para la fabricación de consolas ha limitado el stock disponible, obligando a que gran parte de los lanzamientos sean intergeneracionales o salgan también en PC. No obstante, esta transición generacional motivó un cambio de modelo en la empresa, adaptando sus planes de suscripción mensual para que estuviesen específicamente diseñados para usuarios de PlayStation 5. En especial, destacan los juegos ofrecidos mediante el servicio PlayStation Plus, los cuales rotan a principios de mes.

- **Xbox Series X:** Al igual que la consola de Sony, Series X fue lanzada en noviembre de 2020, aunque no adoleció de tantos problemas de stock como su rival directo. No obstante, Microsoft es más partidario de un modelo de negocio combinado, lanzando simultáneamente todos sus juegos tanto en consola como en PC. Esto se debe principalmente a su servicio de suscripción de Game Pass, que en 2022 sobrepasó los 25 millones de usuarios [24]. Xbox Games Pass ofrece a sus suscriptores un amplio catálogo de juegos, que cada mes se amplía con nuevos lanzamientos, dando la posibilidad de jugar a los títulos tanto en consola como en ordenador.

La competencia más directa reside entre PlayStation y Xbox, pues ambas compañías han realizado la transición generacional al mismo tiempo e intentan hacerse con el control del mercado. Por su parte, Nintendo con su modelo híbrido y experiencias más familiares, ya tiene su propio nicho de mercado y puede establecer cierta convivencia con las otras dos compañías. A finales de enero, las tres empresas sacaron juegos desarrollados por estudios internos y enfocados a experiencias de usuario individuales, por lo que la comparativa entre ellos parece procedente. Como último apunte, daremos también algo de contexto sobre el entorno y las circunstancias en las que se han desarrollado los juegos:

- **Fire Emblem Engage:** Decimoséptima entrega de la aclamada saga de rol táctico Fire Emblem, desarrollada por Intelligent Systems. Tras el éxito de el anterior juego de la saga, Three Houses, y la expansión internacional de la franquicia fuera de Japón con productos como Fire Emblem Heroes, tiene una amplia y consolidada base de seguidores por todo el mundo. En el juego encarnamos a Alear, vástago del dragón divino, que debe hacer frente a la amenaza de Sombron, el dragón de las sombras que planea arrasar el continente.
- **Forspoken:** Desarrollado por el estudio Luminous Productions, subsidiaria de la empresa de videojuegos Square Enix, este juego supone su primer proyecto individual tras su colaboración en el desarrollo de Final Fantasy XV. El juego narra la historia de Frey, una joven neoyorquina que acaba en el hermoso y cruel mundo de Athia. Mientras averigua cómo volver a casa, deberá usar sus nuevas dotes mágicas para recorrer paisajes enormes y enfrentarse a seres monstruosos.
- **Hi-Fi Rush:** Shinji Mikami, creador de la franquicia de videojuegos Resident Evil, fundó en 2010 su propio estudio de videojuegos alejado del paraguas de Capcom. En un cambio de tono radical con respecto a su anterior obra, Hi-Fi Rush cuenta la alegre historia de Chai, un chico que sueña con ser estrella del rock y que, tras un extraño accidente, ve como su corazón es sustituido por un reproductor de música que potencia los poderes de su brazo robótico cuando se mueve al ritmo de la música. Este curioso *hack and slash* rítmico de plataformas supone el primer juego del estudio, Tango Gameworks, y es publicado por Bethesda Softworks. El juego se anunció y publicó de inmediato en una presentación de Microsoft, añadiendo que además los suscriptores de Game Pass podrían jugar a él sin ningún coste adicional, debido a que la multinacional estadounidense adquirió Bethesda en 2020 como parte de su estrategia para fomentar el desarrollo de juegos exclusivos [67].

3.2. Entorno de negocio

En esta sección se presenta una introducción al análisis de datos sociales, dando una terminología básica de la teoría subyacente a dicho campo, así como algunos casos de aplicación de dichos estudios en la actualidad.

3.2.1. Análisis de datos sociales

Como ya se comentó en la introducción, hoy en día generamos cada vez más información diariamente, bien sea en redes sociales, Internet en general o incluso mediante dispositivos pertenecientes al denominado *Internet of Things* (IoT). En vista de la ingente cantidad de datos que están a nuestra disposición, cabe preguntarse qué conclusiones podemos extraer de ellos, pues gran parte de estos datos pueden no tener ninguna relevancia. Por ello, es necesario identificar primero a qué hacen referencia términos como *conocimiento*, *utilidad* o *relevancia*.

Tradicionalmente, las definiciones de conocimiento provienen del ámbito de la ciencia de la información. El concepto de conocimiento suele representarse como una pirámide conocida como **jerarquía del conocimiento**. Dicha pirámide se cimenta sobre los datos, permitiendo construir a partir de ellos información, que acaba desembocando en el conocimiento mismo.

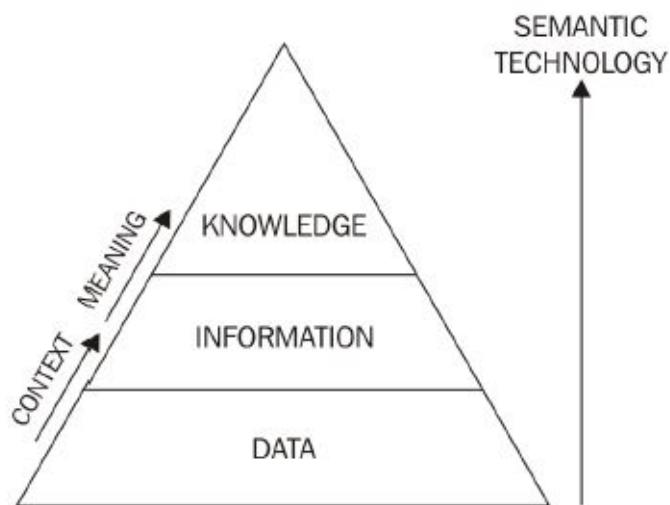


Figura 3.2: De los datos sin procesar a la información semántica

Por lo tanto, escalar esta pirámide consiste en pasar de los datos sin procesar al conocimiento, lo cual se logra a partir de analizar el contexto y significado de los datos. Así, la tecnología que construimos logra un entendimiento más profundo de los datos originales y, más importante aún, de los usuarios que los han generado. En otras palabras, pasa a resultar útil.

En este contexto, el **conocimiento útil** es aquel que puede entenderse como conocimiento *procesable*, que es aquel que permite a los responsables de la toma de decisiones poder implementar estrategias de negocio de manera justificada [15].

El conjunto de datos extraídos de los medios de comunicación social se conocen como *social data* ó **datos sociales**, los cuales pueden ser **estructurados**, si son numéricos o cuantificables; o **no estructurados**, como es el caso de vídeos o imágenes, los cuales son una gran fuente de información pero resultan más difíciles de analizar. El proceso de aplicar métodos rigurosos para dar sentido a los datos sociales se denomina *social data analytics* ó **análisis de datos sociales** [17], que va desde la construcción de modelos para comprender y analizar los datos, hasta la visualización de los resultados obtenidos mediante dichos modelos.

En la sección 2.1.1 ya se profundizó en los diferentes pasos que deben seguirse para la correcta planificación y ejecución de proyectos de *data science*, disciplina bajo la cual se encuadra el análisis de datos sociales, por lo que para ver un análisis detallado del proceso basta referirse a dicha sección.

3.2.2. Casos de aplicación

Veamos ahora algunos ejemplos de aplicación de técnicas de análisis de datos. Al ser metodologías extrapolables tanto al análisis general como al aplicado específicamente a los datos sociales, presentaremos ejemplos de ambos casos para ver la potencia y versatilidad de dichas técnicas:

- **Política.** 2012. Barack Obama se presenta a las elecciones generales de Estados Unidos intentando revalidar su mandato tras haber sido Presidente los últimos 4 años. La principal diferencia con respecto a su campaña anterior es que ahora su equipo de análisis de datos está conformado por 20 personas, quintuplicando así el personal en dicho apartado con respecto a los anteriores comicios. Gracias a esto, el equipo de campaña de Obama es capaz de recopilar toda la información que los ciudadanos estadounidenses publican en la red, identificando qué temas son los que más importancia tienen para sus votantes y así incidir en ellos durante la campaña; o captando el voto de los indecisos gracias a mensajes publicitarios en espacios que éstos puedan ver fácilmente, como son las pausas publicitarias de la serie *The Walking Dead* o en hilos del foro Reddit [46].

Con el uso de estos métodos, Obama ganó en estados clave como Ohio, y logró mantener su puesto como Presidente de los Estados Unidos de América. Y todo ello, apoyado en el uso del análisis de datos sociales y estrategias *data-driven*.

- **Deporte.** Una de las historias recientes más famosas del fútbol inglés es cuando, el antiguo analista del Liverpool Ian Graham, fue a conocer por primera vez al entrenador que acababa de fichar por el equipo esa temporada: Jürgen Klopp. En dicha reunión, Graham habló a Klopp sobre diversos partidos de la temporada pasada del entrenador alemán con su antiguo equipo, el Borussia Dortmund, en los que la escuadra bávara debería haber ganado a sus rivales con holgura, pero factores como la

suerte o el gran rendimiento de sus contrarios desembocaron en resultados adversos para el equipo de Klopp. Al ver los amplios conocimientos de Graham, Klopp se sorprendió por la cantidad de partidos que había debido de ver el analista inglés. No obstante, Graham no había visto ni uno solo de los encuentros: únicamente había recurrido a los informes generados por el equipo de análisis para poder entender perfectamente qué había ocurrido en cada partido [56]. Gracias a esto, el entrenador alemán quedó convencido del poder de esta innovación y se mostró abierto a que el equipo de análisis de datos le asesorase en la confección de la plantilla, así como en el perfeccionamiento de sus sistemas de juego. Dicha colaboración resultó tremadamente fructífera, pues a día de hoy el equipo de Merseyside ha recuperado su estatus como uno de los grandes equipos europeos, conquistando por el camino una liga y una Copa de Europa.

Esta historia supone el ejemplo perfecto de cómo hasta las instituciones que antaño resultaban tan anquilosadas como los equipos de fútbol, obcecadas en los métodos y sistemas clásicos que habían ido funcionando a lo largo de los años, se han visto beneficiadas de enfoques innovadores como el análisis de datos.

Sin embargo, esta metodología ha resultado ser un modelo efectivo no sólo para los grandes equipos europeos, si no que se muestra igualmente válida para aquellos más modestos. Aplicando estas herramientas, equipos como el Brentford o el Brighton, cuyos presidentes son propietarios de empresas de análisis de datos y emplean dicha metodología en sus propios clubes, han logrado escalar desde las divisiones inferiores del fútbol inglés hasta lograr colarse esta temporada entre los 10 mejores equipos de la Premier League. Todo ello, gracias al uso de análisis de datos para fichar a sus jugadores o para encontrar debilidades y posibles mejoras en sus sistemas de juego [12].

- **Análisis retail.** Tal y como se vió en la sección 3.1, la venta al por menor puede obtener grandes beneficios a partir del análisis de datos, permitiendo a las empresas adecuarse a las tendencias del mercado y satisfacer las nuevas necesidades de los consumidores. Dicha metodología es extrapolable también a las grandes marcas, como es el caso de Amazon, que a partir de los datos de sus consumidores es capaz de ofrecer recomendaciones de productos personalizadas que se adecúen más a sus preferencias.
- **Medicina.** El análisis de datos masivos también puede resultar útil para controlar y predecir la evolución de epidemias y brotes de enfermedades. A través del uso de teoría de grafos, basta identificar a los nodos más influyentes de una red para limitar en la medida de lo posible la propagación de enfermedades. Dicho uso puede verse en estudios como [5] en el que, entre otras muchas aplicaciones, se habla sobre la detección y rastreo de contactos para minimizar la propagación del COVID-19.

3.3. Contexto científico-técnico

3.3.1. Machine Learning

Dentro de la Inteligencia Artificial, el *Machine Learning* o Aprendizaje Automático es la disciplina encargada del estudio y desarrollo de algoritmos que, a partir de los datos, puedan hacer predicciones sobre los propios datos. En otras palabras, se fundamenta en realizar predicciones basadas en propiedades conocidas de los datos [15].

Los programas basados en el uso del Aprendizaje Automático son una constante en nuestro día a día, pasando desde tareas tan simples como decidir si un correo electrónico es *spam* o no, a otras más delicadas como analizar transacciones bancarias para poder identificar posibles intentos de estafa.

A grandes rasgos, podemos diferenciar dos categorías principales dentro de las metodologías más populares del *Machine Learning*. A pesar de que esto supone una simplificación que no es capaz de abarcar toda la profundidad y amplitud dentro del Aprendizaje Automático, resulta un buen punto de partida para apreciar algunos de sus aspectos técnicos.

- **Aprendizaje supervisado.** Esta metodología se emplea para resolver problemas de clasificación, en los que los datos contienen atributos adicionales aparte del que se desea predecir. En el escenario ideal, el modelo construido asociaría a cada entrada la salida esperada, sin posibilidad alguna de fallo. Para ello, se construye un modelo matemático usando unas entradas de prueba que servirán como conjunto de entrenamiento, para que después el modelo intente inferir el atributo deseado, denominado clase o etiqueta, de un conjunto de datos distinto de los que ha usado para entrenarse (conjunto de datos de prueba). Las técnicas más habituales de aprendizaje supervisado son Naive Bayes, máquinas de vectores de soporte o modelos pertenecientes a la familia de las redes neuronales, como los perceptrones o las redes convolucionales.
- **Aprendizaje no supervisado.** Al contrario del caso anterior, los métodos no supervisados se emplean en problemas que tienen un conjunto de datos no etiquetados como entrada, por lo que se desconoce el resultado final. El ejemplo típico de este tipo de retos son los llamados problemas de agrupamiento o *clustering*, en los cuales se intenta encontrar patrones subyacentes bajo los datos y así ser capaces de dividirlos en grupos, o detectar aquellos elementos que no comparten características similares con el resto (detección de *outliers*). Ejemplos de esta clase de algoritmos serían k-medias, k-medianas o mapas de Kohonen.

3.3.2. Procesamiento del Lenguaje Natural

Natural Language Processing (NLP) o Procesamiento del Lenguaje Natural (PNL) es la disciplina relativa al estudio de métodos y técnicas para el análisis, comprensión y generación de lenguaje natural; es decir, aquél hablado o escrito por los seres humanos [15].

Esta definición evoca irremediablemente a los orígenes de la Inteligencia Artificial (IA), cuando Alan Turing presentaba en 1950 su test homónimo para discernir si una máquina poseía o no inteligencia. En él, una persona mantenía una conversación por escrito con dos agentes, un humano y una máquina. Si el evaluador no era capaz de distinguir si las respuestas dadas a una serie de preguntas habían sido dadas por el humano o por la máquina, se determinaba que la máquina poseía inteligencia. La principal complejidad del test radica en que el ordenador debe poseer competencias de procesamiento de lenguaje natural, representación del conocimiento, razonamiento y aprendizaje automático, para ser capaz de pasar la prueba. Podemos entonces ubicar PNL como una rama dentro de la IA que, a partir del uso de técnicas de aprendizaje automático y los conocimientos de la lingüística, permite analizar, comprender e incluso emular las maneras de expresarse propias del ser humano.

A pesar de que actualmente han surgido bots conversacionales e inteligencias artificiales altamente avanzadas como ChatGPT, al intentar mantener con una conversación sostenida con ellos, siguen pudiéndose apreciar ciertas asperezas que dejan ver que nuestro interlocutor no es en verdad un ser humano. Es por ello que el procesamiento del lenguaje natural se mantiene a día de hoy como un área de investigación en auge, con cada vez más posibles aplicaciones a diversos ámbitos.

En el contexto de los medios sociales, resulta evidente que existe una gran cantidad de información textual que está esperando a ser recolectada y analizada. La cantidad de información aumenta cada día sin cesar bien sea en forma de conversaciones en redes sociales o como reseñas de productos dadas por usuarios. No obstante, la transición desde los datos sin procesar es una labor complicada. Por suerte, a partir del uso de técnicas PNL, podemos ser capaces de descubrir los principales temas de los que se hablan en redes o identificar el sentimiento asociado a las reseñas de una página de comentarios.

El procesamiento del lenguaje natural está compuesto por una amplia variedad de herramientas y metodologías. Es por ello que, a lo largo del trabajo y según vayamos recurriendo a ellas, se irán presentando las diferentes bases teóricas en cada uno de los pasos necesarios para el análisis de datos sociales a partir de una perspectiva PNL.

3.4. Estado del arte

Al ser un estudio teórico sobre un tema específico, resulta complicado hacer una comparativa directa con trabajos relacionados. En cambio, existe una amplia variedad de artículos y publicaciones donde se llevan a cabo diferentes fases del proceso de análisis de datos sociales. Por lo tanto, a continuación se presentan una serie de estudios cuyo contenido se corresponde de manera directa con diferentes partes de este mismo trabajo, para así observar la metodología y herramientas empleadas por los autores, así como las posibles conclusiones que se puedan extraer de los mismos.

3.4.1. *Web scraping* de Metacritic

Proyecto del científico de datos Martín Pellarolo, publicado en 2018 en su página de GitHub [52]. El trabajo se basa en el análisis de datos de videojuegos obtenidos de la página de reseñas Metacritic. Más concretamente, obtiene información detallada y reseñas de los juegos disponibles para Nintendo Switch, PlayStation 4 y Xbox One, las consolas predominantes de dicha generación. El estudio se compone de cuatro cuadernos de Jupyter:

1. ***Data scraping.*** A través del formato DOM que siguen las páginas de Metacritic, pueden extraerse el conjunto de juegos disponibles de cada plataforma. Para ello, de la página principal se extrae una lista del conjunto de títulos y después se procede a obtener la información individual para cada uno de ellos (las críticas de prensa y usuarios).
2. ***Data cleaning.*** El objetivo de este cuaderno consiste en el cribado de información no relevante y la unificación de todos los datos en un formato único, para luego poder trabajar fácilmente con la información recabada.
3. ***Exploratory data analysis.*** Estudio comparativo entre las diferentes consolas y sus lanzamientos, analizando datos en términos de exclusivos, juegos mejor valorados o la distribución de juegos de cada plataforma conforme al sistema de clasificación del contenido por edades (ver figura 3.5). También se analiza la datos generales como en qué época del año salen más juegos o si existe una gran diferencia entre la valoración por parte de la crítica y los usuarios (ver figuras 3.3 y 3.4). Se realiza un análisis exploratorio de los datos para tratar de descubrir sus características principales.
4. ***Sentiment analysis.*** Creación de un modelo para la clasificación de reseñas en base al texto de las mismas, para dictaminar si son positivas o negativas. Para ello, se estudian dos modelos diferentes para su implementación: Naive Bayes multinomial y redes convolucionales.

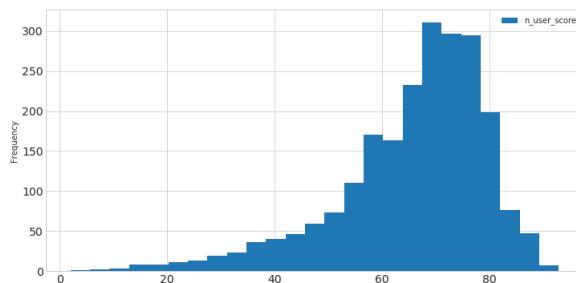


Figura 3.3: Puntuaciones de prensa en Metacritic

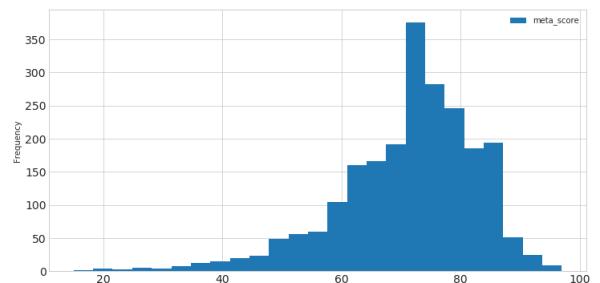


Figura 3.4: Puntuaciones de usuarios en Metacritic

Las similitudes entre el proyecto de Pellaro y el propuesto en este documento resultan evidentes. Sin embargo, nuestro estudio se enfoca más en el análisis de datos provenientes de redes sociales, considerando los datos a través de Metacritic como una vía de información más sobre la que trabajar. Por este motivo, tampoco se profundizará tanto en el análisis de reseñas como hace Pellaro en su trabajo.

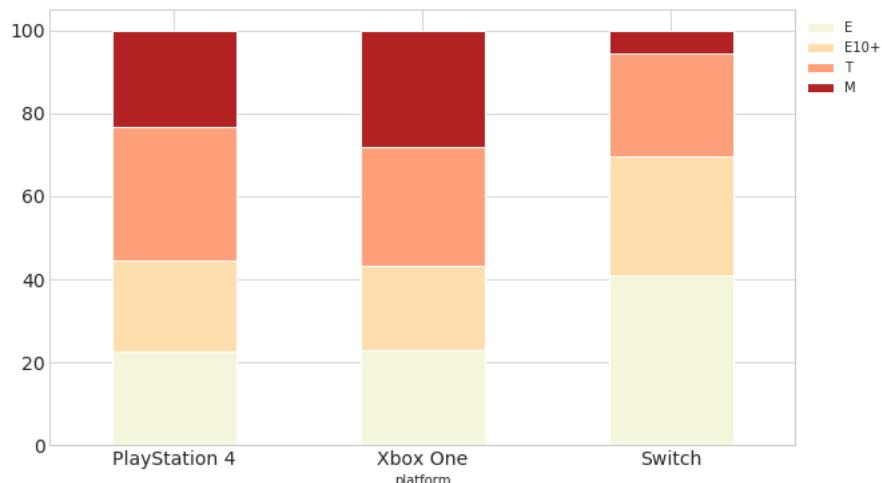


Figura 3.5: Proporción de juegos según su clasificación por edades para cada consola

3.4.2. *Topic modelling* de un conjunto de tweets

La página web Medium es una plataforma online norteamericana que supone uno de los máximos exponentes del **periodismo social**, páginas que combinan la publicación de artículos por parte de los usuarios con otros redactados por profesionales especializados, siendo estos últimos una manera de incentivar a los visitantes a que se suscriban para que puedan acceder a contenido similar. Dentro de este segundo grupo, se encuentra un artículo del ingeniero informático francés Clément Delteil, especializado en IA. En su artículo, publicado originalmente en la página especializada *Towards AI*, Clément presenta una guía paso a paso para analizar un conjunto de medio millón de tweets sobre la figura de Elon Musk [19].

A lo largo del artículo, se muestra cómo construir un cuaderno Google Colab para ejecutar en el propio navegador todo el estudio, implementado usando Python. Tras realizar un preprocesado sencillo de los datos, se analizan los términos más frecuentes para poder extraer los temas predominantes en el texto y, una vez obtenidos, realizar también un análisis básico de sentimientos sobre ello. La principal fortaleza de este estudio con respecto al nuestro reside en el análisis de comunidades y entidades pues, tal como puede verse en la figura [3.6], se pueden obtener cuentas mencionadas en los tweets e incluso analizar el sentimiento asociado a ellas. No obstante, dicho análisis resulta posible al tratarse de entidades u organizaciones reales, pero como en nuestro estudio se analizan productos

concretos -para los cuales resultaría inviable crear una cuenta oficial específica de cada uno-, dicho análisis no resulta posible aquí. Además, la brevedad del artículo obliga al autor a constreñir más la longitud de los temas abordados, por lo que acaba resultando un análisis más superficial.

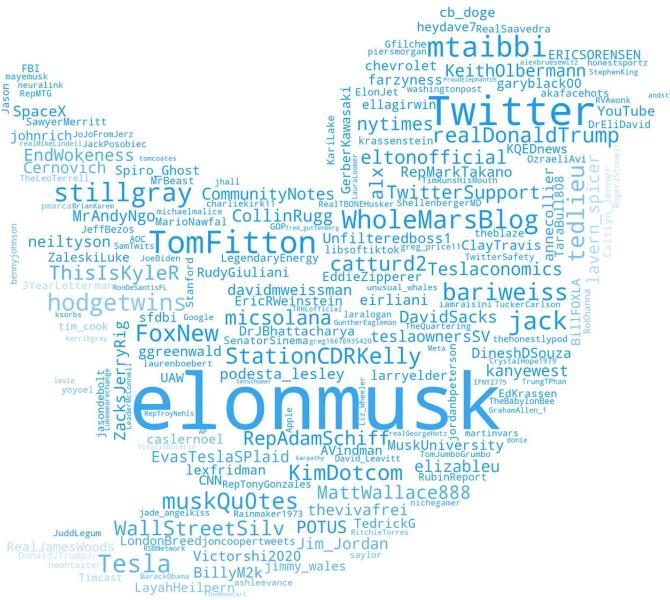


Figura 3.6: *Wordcloud* con las cuentas de usuario más mencionadas en los tweets

3.4.3. *Sentiment analysis* de reseñas de Amazon

De nuevo, dentro de la página Medium, encontramos el estudio realizado por el científico de datos Enes Gokce, de la universidad de Penn State. En el trabajo publicado originalmente en la página especializada *Towards Data Science*, Enes realiza un análisis de sentimientos sobre un conjunto de reseñas de productos de Amazon [28]. El análisis se desarrolla a lo largo de un cuaderno de Jupyter, el cual conforma junto con otros cuatro un proyecto completo de estudio realizado usando PNL, que se encuentra disponible en su página de GitHub [27].

Sin embargo, tanto los cuadernos como su presentación final están enfocados fundamentalmente al análisis de sentimientos, por lo que será este caso el que analizaremos aquí. A partir de los datos obtenidos, los cuales han sido ya cribados y analizados en los dos cuadernos previos, se realiza un análisis de las reseñas obtenidas y comparándolas con las puntuaciones dadas. También se analizan factores que a primera vista podrían parecer no tan relevantes, como es la longitud de las propias reseñas en la valoración final percibida del texto (ver figura 3.7). Además, se considera también la influencia de la subjetividad y se analizan los casos extremos presentes en el análisis.

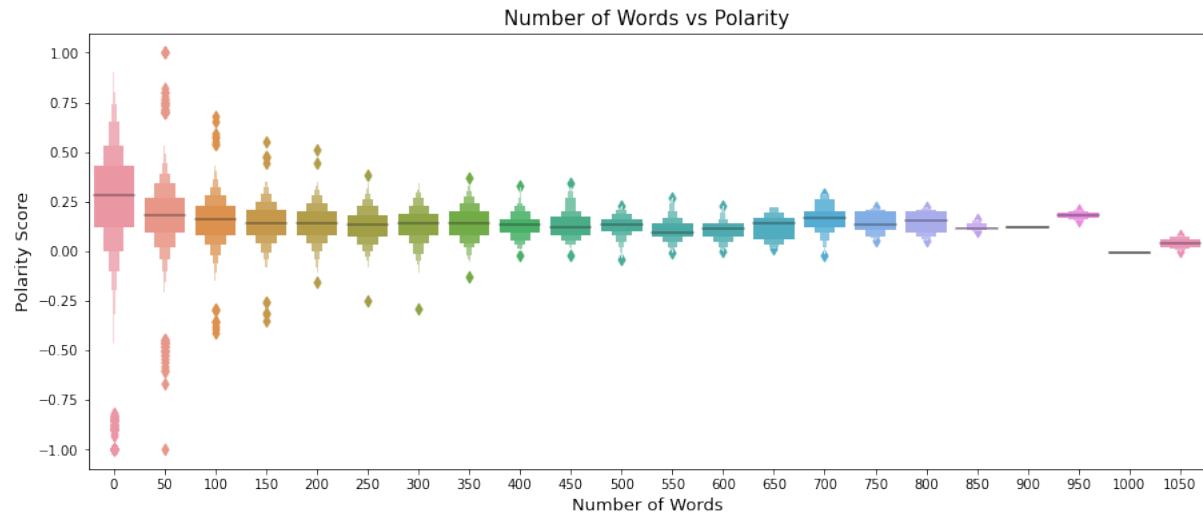


Figura 3.7: Valoración de las reseñas en función de la longitud de las mismas

Tal y como comenta el propio Gokce, la limpieza de los datos se realizó de manera algo superficial, por lo que el uso de técnicas como *stemming* o *lemmatization* permitiría obtener unos resultados más certeros. Como último apunte, en los cuadernos tampoco se muestra la manera de obtener los datos usados para el análisis, parte fundamental dentro de cualquier proyecto de análisis de datos.

Parte II

Desarrollo de la propuesta y resultados

Capítulo 4

Data extraction

4.1. APIs

Una **API** (*Application Programming Interface*) o interfaz de programación de aplicaciones es la manera de intercambiar datos entre un servicio y el programador o usuario. En el contexto de los medios sociales, permite a éstos compartir datos con desarrolladores de aplicaciones de terceros. No obstante, la popularización de la ciencia de datos ha convertido a las APIs en una manera de minar información de manera efectiva para la creación de conocimiento [17].

A pesar de que para cada medio social se implementa de manera distinta con sus respectivas particularidades, en la actualidad existen dos tipos principales de APIs:

- **RESTful:** El tipo de API más comúnmente proporcionado por los medios sociales. La información que recoge es estática y se recupera a partir de los datos históricos. **REST** (*REpresentational State Transfer*) es una arquitectura de software que impone una serie de restricciones a las comunicaciones basadas en el protocolo HTTP para la transferencia de datos. Los métodos más importantes son GET y POST, que permiten la obtención y publicación de datos de máquinas lejanas, respectivamente.
- **Stream:** Permite la obtención de datos en tiempo real. El resultado obtenido es prácticamente igual al de los datos históricos.

El empleo de una API permite obtener datos sociales para fines comerciales y de marketing, o da la posibilidad a desarrolladores de integrar en sus proyectos los medios sociales en cuestión. Sin embargo, el uso de APIs presenta una serie de **limitaciones** que deben tenerse en consideración:

- **Límites de tarifa.** Debido a limitaciones de infraestructura y de negocio, las compañías limitan la cantidad de datos que entran o salen de sus sistemas. Por lo tanto, esto obliga a los usuarios a realizar una planificación minuciosa a la hora de obtener datos.

- **Cambios en la API.** Al pertenecer a entidades privadas, éstas pueden modificar o limitar el acceso a su API en cualquier momento, lo que obliga a los expertos en análisis de datos a estar preparados para amoldarse a posibles casos futuros. El ejemplo más claro de esto es el cambio a un modelo de pago que experimentó la API de Twitter tras la compra de la compañía por parte de Elon Musk [34].
- **Legalidad.** El uso de datos provenientes de API que no se adecúen a las políticas de uso que establece la empresa puede conllevar acciones legales, por lo que es de vital importancia ceñirse a estos marcos a la hora de desarrollar proyectos que empleen estas herramientas.

Conexión a una API

Para poder realizar la conexión a la API de un medio social, es necesario hacer una configuración previa, la cual depende de la plataforma en cuestión. No obstante, el proceso general suele poder resumirse en los siguientes pasos:

1. **Registro en la aplicación.** Dar información personal y relativa a los objetivos con los que se planea usar la API. Una vez hecho, se generan unas claves para poder acceder a las funcionalidades de la API, denominadas **claves de autenticación o consumidor**.
2. **Autenticación** a través de las claves de autenticación generadas en el paso previo.
3. **Búsqueda de endpoints.** En función del proveedor, los *endpoints* de cada API varían, por lo que es necesario leer atentamente la documentación provista por las empresas para identificar los *endpoints* que serán necesarios durante el desarrollo del proyecto.

De todos estos pasos, el que puede resultar más complejo es la **autenticación**. Por suerte, hoy en día este proceso se ha unificado para casi todas las plataformas gracias al uso de OAuth.

OAuth es un protocolo de autorización que permite a los usuarios compartir datos con una aplicación sin necesidad de usar su contraseña. Su principal ventaja es que permite el acceso a terceros en función de las limitaciones de su tarifa, estableciendo una conexión estandarizada y segura. Puede realizarse como usuario o como aplicación (sin necesidad de contexto de usuario). Para la extracción de datos, este último caso es el que se utiliza más frecuentemente, y requiere de varios pasos previos antes de poder obtener cualquier clase de información:

1. **Creación de una cuenta de usuario o desarrollador.**
2. **Creación de una aplicación.** Una vez se dispone de una cuenta, se puede acceder al panel de control o consola de desarrollador, la cual da acceso a todas las funcionalidades para gestionar la cuenta en cuestión, así como crear y borrar otras

aplicaciones o comprobar el uso de las cuotas por parte de la aplicación. Para acceder a este *dashboard*, es necesario haber creado primero una aplicación. La figura 4.1 muestra el panel de control de una cuenta de desarrollador de Twitter.

3. **Obtener *tokens* de acceso.** Generar los *tokens* de acceso de la aplicación creada para poder conectarse después a la API. En función de la compañía, puede ser necesario especificar permisos adicionales dentro de las peticiones HTTP realizadas o indicar cuál será el alcance de las acciones realizadas antes de generar los *tokens* de acceso.
4. **Conexión a la API.** Tras haber obtenido los *tokens* en el paso previo, ya se pueden realizar peticiones teniendo siempre en cuenta las limitaciones de cuota para cada plataforma.

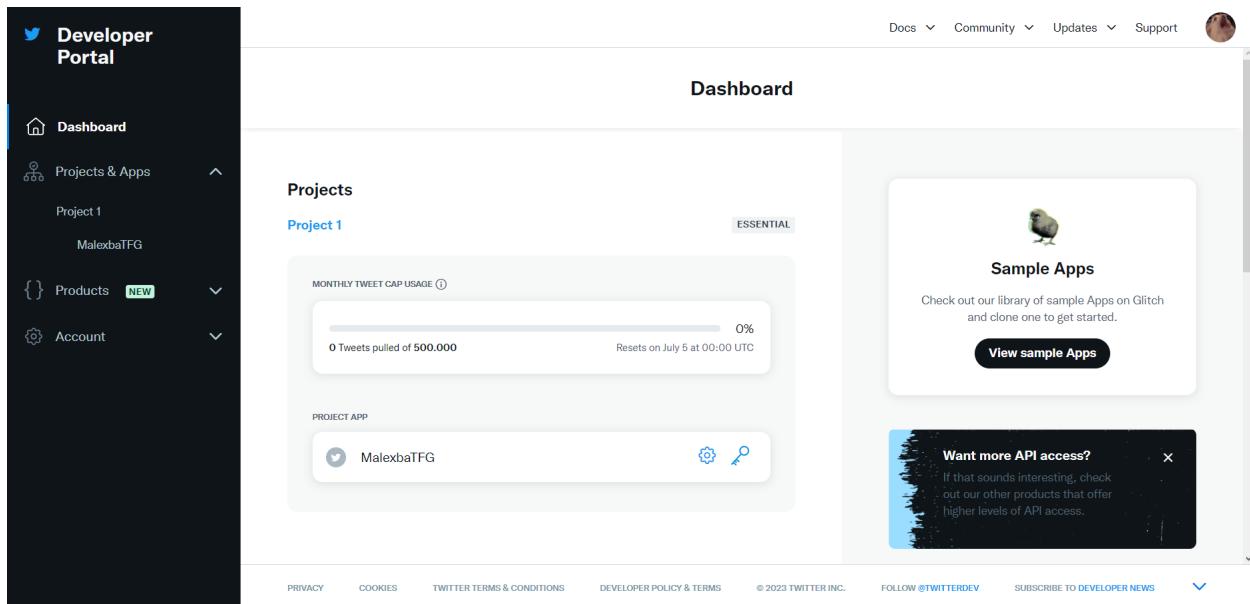


Figura 4.1: *Dashboard* de una cuenta de desarrollador de Twitter

4.1.1. Twitter

El *microblogging* es un servicio online mediante el que los usuarios pueden compartir mensajes, enlaces a sitios externos, imágenes o incluso vídeos, para que sean visibles a todos aquellos suscriptores al servicio. En contraposición con los blogs tradicionales, el *microblogging* se basa fundamentalmente en textos cortos. Algunos de los más conocidos son Tumblr o Mastodon, aunque a día de hoy la plataforma por excelencia de *microblogging* sigue siendo Twitter.

Lanzada en 2006, Twitter tiene cada día más de 200 millones de usuarios activos, cuya franja de edad ronda de media los 30 años. Es especialmente popular en Estados Unidos y Japón, pues ya sólo en estos dos países alcanza los 120 millones de usuarios. Según las estadísticas, el usuario medio de Twitter pasa de media 6 minutos en la aplicación, habitualmente para buscar noticias o mantenerse entretenido [18]. Gracias a la sencillez que otorga para acceder y publicar entradas, supone uno de los mayores conjuntos de datos de contenido generado por usuarios. Dentro de este contexto, debemos identificar la terminología básica dentro de Twitter [26]:

Tweet Cada mensaje publicado en Twitter. Su contenido puede ser imágenes, vídeos o links a otras páginas, aunque principalmente suele ser texto. En sus orígenes existía una limitación de 140 caracteres, que a día de hoy se ha extendido hasta los 280. No obstante, los usuarios de Twitter Blue pueden publicar mensajes de hasta 10.000 caracteres de longitud.

Usuario Un usuario debe estar registrado para poder publicar mensajes. En el momento de registrarse, es necesario indicar un nombre de usuario que será el asociado a los mensajes publicados.

Mención En un tweet puede mencionarse a otro usuario especificando el nombre de usuario de quién se quiere mentar tras un @ (@usuario_mencionado)

Respuesta Un tweet marcado como respuesta va ligado a otro tweet, permitiendo crear **hilos** (cadenas de tweets enlazados como respuestas). También se indica mediante menciones a quién se está respondiendo. En función de las opciones de privacidad seleccionadas por el autor del tweet orginal, otros usuarios pueden responder o no a sus tweets.

Seguidor Usuario que sigue a otro, así como a su actividad. También pueden activarse las notificaciones, para recibir de inmediato información sobre la actividad de otro usuario.

Retweet Redistribuir un tweet. Permite al usuario volver a publicar el tweet en su perfil, manteniendo los atributos originales de éste. Dentro de este comando, existe también la opción de mencionar el tweet, creando el usuario uno nuevo que mantenga una referencia al tweet original del que se habla.

Me gusta Método de interacción para indicar que una publicación es del agrado del usuario, fomentando su visibilidad para otros miembros de Twitter.

Visualizaciones Contador con estadísticas con el número de gente que ha visto el tweet, así como otras más específicas relativas a si la publicación ha provocado una visita al perfil o cuántas veces se ha reproducido un contenido multimedia, en caso de adjuntarlo.

Hashtag Método para etiquetar mensajes en función de temas. Se generan de manera espontánea por los usuarios y permiten aumentar la visibilidad de las publicaciones, al agruparlas por temáticas y facilitar su búsqueda.

Timeline Página de inicio de cada usuario en la que puede ver distintos tweets. Desde el año 2023 se ha dividido en la sección "Para ti", donde se muestran tweets que el algoritmo de Twitter considera que pueden ser interesantes para el usuario, y "Siguiendo", en la que se muestra la actividad de los usuarios seguidos.

Perfil de usuario Página principal del usuario, donde se pueden ver datos como su nombre, una breve descripción de su perfil o las cuentas a las que sigue/le siguen. Se compone de cuatro subventanas: "Tweets"(incluye retweets), "Tweets y respuestas"(muestra únicamente las publicaciones hechas por el usuario), "Multimedia"(que aglutina todas las publicaciones que incluyen contenido multimedia) y "Me gusta"(recoge todos los tweets los cuales el usuario ha indicado que le gustan).

Privacidad Opciones de twitter sobre la visibilidad con respecto al mundo. Aparte de si un perfil es privado o no, se incluyó también el concepto de círculos, que permite la publicación de mensajes para que sólo los usuarios seleccionados puedan verlos.

4.1.2. Twitter API

Twitter ofrece a los desarrolladores varias APIs distintas: la API Rest, la API Streaming (que permite obtener tweets en tiempo real) y la API Ads. Este trabajo se enfocará en el uso de la primera. Documentación detallada, así como guías de uso de la misma, pueden encontrarse en [20]. A la hora de realizar peticiones, es importante tener en cuenta los límites de tarifa específicos de cada comando, los cuales pueden consultarse en [70].

Para poder hacer uso de la API, es necesario seguir los pasos de conexión y autenticación indicados en 4.1. En este caso, para conectarse a la API se hará uso de la librería de Python **Tweepy** [69], que permite acceder fácilmente a la API de Twitter.

Tweepy emplea como modelo de autenticación **OAuth2**, el cual comparte los objetivos de su anterior versión, pero ha sido construido completamente desde cero. Para conectarse, basta con instanciar un cliente de Tweepy indicando el *bearer token* de la aplicación y el formato en que se desea obtener los datos (en nuestro caso, como un diccionario). A la hora de realizar peticiones con el método correspondiente, se deben especificar los campos que se desean obtener en cada petición en términos de tweet, usuario, ubicación y contenido multimedia. No obstante, debido a que muchos de éstos son opcionales o pueden no estar accesibles por cuestiones de privacidad, los únicos campos que se obtendrán siempre son los del tweet. Además, si lo que se busca es obtener tweets, deben especificarse las opciones de paginación, pues la descarga de éstos debe realizarse en peticiones sucesivas de un tamaño limitado para evitar sobrepasar los límites de tarifa y ser considerados como usuarios abusivos. Como ejemplo, una petición realizada con las opciones especificadas en 4.1 devolvería un total de 50.000 tweets, en caso de que se hubiesen generado suficientes publicaciones en la semana de plazo a la que da acceso el plan básico de la API de Twitter.

```

1 import tweepy
2 # Creacion del cliente
3 bearer_token = "aqui_iria_el_bearer_token_de_la_aplicacion"
4 client = tweepy.Client(bearer_token = bearer_token, return_type=
    dict)
5 # Definir los campos a obtener
6 tweetFields = ['id', 'text', 'edit_history_tweet_ids', 'attachments', ,
    'author_id', 'context_annotations', 'conversation_id', 'created_at',
    'entities', 'in_reply_to_user_id', 'lang', 'possibly_sensitive', ,
    'public_metrics', 'referenced_tweets', 'reply_settings', 'source', ,
    'withheld']
7 userFields = ['id', 'name', 'username', 'created_at', 'description', ,
    'public_metrics', 'verified']
8 placeFields = ['full_name', 'id', 'country', 'country_code', 'geo', ,
    'name', 'place_type']
9 mediaFields = ['public_metrics', 'alt_text', 'type', 'url', 'variants']
10 # Configuracion de paginacion
11 nextToken = None # Token para encadenar las peticiones
12 maxResults = 100 # Maximo de resultados obtenidos por peticion
13 pagTimes = 500 # Numero de peticiones que se realizaran

```

Código 4.1: Ejemplo de configuración de cliente Tweepy

4.1.3. Caso de estudio elegido

Si bien la idea general del Trabajo Fin de Grado fue dada por la empresa, la elección de una temática específica de estudio quedó ya a disposición del propio estudiante. En un primer momento, se tanteó la posibilidad de elegir como objeto de estudio empresas altamente implicadas en tener presencia y relevancia en redes sociales a través de sus *Community Managers*, como es el caso de KFC. No obstante, al iniciar el proceso de extracción de información a través de la API de Twitter, se pudo observar que como máximo podían recabarse tweets de hasta una semana de antigüedad. Dicha limitación temporal motivó a pensar en temas de actualidad sobre los que centrar la recogida de información, proceso que se realizó la primera semana de febrero de 2023, dando dos temáticas fundamentales:

- **Videojuegos.** Tal y como se mencionó en la sección 3.1.2, las tres principales compañías de la industria del videojuego acababan de sacar cada una un juego a finales de enero, por lo que en redes sociales habría bastantes publicaciones hablando tanto de las empresas como de sus recientes lanzamientos.
- **Fútbol.** El 31 de enero se cerró el mercado de fichajes invernal en el fútbol europeo. En la liga inglesa, prácticamente todos los clubes más importantes del país -aquellos que conforman el denominado como *Big Six*- habían cerrado altas o bajas relevantes en sus plantillas, por lo que los aficionados habrían manifestado sus opiniones con

respecto a dichos movimientos en redes sociales, generando una cantidad importante de datos a analizar. En este caso, cada uno de los equipos sería una empresa, pues tienen su propia cuenta oficial de Twitter, y cada jugador sería su respectivo "producto a analizar".

Una vez definidos los dos posibles temas sobre los que se podría realizar el estudio, se obtuvo la información relativa a los perfiles oficiales de cada empresa, así como las publicaciones de cada cuenta en la última semana y los tweets que mencionaron a la empresa y/o sus productos.

Como ejemplo, veamos **cómo obtener los tweets que mencionen al videojuego Fire Emblem Engage**, último lanzamiento de la compañía Nintendo. Para ello, se hace una petición especificando que se hable del juego, bien mediante el nombre completo del juego o usando las siglas de la franquicia (FE). Además, limitaremos los resultados a aquellos tweets que no sean ni respuestas ni retweets de ningún tipo (simple o menciones), y hayan sido publicados antes de las 23:00 horas del 7 de febrero de 2023. En 4.2 vemos la implementación de esto a partir de la configuración establecida previamente, obteniendo hasta 50.000 tweets que cumplan dichas condiciones. Las peticiones se encadenan mediante el uso de los tokens presentes en los metadatos de la respuesta dada. Esto permite identificar si sigue habiendo resultados que puedan extraerse o, en caso contrario, debe cesar ya la cadena de peticiones. Además, se incluye una hibernación de 1 segundo tras cada petición para evitar exceder los límites de tarifa.

```

1 q = '((fe engage) OR (fire emblem engage)) lang:en -is:retweet -is:
      reply -is:quote'
2 # Primera petición
3 datos = client.search_recent_tweets(query=q, end_time="2023-02-07T2
      3:00:00Z", tweet_fields=tweetFields, user_fields=userFields,
      place_fields=placeFields, media_fields=mediaFields, next_token=
      None, max_results=maxResults)
4 for i in range(pagTimes-1):
5     tweets = client.search_recent_tweets(query=q, end_time="2023-02
      -07T23:00:00Z", tweet_fields=tweetFields, user_fields=
      userFields, place_fields=placeFields, media_fields=
      mediaFields, next_token=datos['meta']['next_token'],
      max_results=maxResults)
6     # Agregar datos obtenidos
7     if ('data' in tweets): # Comprobar la petición ha tenido éxito
8         for tweet in tweets['data']:
9             datos['data'].append(tweet)
10    else:
11        break
12    # Actualizar metadatos para la siguiente petición
13    datos['meta']['oldest_id'] = tweets['meta']['oldest_id']
14    datos['meta']['result_count'] += tweets['meta']['result_count']
15    if ('next_token' in tweets['meta']): # Comprobar si quedan

```

```
16     tweets que se puedan extraer
17     datos['meta'][‘next_token’] = tweets['meta'][‘next_token’]
18 else:
19     break
20 print(datos[‘meta’][‘next_token’])
21 # Evitar exceder los límites de tarifa
22 time.sleep(1)
```

Código 4.2: Ejemplo de obtención de los tweets relativos a una query específica

4.2. Web scraping

Web scraping o extracción de datos web hace referencia al proceso de extraer información de manera automática de una página web, bien sea gracias al uso del protocolo HTTP o mediante un navegador web. Aunque este proceso puede realizarse manualmente por quien desea obtener los datos, lo habitual es usar bots o *spiders* que realicen una búsqueda sistemática en la Web para encontrar la información deseada. Esta metodología es la que siguen los motores de búsqueda para poder ofrecer a sus usuarios los resultados pedidos.

En el campo de la ciencia de datos, el *web scraping* permite acceder a aplicaciones que no ofrecen una API para facilitar al acceso a la información que contienen, por lo que suponen un medio vital para la obtención de datos. En general, las metodologías empleadas para realizar *web scraping* pueden englobarse en seis categorías:

- **Extracción humana.** Procedimiento más básico, consistente en que una persona analice y extraiga de manera manual la información proveniente de una web. A pesar de ser muy ineficiente e inviable para proyectos a gran escala, en ocasiones puede ser la única manera de obtener información de sitios que usan métodos para prevenir la actuación de bots de extracción automática.
- **Patrones de texto.** Búsqueda a partir de coincidencia de un texto plano prefijado (como usar el comando *grep* en Linux) o expresiones regulares dadas.
- **Análisis del DOM.** Hoy en día, la mayoría de los sitios de Internet son **páginas web dinámicas**; véase, se cimentan en una aplicación que modifica la estructura básica del DOM (*Document Object Model*) de la web para mostrar de manera dinámica el contenido de la misma. Esto produce que el formato de las páginas esté altamente jerarquizado, lo que permite obtener de una forma sencilla su contenido guiándose por dicha estructura.
- **Anotación semántica.** Las páginas pueden incluir metadatos u otra clase de anotaciones semánticas que aporten información adicional sobre su estructura. En caso de que se incluyan en las propias páginas, sería un tipo especial de análisis del DOM. No obstante, a veces dicha información se almacena en una capa semántica aparte, lo que permite realizar un análisis previo antes de realizar la extracción de datos.

- **Visión computacional.** Las aproximaciones más recientes a este campo buscan aplicar técnicas de Inteligencia Artificial y visión computacional para automatizar la extracción humana gracias a estas herramientas.

Para este trabajo, se ha decidido implementar un cuaderno de Jupyter que obtenga información de una web basándose en la estructura DOM de la misma. La manera más común de hacer esto es mediante el uso de *parsers* o analizadores sintácticos como ***Beautiful Soup*** [10], que permite a sus usuarios navegar, buscar y modificar las estructuras de archivos escritos en lenguajes de marcado como HTML o XML.

4.2.1. Metacritic

Metacritic es un sitio web que recopila críticas de películas, programas de televisión, álbumes de música, videojuegos y, antiguamente, libros. Para cada producto, se realiza una media ponderada de las puntuaciones de cada crítica, las cuales siguen un código de colores (verde, amarillo y rojo) para provocar mayor impacto visual.

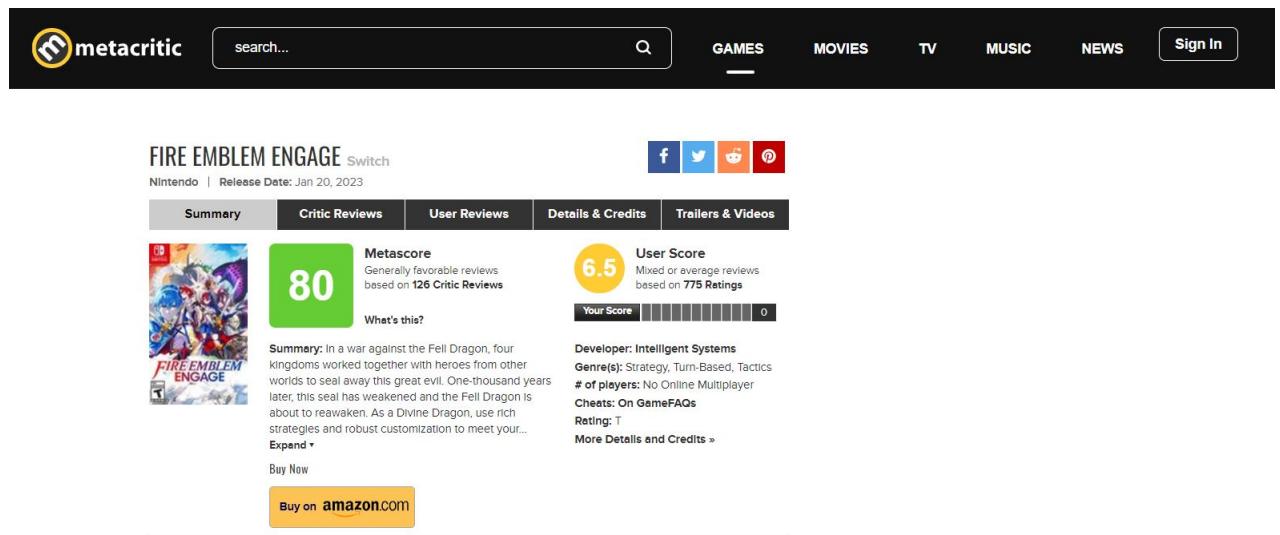


Figura 4.2: Ejemplo de página de resumen de un videojuego

Lanzada en 2001, su idea era recoger un mayor espectro de valoraciones a diversos medios que la web Rotten Tomatoes, la cual se dedicaba únicamente a las críticas cinematográficas. A día de hoy, ha ganado dos premios Webby por su excelencia como sitio web de agregación y está considerado como el principal sitio en línea de recopilación de reseñas de videojuegos, por lo que muchas compañías la utilizan como baremo a la hora de planificar sus proyectos y lanzamientos [59]. Sin embargo, esta clase de fuentes de información deben ser examinadas con precaución, pues en ocasiones pueden adolecer del fenómeno conocido como *review bombing*, consistente en una afluencia masiva de reseñas de usuarios -bien sean positivas o negativas- para afectar de alguna manera la popularidad o la reputación de un producto, servicio o empresa [62].

A pesar de que existen APIs no oficiales para la extracción de datos de Metacritic, debido al carácter didáctico de este trabajo se ha creído conveniente implementar un cuaderno que ejemplifique cómo realizar la extracción de datos de una web, pues la metodología empleada es extrapolable a cualquier otro proyecto de *web scraping* y basta adaptar los métodos empleados al formato de la web a analizar.

4.2.2. Caso de estudio elegido

A pesar de que en la sección 4.1.3 se obtuvieron dos bases de datos distintas, tal y como se comentó en el apartado 3.1.2, el estudio se centrará en **compañías de videojuegos y sus recientes lanzamientos**, pues la existencia de páginas de reseñas como Metacritic permiten obtener información adicional para analizar. Los tres juegos elegidos tienen sendas páginas en Metacritic que recopilan críticas de prensa especializada y usuarios. No obstante, para el caso de Forspoken y Hi-Fi Rush, existe una página adicional para cada uno debido a que ambos títulos fueron lanzados también en PC y no sólo en las consolas de nueva generación de cada una de las compañías. Por lo tanto, recogeremos también las críticas correspondientes a los lanzamientos en ordenador.

Para cada videojuego, se buscará obtener tres archivos de datos que recojan la información general disponible en la página: sumario del total de críticas existentes, reseñas de crítica especializada y reseñas de usuarios. Como ejemplo, explicaremos cómo fue el proceso de obtención de éstas últimas. Para ello, se tuvo que analizar el DOM de la página. Una vez escrutado, se identificó que la estructura de la página se organiza en base a una jerarquía de `<div>` a los que se dota de una clase determinada para identificarlos. En concreto, las reseñas estaban contenidas dentro de una lista ordenada como ítems individuales que seguían el esquema de la figura 4.3.

A la hora de implementarlo, debe también considerarse la posibilidad de ligeras variaciones en la estructura, como es el caso del cuerpo de la reseña. Algunos usuarios realizan críticas con textos de gran longitud, por lo que la página crea dos ``; uno con una vista previa del texto que puede ampliarse al segundo pulsando en los puntos suspensivos, por lo que la reseña completa se ubica en ``.

Además, debido a la gran cantidad de valoraciones que suele haber por parte de los usuarios, pueden ordenarse las críticas en base a diversos criterios como fecha de antigüedad o puntuación. El criterio por defecto lo hace en función de la utilidad de la reseña valorada por la comunidad. Para acceder al resto de reseñas, únicamente se debe cambiar la página pedida añadiendo la `query "?page=n"`, siendo n el número de la página. En caso de que no hubiese más valoraciones, en lugar de la lista ordenada, Metacritic muestra el texto `"There are no user reviews yet - Be first to review {Nombre del juego}."`, lo cual indicará el momento en que se debe finalizar el proceso de extracción.

En la sección 4.3.2 se puede consultar un análisis en mayor profundidad del total de datos obtenidos, considerando también el resumen global de todas las críticas existentes, así como de las reseñas de prensa especializada..

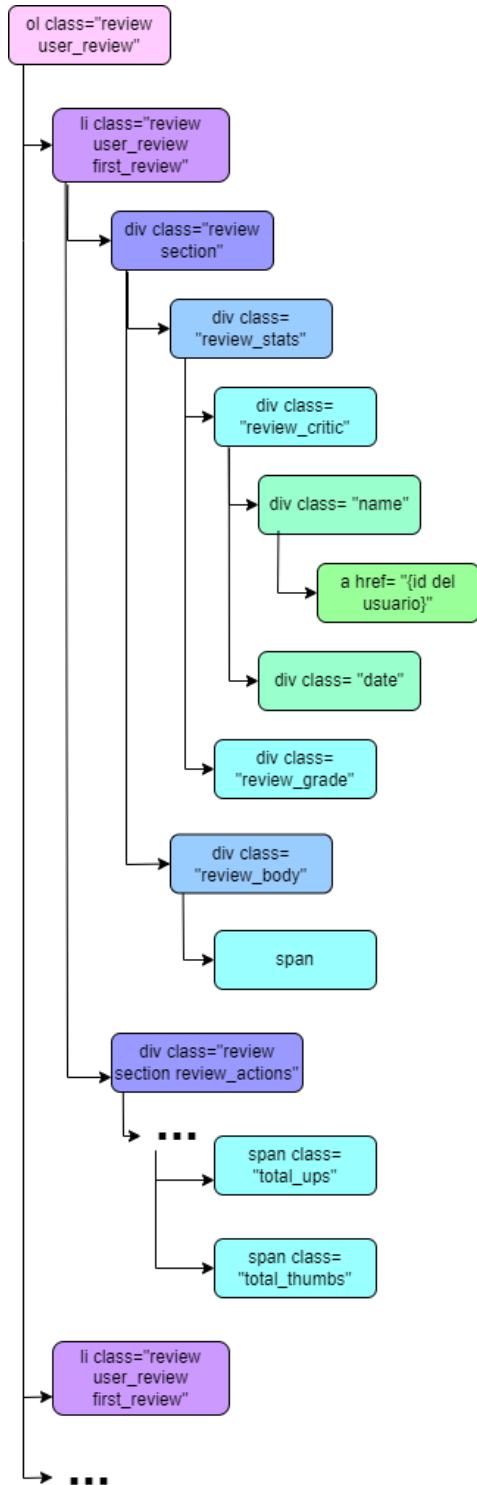


Figura 4.3: Esquema del DOM de las reseñas de usuarios

4.3. Estructura de los datos

Antes de poder realizar cualquier tipo de operación con los datos, resultará crucial hacer un primer análisis de la estructura y formato de los mismos, para ser capaces de comprender de qué información disponemos y cómo podemos hacer uso de ella. En el caso de los datos obtenidos por *web scraping*, en el cuaderno de Jupyter correspondiente ya los guardamos definiendo una estructura que nos resultase beneficiosa, aunque un segundo análisis de los mismos nunca resultará excesivo. Para los datos obtenidos de Twitter, sí será conveniente un escrutinio algo más minucioso, pues se obtuvieron gran cantidad de datos, cada uno con una elevada cantidad de campos y atributos.

4.3.1. Twitter

La información sin tratar obtenida de Twitter se estructura en tres carpetas, conteniendo cada una un fichero cuyo nombre corresponde a una de las empresas o a uno de los títulos a analizar. Éstos son archivos json compuestos por diccionarios, los cuales constan de atributos y valores, o bien otros diccionarios que permiten anidar el contenido de los datos.

- *users*. Información general sobre los perfiles oficiales de las compañías. Incluye el identificador único del perfil (id), además de estadísticas generales como el número de seguidores o la cantidad total de tweets realizados por el usuario.

```
data : <class 'dict'>
    username : <class 'str'>
    name : <class 'str'>
    created_at : <class 'str'>
    verified : <class 'bool'>
    id : <class 'str'>
    description : <class 'str'>
    public_metrics : <class 'dict'>
        followers_count : <class 'int'>
        following_count : <class 'int'>
        tweet_count : <class 'int'>
        listed_count : <class 'int'>
```

Figura 4.4: Formato y datos de un usuario

- *tweets*. Tweets que incluyen las palabras claves indicadas en la petición. La información se encuentra dentro del diccionario "data", pues "meta" son únicamente los metadatos empleados durante la encadenación de peticiones a la API, aunque incluye un total de los resultados obtenidos; "result_count". Cada tweet consta de los siguientes campos, los cuales se pueden encontrar en [72]:

- *author_id*. Identificador único del usuario que ha hecho la publicación.
- *entities*. Las entidades proporcionan metadatos e información contextual adicional sobre los contenidos publicados en Twitter. Son objetos JSON que proporcionan información adicional sobre hashtags, urls, menciones de usuarios... asociados a un Tweet.
- *context_annotations*. Anotaciones semánticas que la plataforma hace sobre el tweet en base a palabras clave, hashtags o menciones relevantes respecto a un tema dado.
- *public_metrics*. Métricas de participación pública (retweets, respuestas, me gusta...) del Tweet en el momento de la solicitud.
- *lang*. Idioma de la publicación.
- *id*. Identificador único del tweet.
- *edit_history_tweet_ids*. Identificadores únicos que indican todas las versiones de un Tweet. Para los Tweets sin ediciones, habrá un ID. Para los Tweets con un historial de ediciones, habrá varios ID, ordenados en orden ascendente reflejando el orden de las ediciones. La versión más reciente es la última posición de la matriz.
- *created_at*. Momento de publicación del tweet.
- *reply_settings*. Muestra quién puede responder al tweet. Las posibles opciones son "*everyone*", "*mentioned_users*" y "*followers*".
- *conversation_id*. ID del tweet original de la conversación, en caso de ser un hilo. Si es una publicación individual sin ninguna clase de interacción, se corresponde con el propio ID del tweet.
- *possibly_sensitive*. Indicativo de si el contenido del tweet es potencialmente sensible, bien sea porque el propio creador lo ha marcado como tal, o porque un agente moderador de Twitter así lo ha considerado.
- *text*. Contenido del tweet (texto) en formato UTF-8.

En total, tanto de Playstation como de Xbox se han obtenido 50.000 tweets, mientras que de Nintendo sólo se ha conseguido obtener 40.000. Dicha diferencia se debe a que Nintendo es una empresa con sede en Japón y especial énfasis en su mercado nacional, por lo que, a pesar de tener gran relevancia, no goza de un impacto tan marcado en Occidente como sus competidores. Con respecto a los títulos, de cada uno de ellos se han obtenido un total de aproximadamente 7.000 tweets para cada juego.

- *usersTl*. Tweets que el usuario ha realizado o ha retuiteado. Su formato es análogo al previo, aunque esta carpeta sólo consta de tres archivos correspondientes a cada uno de los perfiles de cada empresa. La cantidad de tweets de cada empresa en la semana en que se recogieron los datos oscila desde los 2.300 en el caso de Nintendo hasta los 3.000 en el caso de Xbox.

El total de datos recogidos tiene cierta **coherencia con respecto al conjunto obtenidos relativos a la primera división inglesa**, pues de cada equipo se obtuvieron entre 40.000 y 50.000 tweets; mientras que de cada jugador había aproximadamente unos 7.000 tweets, por lo que los dos casos de estudio considerados habrían sido igualmente válidos para la realización del estudio. El caso más curioso es el del traspaso del jugador Jorginho, que pasó de un equipo del *Big Six* a otro (del Chelsea al Arsenal), por lo que el número de tweets obtenidos que hablan sobre él es el doble de lo habitual (15.000).

4.3.2. Metacritic

Los datos obtenidos en el cuaderno previo se estructuraron en base a los dataframes de la librería pandas. **Pandas** es una biblioteca de software para la manipulación y análisis de datos para el lenguaje de programación Python. En particular, ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales. El nombre deriva del término "datos de panel", término de econometría que designa datos que combinan una dimensión temporal con otra dimensión transversal. Se implementa como una extensión de la librería **NumPy**, la cual permite manipular vectores y matrices en Python.

Pandas permite utilizar tanto **series** (vectores indexados) como, por supuesto, **dataframes**. Los dataframes son arrays multidimensionales con etiquetas para las filas y las columnas, que pueden contener datos heterogéneos o incluso nulos. La principal virtud de pandas es que permite manejar los dataframes de manera eficaz, proporcionando una gran cantidad de métodos y operaciones para su manipulación.

	source	link	date	grade	scoreType	text	upThumbs	totalThumbs	helpfulness
0	NagisaNeko	http://www.metacritic.com/user/NagisaNeko	Jan 24, 2023	7	Mixed	I admit that this work is better than the prev...	19	19	1.000000
1	Belonski	http://www.metacritic.com/user/Belonski	Jan 25, 2023	6	Mixed	One of the best Fire Emblems in technical aspe...	17	17	1.000000
2	avantic00	http://www.metacritic.com/user/avantic00	Feb 6, 2023	0	Negative	Combat was great and all, but the amount of cr...	16	17	0.941176
3	Faetori	http://www.metacritic.com/user/Faetori	Jan 25, 2023	5	Mixed	The gameplay itself and the technical portion ...	16	18	0.888889
4	SomethingSom	http://www.metacritic.com/user/SomethingSom	Jan 26, 2023	5	Mixed	Gameplay is fun. The story and characters are ...	15	17	0.882353
...
379	Aurok11	http://www.metacritic.com/user/Aurok11	Mar 1, 2023	10	Positive	Honestly, after reading user reviews of FE Eng...	0	3	0.000000
380	hyper06	http://www.metacritic.com/user/hyper06	Mar 5, 2023	9	Positive	Fire Emblem Engage is an amazing game and the ...	0	4	0.000000
381	PMG-Writer	http://www.metacritic.com/user/PMG-Writer	Mar 18, 2023	9	Positive	Nintendo and the developers from Intelligent S...	0	0	NaN
382	shw079	http://www.metacritic.com/user/shw079	Mar 20, 2023	8	Positive	This is my first fire emblem game and it is ov...	0	0	NaN
383	Snoopy64	http://www.metacritic.com/user/Snoopy64	Jan 31, 2023	7	Mixed	The gameplay alone deserves a 10, but the othe...	0	0	NaN

Figura 4.5: Dataframe con las reseñas de usuarios de Fire Emblem Engage

La flexibilidad de este formato permite estructurar tanto los datos obtenidos de Metacritic como los de Twitter, pues pueden crearse dataframes importando archivos csv o json, pues dicha conversión se basa en simplemente usar los métodos *read_csv()* o *DataFrame()* de la librería pandas, respectivamente.

Volviendo a los propios datos extraídos de Metacritic, éstos se han organizado en tres carpetas, las cuales contienen por lo general cinco archivos, uno para cada juego y plataforma en que está disponible (pues también se incluyen las valoraciones de PC de Hi-Fi Rush y Forspoken).

- *overviews*. Resumen de todas las reseñas de cada juego. Puntuación media total y cantidad de reseñas, diferenciando si son positivas, negativas o de puntuación intermedia. Se incluyen dos entradas, una para el conjunto de críticas de los usuarios y otra para las de prensa especializada. Cabe destacar que los totales obtenidos no coincidirán con el número de reseñas presentes en las dos siguientes carpetas, pues tanto críticos como usuarios tienen la opción de valorar numéricamente el producto sin necesidad de proporcionar un comentario que acompañe a su valoración.
- *criticReviews*. Dentro de esta carpeta, se realiza una distinción entre las críticas que incluyen una valoración numérica (*scored*) de las que no lo incluyen (*unscored*). Para el caso de reseñas con puntuación, existen seis columnas de datos:
 - *source*. Nombre de la entidad autora de la reseña.
 - *link*. Link que lleva a la reseña original del juego en la página del autor, pues Metacritic únicamente incluye un resumen general de ésta.
 - *date*. Fecha en que se publicó la reseña.
 - *grade*. Nota otorgada al juego sobre 100.
 - *scoreType*. Tipo de calificación otorgada por Metacritic en base a la nota dada. Si es inferior a 50 se considera negativa, mientras que una nota igual a superior a 75 computa como positiva. Si la valoración se encuentra en el rango ubicado entre ambos valores, la crítica se califica como intermedia o mezclada. Este baremos solo es aplicable a críticas de videojuegos realizadas por prensa especializada, pues tanto las notas de críticos como de otros medios artísticas tienen rangos diferentes.
 - *text*. Texto de la reseña en sí.

Las críticas sin nota final siguen la misma estructura, aunque omitiendo las columnas *grade* y *scoreType*.

- *userReviews*. Conjunto de todas las críticas realizadas por los usuarios. Sigue una estructura análoga al de las críticas de prensa especializada, tal y como puede verse en 4.5.
 - *source*. Nombre del usuario que escribe la reseña.
 - *link*. Link al perfil del usuario.
 - *date*. Fecha en que se publicó la reseña.
 - *grade*. Nota otorgada al juego sobre 10.

- *scoreType*. Tipo de calificación otorgada por Metacritic en base a la nota dada. En este caso, se considera positiva si es superior a 7, negativa si es inferior a 5 e intermedia en cualquier otro supuesto.
- *text*. Texto de la reseña en sí que puede ser tan extensa como se deseé, al no ser simplemente un resumen de la misma.
- *upThumbs*. Número de usuarios que han considerado la reseña de utilidad.
- *totalThumbs*. Número total de usuarios que han opinado sobre la utilidad de la reseña.
- *helpfulness*. Ratio calculado en base al número de usuarios que han considerado la valoración útil con respecto al total de personas que han opinado (valor entre 0 y 1).

Además, tal y como se ve en el ya mencionado estudio de Pellarolo [52], debe tenerse en cuenta que las críticas de usuarios suelen dar una valoración menor que la que otorgan los críticos especializados (ver 3.4).

Capítulo 5

Preprocesado de datos

La recolección de datos es sólo el inicio dentro de cualquier proyecto de *data science*. Una vez guardados, es vital saber identificar qué datos son relevantes para el estudio, cuáles carecen de utilidad en el contexto del trabajo o incluso aquellos que pueden llegar a contaminar el desarrollo del mismo. Tras esta criba inicial, y dado que estamos ante un proyecto enfocado en el uso de herramientas PNL, resulta crucial hacer un preprocesado de los datos para que éstos adquieran un formato que facilite la manipulación de los mismos, así como la visualización de resultados finales.

Este objetivo final es lo que se conoce en PNL como corpus. Un **corpus** hace referencia a una colección de texto o audio autenticado, organizado como conjuntos de datos. En este contexto, el término *autenticado* indica que la fuente de dicha información es una persona que domina con cierta fluidez el idioma. Una vez obtenido, con el corpus pueden abordarse labores más complejas como el entrenamiento de IAs o modelos de *Machine Learning* [35].

La mayoría de estas tareas harán uso de **Natural Language Toolkit** o **NLTK** [13], un conjunto de librerías y programas de Python que permiten llevar a cabo una gran variedad de tareas relacionadas con PLN. Dichos recursos se desarrollan gracias a las más de 50 corpora (plural de corpus) a las que tiene acceso, algunas tan relevantes como **WordNet**, una amplia base de datos de léxico anglosajón.

Es por ello que este capítulo se centrará en quedarse únicamente con los datos que vayan a usarse a lo largo del estudio, dotándolos además del formato idóneo para cada una de las fases del procesado de información. Se visitarán los principales pasos de un proceso de limpieza de datos, enfocándonos en las particularidades específicas de este estudio y aplicándose para los datos recogidos tanto de Metacritic como de Twitter. Al querer realizarse un análisis de sentimientos y modelado de temas, el preprocesado se centrará únicamente en la información textual obtenida por ambas vías (el texto de reseñas y tweets).

5.1. Data cleaning

El primer paso para procesar datos consiste en la eliminación de todas aquellas entidades y caracteres no deseados. Para este trabajo, se identificaron los siguientes elementos a quitar:

- Caracteres no ASCII (caracteres de control no imprimibles)
- Hipervínculos y enlaces
- Etiquetas html
- Entidades de Twitter (hashtags, urls y nombres de usuarios)
- Signos de puntuación

Además de suprimir todos estos elementos, también se unificará la escritura de todas las palabras a minúsculas para evitar la redundancia de términos pues, en caso de no hacerse, las palabras "*Nintendo*" y "*nintendo*" podrían llegar a ser considerados como distintas a pesar de hacer ambas referencia a la compañía nipona. Este proceso se conoce como *case folding*.

El reconocimiento de todas estas entidades se ha realizado gracias al uso de la librería NLTK ya mencionada anteriormente, junto al empleo de **expresiones regulares** (secuencia de caracteres que especifica un patrón de coincidencia), las cuales pueden manipularse gracias al módulo de Python **re**.

5.2. Language filtering

Una vez realizado el procesado inicial de los datos, será necesario quedarse únicamente con los textos escritos en inglés. Si bien dicho cribado ya se realizó al hacer las peticiones de los tweets mediante el uso de filtros idiomáticos, para las reseñas de Metacritic no se produjo dicho filtrado, por lo que es obligatorio desechar aquellos textos que no estén escritos en el idioma deseado. La necesidad de este paso queda evidenciada al consultar, por ejemplo, las reseñas de Hi-Fi Rush para Xbox pues, tal y como se ve en la figura 5.1, aparte de las reseñas en inglés hay otras en portugués, turco e incluso chino.

La detección de idiomas se realiza gracias a la librería **langdetect**, que es una adaptación directa del mecanismo de detección de idioma implementado por Google para Java.

```
0      Amazing Game, one of the best surprises ever! ...
1      It's highly recommended and one of the best vi...
2      Jogo perfeito! Sua jogabilidade, gráficos e hi...
3      The good:\r\nAn interesting mix of Sunset Overdr...
4      Başında oturdun mu saatlerce kalkamıyorsun. Yi...
...
1419    Love it. Great gameplay. Incredibly immersive....
1420    Worst game, the worst game in my entre life pl...
1421    This game lives up to the hype. The world conc...
1422          我歌唱火焰, 在我的眼睛周围, 他们永远不会害怕, 就像敌人奔
向太阳
1423    Amazing game! I loved the art style, all plent...
Name: text, length: 1424, dtype: object
```

Figura 5.1: Reseñas de usuarios de Hi-Fi Rush en Xbox

5.3. Tokenization

La tokenización de texto o *text tokenization*, comúnmente abreviada como **tokenización**, es el proceso que consiste dividir el texto en unidades simples denominadas tokens. Para la mayoría de idiomas, dicha división se realiza a partir de los espacios en blanco o los signos de puntuación, aunque para lenguajes que no incluyen espacios, como el chino o el japonés, dicha partición requiere de una mayor complejidad. No obstante, al hacer una división tan rígida en unidades, se corre el riesgo de perder parte del significado del texto original.

Sin embargo, el proceso empleado en este apartado se ha simplificado bastante, pues la consideración de significados conjuntos se realiza en las secciones propias al modelado y análisis de los datos, tal y como se explica en secciones como 6.3. Además, al estar todos los textos analizados escritos en inglés, la separación en tokens puede realizarse simplemente por palabras.

La implementación práctica de este apartado se ha realizado gracias a la herramienta *TweetTokenizer* proporcionada por NLTK, diseñada especialmente para la tokenización de textos identificados como tweets. También se ha usado en el caso de las reseñas, al tratarse ambos supuestos de textos breves donde se manifiesta una idea u opinión. Otra alternativa para la tokenización sería la librería **gensim**, diseñada para el *topic modelling*, la indexación de documentos o la identificación de similitudes entre corpora; que a su vez incluye herramientas para la tokenización de textos [54].

5.4. Stop words

En 1959, Hans Peter Luhn proponía que las palabras más frecuentes en un texto no son las que mayor cantidad de información aportan. Esto ha sido avalado por diversos estudios posteriores como [36], donde se puede observar que los términos más relevantes son aquellos que aparecen con una frecuencia intermedia, y no las palabras que más aparecen o que casi no lo hacen. De hecho, los términos que más frecuentemente aparecen pueden incluso llegar a suprimirse de un texto sin que éste pierda su significado primordial.

En base a esta idea, es el propio Luns quien acuña primero la terminología **stop words**, **palabras vacías** en castellano, para hacer referencia a aquellos términos carentes de un significado propio -al menos no de forma aislada-, pero que se utilizan con mucha frecuencia en la mayoría de los idiomas. Algunas de estas palabras carente de contenido son artículos, pronombres, proposiciones o conjunciones.

NLTK proporciona un diccionario de palabras vacías en varios idiomas para poder eliminarlas fácilmente de cualquier corpus, simplificando la tarea de los científicos de datos.

5.5. Stemming y lemmatization

La **normalización de palabras** consiste en unificar las palabras o tokens en torno a un formato estándar común. Dicho proceso suele comenzar por realizar la fase de ***case folding*** ya realizada en el apartado 5.1, para después obtener el significado subyacente detrás de cada palabra, el cual puede recuperarse gracias a la raíz de cada palabra. La obtención de éstas puede realizarse de dos maneras diferentes:

- ***Stemming.*** Reducción de las palabras flexionadas o derivadas a su raíz. Así, las flexiones derivadas de variaciones de género y número pueden unificarse bajo la raíz común que comparten, únicamente eliminando la desinencia correspondiente. Este procedimiento tan crudo puede desembocar en errores en el texto normalizado final, como significados incorrectos o errores ortográficos, tal y como se aprecia en la figura 5.1.
- ***Lemmatization.*** Refinación del método previo, consistente en agrupar las formas flexionadas de una palabra para que puedan analizarse como un único elemento, gracias a la forma diccionario de cada palabra.

Por lo tanto, *stemming* es un proceso simple de truncado (y en ocasiones sustitución por una desinencia general) para la obtención de la raíz de las palabras; mientras que la lematización conlleva el uso de métodos más sofisticados de análisis morfológico completo de la palabra, distinguiendo si a la raíz se ha añadido una desinencia o un afijo (morfema que sólo modifica gramaticalmente la raíz o aporta algún significado adicional). Ambos procedimientos pueden realizarse mediante funcionalidades de NLTK, como el *SnowballStemmer* para el *stemming* o *WordNetLemmatizer* para la lematización.

```

1 # Ejemplo de errores que da stemming
2 print('Texto preprocesado y tokenizado: ', tweets['Nintendo'].
3     finalText[36833])
4 print('Texto stemmizado: ', tweets['Nintendo'].stemmedText[36833])
5 print('Texto lematizado: ', tweets['Nintendo'].lemmatizedText
6     [36833])

Texto preprocesado y tokenizado: february means definitely
                                official nintendo direct watch 2019 february 13th 2020 march 26
                                th 2021 february 17th 2022 february 9th nothing ever 100
                                guaranteed odds good
Texto stemmizado: februari mean definit offici nintendo direct
                                watch 2019 februari 13th 2020 march 26th 2021 februari 17th 2022
                                februari 9th noth ever 100 guarante odd good
Texto lematizado: february mean definitely official nintendo
                                direct watch 2019 february 13th 2020 march 26th 2021 february 17
                                th 2022 february 9th nothing ever 100 guaranteed odds good

```

Código 5.1: Diferencia entre *stemming* y *lemmatization*

Capítulo 6

Topic modelling

El **topic modelling** o modelado de temas es una técnica de aprendizaje automático, de la rama del procesamiento del lenguaje natural, que trata de asignar temas (*topics*) a un conjunto de textos. En concreto, hace referencia a los modelos estadísticos que permiten descubrir los temas subyacentes en colecciones de textos de cualquier índole.

Los modelos para la identificación de temas son esencialmente algoritmos iterativos que trabajan con **matrices de características de documentos** o *document feature matrices* (matrices que describen la frecuencia con la que aparecen términos a lo largo de una serie de documentos), para así agruparlos en base a sus elementos comunes. Si bien las matrices suelen recoger simplemente la frecuencia de aparición de cada palabra, también pueden hacer referencia a sustantivos o entidades como nombre. Un ejemplo simple de aplicación sería una colección de documentos donde aparecen palabras como "*partido*", "*equipo*" o "*marcar*", que podrían agruparse bajo un único tema denominado "*deporte*"; mientras que otros términos como "*caso*", "*ley*" o "*crimen*", que aparecen también en los documentos, se aglutarían bajo la temática de "*legalidad*" [17]. La figura 6.1 ilustra el caso descrito considerando los términos ingleses análogos.

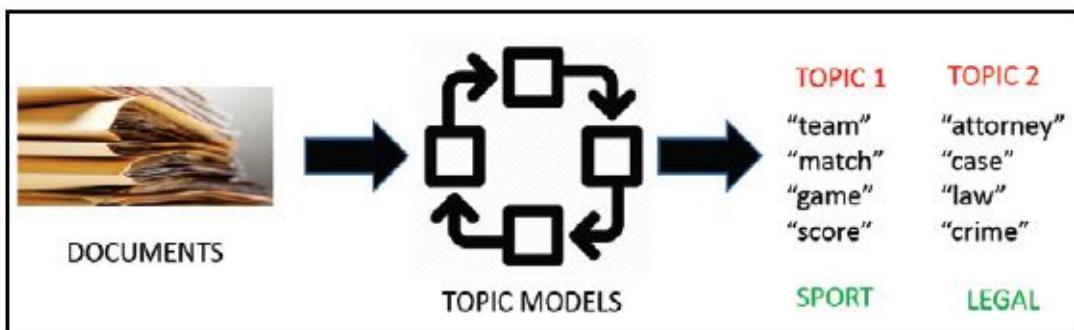


Figura 6.1: Esquema básico de funcionamiento de un proceso de *topic modelling*

A lo largo de este capítulo, se realizará un análisis preliminar de los datos antes de aplicar modelos para la extracción de temas. En concreto, se aplicarán dos modelos concretos de *topic modelling*: Latent Dirichlet Allocation y Biterm Topic Modelling.

6.1. Exploratory Data Analysis

En los capítulos previos se ha realizado un análisis de la estructura de los datos obtenidos, así como un procesado de los mismos. No obstante, el contenido como tal de éstos casi no ha sido examinado. Si bien la figura 2.1 explicaba los pasos básicos de los que consta un proyecto de *data science*, en un proyecto real no se sigue una metodología lineal, si no que al acabar cada fase del desarrollo se emplean los nuevos resultados obtenidos para mejorar los pasos previos. Por lo tanto, una aproximación más verídica de estos pasos sería la que muestra la figura 6.2.

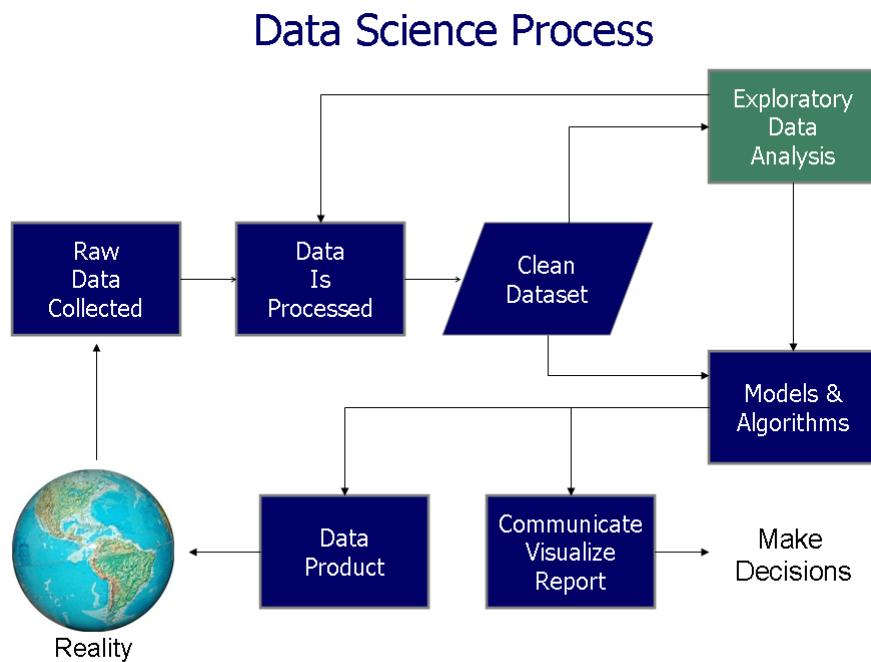


Figura 6.2: Esquema real del desarrollo de un proyecto de *data science*

Exploratory Data Analysis (EDA) o análisis exploratorio de datos hace referencia al proceso clave de realizar investigaciones iniciales sobre los datos para descubrir patrones, detectar anomalías, probar hipótesis y comprobar suposiciones con la ayuda de estadísticas resumidas y representaciones gráficas [51]. Esto permite determinar la mejor manera de manipular los datos de los que se dispone para así obtener las respuestas buscadas, así como corroborar la adecuación de las herramientas ya utilizadas.

Desarrolladas originalmente por el matemático estadounidense John Tukey en su libro homónimo publicado en 1977, hoy en día las estrategias EDA pueden clasificarse en cuatro categorías en función del tipo de datos (univariante o multivariante) considerados y la representación que se hace de los mismos (gráfica o no) [75].

- **Univariante no gráfica.** Descripción básica de los datos para encontrar los patrones que existen en ellos. Al ser una sola variable, no es necesario estudiar ni causas ni relaciones con otras variables.
 - **Gráfico univariante.** Variante del método anterior que busca aportar una visión más clara de los datos gracias al uso de histogramas o diagramas de caja.
 - **Multivariante no gráfica.** En este caso sí resulta necesario estudiar la dependencia y correlación de las variables. La manera más rápida de hacerlo es mediante tabulación cruzada o estadísticas como el coeficiente de correlación.
 - **Gráfico multivariante.** La forma más efectiva de visualizar la relación existente entre varias variables es mediante el uso de gráficas como mapas de calor o gráficos de burbujas.

6.1.1. WordCloud

Una ***Word Cloud*** o nube de palabras es una manera de representar visualmente un texto a partir de los términos que aparecen en él con mayor frecuencia, aumentando su tamaño en función de si se repiten de manera reiterada a lo largo del texto. Por lo tanto, se encuadraría como un método de EDA gráfico univariante. De este modo, resulta una manera muy visual para comenzar un proceso de *topic modelling*, ya que permite identificar aquellos temas que aparecen con mayor frecuencia a lo largo del texto.

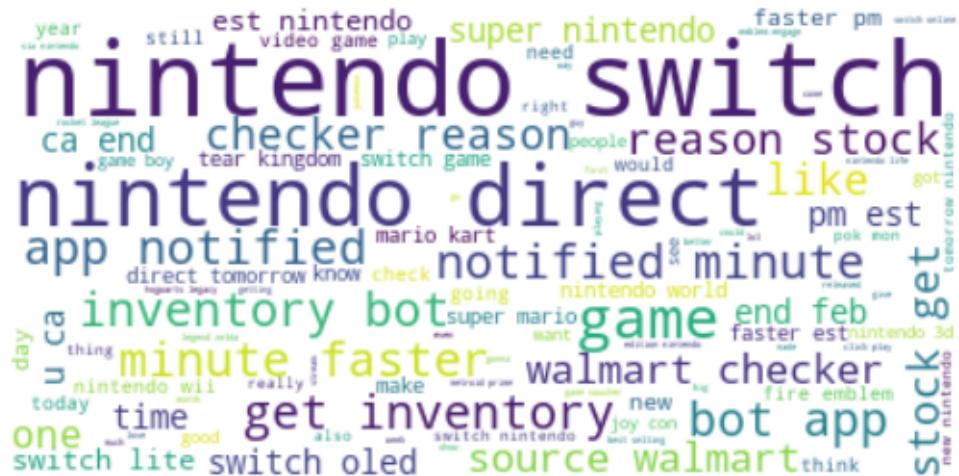


Figura 6.3: *Word cloud* de los tweets de Nintendo

Para este proyecto, se ha utilizado la implementación en Python **WordCloud** presente en [45], que permite generar de manera intuitiva nubes de palabras dando opciones extra de personalización como limitar el máximo número de términos que aparecen o especificar el conjunto de palabras vacías que no deben considerarse.

[74] supone una gran introducción para lograr manejar todas las herramientas y métodos de los que dispone la librería. Por ejemplo, una funcionalidad clave es la posibilidad de decidir si pueden considerarse conjuntos de dos términos en vez de que todos sean tenidos en cuenta de manera individual. Esto resulta clave para identificar términos como los nombres de las consolas, tal y como se ve con el ejemplo de Nintendo Switch de la figura 6.3. Esta idea es la base del modelo Biterm desarrollado en 6.3, y puede llegar a ampliarse a incluso más términos pues, por ejemplo, Fire Emblem Engage son tres palabras y el modelo únicamente identifica dos de éstos.

6.1.2. Resultados

Al generar las nubes de palabras, se detectaron errores producidos durante el proceso de limpieza, como que no se descartaban algunas palabras vacías como el sujeto de las frases, o que los verbos apostrofados se mantenían como tal. Esto obligó a volver al desarrollo de los procesos de procesado de datos hasta obtener unos resultados válidos, lo cual ejemplifica el carácter circular del desarrollo del proyecto ya comentado en la figura 6.2.

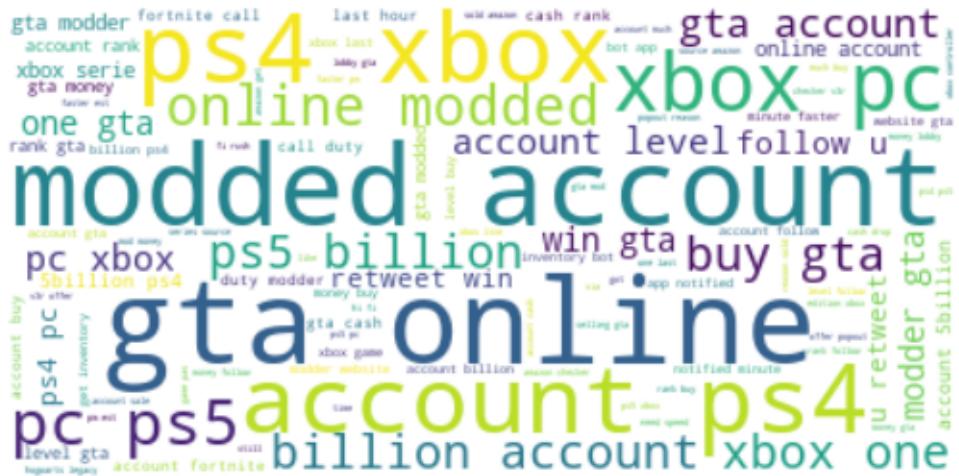


Figura 6.4: *Word cloud* de los tweets de Xbox

Con respecto a los propios *word clouds*, este primer análisis permite identificar algunos elementos que se verán de manera más clara en fases posteriores del proceso de análisis:

- **Nintendo.** Al considerar los tweets que mencionan a la compañía japonesa, se aprecian en la figura 6.3 algunos temas recurrentes como Nintendo Direct, evento que organiza la compañía cada cierto tiempo para presentar sus novedades y suele anunciar poco antes de realizarlo; o Walmart, que acababa de desmentir una serie de rumores que hablaban sobre un posible lanzamiento inminente de Advance Wars, otra saga de videojuegos de Nintendo [23].

- **Xbox.** Entre los tweets que mencionan a la compañía norteamericana, se observa en la figura 6.4 una predominancia de términos como gta o cuentas modificadas. Esto se debe a que la compañía Rockstar acababa de publicar un parche para la versión de PC de su juego GTA online, debido a las vulnerabilidades que presentaba el software y permitía a hackers modificar las estadísticas de otros jugadores, provocando que el propio juego expulsase a usuarios que habían sido objeto de ataque por parte de los hackers al ser detectados como que habían modificado de manera ilegítima sus estadísticas [55]. A pesar de que esta modificación sólo se dio en PC, la aparición masiva de estos términos al realizar la búsqueda sobre Xbox, muestra la estrecha relación que ha establecido Microsoft entre ambos mercados.
 - **Forspoken.** Si bien las reseñas pueden no resultar un objetivo tan prioritario para esta clase de análisis como los tweets al tener un enfoque más específico y concreto, se pueden obtener de todas formas algunas primera ideas como en el caso de la figura 6.5. Tal y como se podía esperar, se mencionan temas que hablan del carácter del propio juego como *mundo abierto*, *magia* o *combate*, aunque aparecen términos como *malo* o *aburrido*, que presagian los resultados que se obtendrán al realizar el análisis de sentimientos de la sección 7.



Figura 6.5: *Word cloud* de las reseñas de *Forspoken* en PS5

Únicamente se han comentado algunos de los *word cloud* generados, destacando los de tweets respecto a los de reseñas, al tener éstos mayor cantidad de datos que escrutar. Para poder ver y analizar todos ellos en profundidad, basta referirse al cuaderno de Jupyter correspondiente.

6.2. Latent Dirichlet Allocation

LDA es un modelo probabilístico generativo de un corpus. La idea fundamental detrás de esto es que los documentos contienen diversos temas subyacentes, los cuales están caracterizados por la distribución respecto a las palabras existentes. En esencia, podríamos definirlo como un modelo no supervisado dentro de la rama del Aprendizaje Automático que permite inducir los términos relativos a grandes conjuntos de texto. Para poder entender su funcionamiento, será necesario analizar el funcionamiento general de los modelos de lenguaje usados en PNL, los cuales suelen ser bayesianos (ver sección 7.2.1).

6.2.1. N-gramas

Si se considera la frase "*Voy a aparcar mi coche en*", lo habitual es suponer que después de esa información vendrán estructuras tales como "*el garaje*" o "*el parking*", en vez de otras como "*la cocina*" o "*paraguas*". A pesar de que esto puede resultar obvio, motiva el estudio de la probabilidad de aparición de términos en PNL. En cualquier tarea de Procesamiento de Lenguaje Natural, el cálculo de la probabilidad de aparición de una palabra resulta determinante, bien sea para identificar si ha habido un error a la hora de transcribir un discurso hablado o bien si se quiere realizar un proceso de traducción.

Los modelos que asignan probabilidades a secuencias de palabras se denominan **modelos de lenguaje**. El más simple de todos estos es el **N-grama**, que se basa en la aplicación de la **regla de la cadena de la probabilidad**¹, en la cual basta sustituir las variables aleatorias por las palabras que conforman una frase o documento:

$$P(w_{1:n}) = P(w_1) \cdot P(w_2|w_1) \cdot \dots \cdot P(w_n|w_{1:n-1}) = \prod_{k=1}^n P(w_k|w_{1:k-1})$$

considerando que estamos evaluando un documento, que en esencia es una colección de palabras ordenadas como una secuencia; $d = (w_1, w_2, \dots, w_n)$ y $w_{1:k}$ es la secuencia de las k primeras palabras. Esta expresión muestra la relación entre la probabilidad conjunta de la secuencia de palabras con respecto a la probabilidad condicional de que aparezcan las palabras previas. No obstante, al no saber tampoco esas probabilidades condicionales previas, lo más habitual es recurrir a métodos de aproximación. En concreto, se suele considerar la propiedad de Markov como válida, para así trabajar con modelos de Markov.

Un **proceso de Markov** hace referencia a modelos probabilísticos en los que la probabilidad de un evento futuro depende sólo de los inmediatamente previos a él. Un ejemplo son las cadenas de Markov, en las cuales la probabilidad del siguiente evento depende de manera exclusiva del acaecido inmediatamente antes, por lo cual permitiría estimar 2-gramas (o **bigramas**) sabiendo la probabilidad de aparición de uno de ellos. Para el caso de N-gramas, el cálculo de la probabilidad condicionada se basa en las $n - 1$ anteriores:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-N+1:n-1})$$

¹El cálculo de probabilidades se realiza siempre en formato logarítmico para evitar problemas de *underflow* al estar multiplicando números entre 0 y 1.

A parte de esta estimación, el cálculo de las probabilidades condicionadas se realiza a partir de **métodos de estimación de máxima verosimilitud**. Para el caso que nos ocupa, podemos simplemente considerar un documento extenso que sirviese como corpus y estimar en base al número de apariciones de esas estructuras, normalizándolo en un rango entre 0 y 1:

$$P(w_n | w_{n-N+1:n-1}) = \frac{C(w_{n-N+1:n-1} w_n)}{C(w_{n-N+1:n-1})}$$

siendo $C(w_{n-N+1:n-1} w_n)$ el número de veces que aparece el N-grama en cuestión buscado. Este ratio se conoce también como **frecuencia relativa**.

6.2.2. TF-IDF

No obstante, estos términos no son variables aleatorias como tal, si no que son palabras con significados asociados que condicionan su propia probabilidad de aparición según el contexto. Para estudiar esto existe la semántica de vectores o ***vector semantics***, que es la manera estándar para representar el significado de las palabras en PNL a través de su contexto. En términos matemáticos, la idea subyacente consiste en representar cada palabra como un punto en un espacio multidimensional que podríamos catalogar de semántico, al obtenerse de las distribuciones de las palabras colindantes. Los vectores para representar palabras se denominan **embebimientos**, cuya idea es similar al de su análogo matemático. Un ejemplo sencillo de embebimiento es el de la figura 6.6, que agrupa los términos ingleses en función de si su connotación es positiva, negativa o neutra. Estos métodos son especialmente útiles para la detección de temas o el análisis de sentimientos.



Figura 6.6: Ejemplo básico de embebimiento para la detección de significados similares

Estos modelos basados en distribuciones emplean como base una matriz de coocurrencias, que permiten representar con qué frecuencia los términos aparecen ligados. Para un documento, esto se representa mediante la ***term-document matrix*** o matriz término-dокументo, en la que cada fila representa una palabra y cada columna un documento del corpus, considerando éste como una colección de M documentos; $D = \{d_1, d_2, \dots, d_M\}$.

Tal y como se comentó en la sección 5.4, los términos que más información aportan son aquellos que aparecen con una frecuencia intermedia [36]. Para poder dilucidar cuáles son estos términos, se emplea el **algoritmo tf-idf**, el cual se basa en el producto de dos términos:

- **Term frequency (tf)**. Frecuencia de aparición de una palabra w a lo largo de un documento d ; $\text{tf}_{w,d} = C(w, d)$.²
- **Inverse document frequency (idf)**. Factor que da mayor peso a los términos que únicamente aparecen en unos pocos documentos. Para ello, es necesario saber la frecuencia de un documento de una palabra w o **document frequency** (df_w), que indica en cuántos documentos del corpus aparece. Gracias a esto, el peso se calcula simplemente como el ratio $\text{idf}_w = \frac{M}{\text{df}_w}$.³

Por lo tanto, esto permite calcular la relevancia de cada término como el siguiente producto, que asigna un peso/importancia entre 0 y 1 a cada palabra de un documento:

$$\mathbf{w}_{w,d} = \text{tf}_{w,d} \times \text{idf}_w$$

El principal inconveniente de esta herramienta es que casi no proporciona una reducción dimensional de los términos analizados y revela poco de las relaciones subyacentes entre los documentos. Para solventar estos problemas surge el análisis semántico latente o **Latent Semantic Analysis/Indexing** (LSA/LSI), que en esencia consiste en aplicar una descomposición en valores singulares sobre la matriz final resultante.

6.2.3. LDA

Si bien LSA da unos resultados aceptables, en los últimos años ha surgido un nuevo modelo que ofrece resultados más certeros para la identificación de temas, a la par que ha resultado ser de mayor utilidad para realizar labores de agrupamiento sobre conjuntos de datos de gran dimensionalidad: Latent Dirichlet Allocation.

Latent Dirichlet Allocation (LDA) es un modelo probabilístico generativo, por lo que retoma el uso de N-gramas para el cálculo de probabilidades. Para ello, LDA asume el siguiente proceso generativo para cada documento d_i en un corpus D :

1. Definir $N \sim \text{Poisson}(\zeta)$.
2. Definir $\theta \sim \text{Dir}(\alpha)$.
3. Para cada una de las N palabras w_n :
 - a) Elegir un tema $z_n \sim \text{Multinomial}(\theta)$.
 - b) Elegir una palabra w_n de $P(w_n|z_n, \beta)$, una probabilidad condicionada multinomial en función del tema z_n escogido.

²Otras alternativas para este cálculo se basan en mitigar el efecto de palabras que aparezcan múltiples veces tomando el logaritmo decimal; $\text{tf}_{w,d} = \log_{10}(C(w, d) + 1)$.

³También suele mitigarse mediante el uso del logaritmo decimal; $\text{idf}_w = \log_{10}(\frac{M}{\text{df}_w})$.

donde la probabilidad de las palabras se establece gracias a una matriz $\beta \in \mathcal{M}_{k \times V}$ siendo $\beta_{ij} = P(w_j|z_i)$, suponiendo que el conjunto de palabras se extrae de un vocabulario de tamaño V y k es la dimensión de la distribución de Dirichlet, la cual determina el número total de temas posibles y tiene densidad de probabilidad:

$$P(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \dots \theta_k^{\alpha_k-1}$$

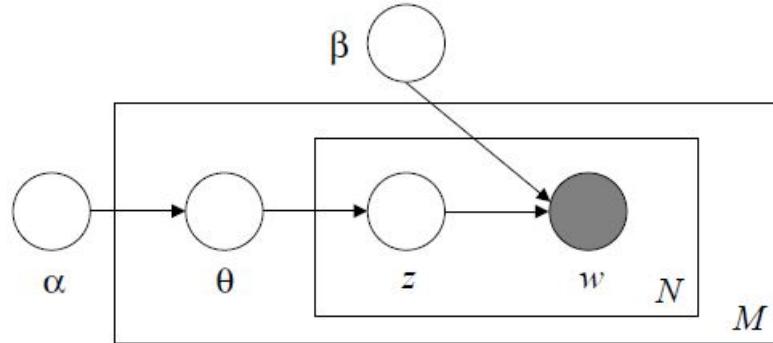


Figura 6.7: Representación gráfica de la generación a partir de LDA. El primer cuadro indica que el proceso se realiza sobre el conjunto de documentos disponible, mientras que el segundo hace lo propio para cada documento (elección de temas y palabras)

En términos algo más simples, lo que se hace es asumir distribuciones de Poisson y Dirichlet sobre los modelos para después crear los miembros de cada grupo maximizando la probabilidad de que una nueva palabra esté en función de los componentes actuales del grupo. La clave de LDA reside en considerar los temas como variables aleatorias **intercambiables**⁴, lo cual permite dar la probabilidad de un conjunto de palabras y temas como:

$$P(\mathbf{w}, \mathbf{z}) = \int P(\theta) \left(\prod_{n=1}^N P(z_n|\theta) P(w_n|z_n) \right) d\theta$$

Para obtener la distribución de palabras, basta considerar la marginal, obteniendo la distribución marginal de cada documento como una mezcla de distribuciones continuas ponderadas:

$$P(\mathbf{w}|\alpha, \beta) = \int P(\theta|\alpha) \left(\prod_{n=1}^N P(w_n|\theta, \beta) \right) d\theta$$

⁴Un conjunto de variables aleatorias $\{z_1, \dots, z_N\}$ se denomina **intercambiable** si su distribución conjunta es invariantes a permutaciones; $p(z_1, \dots, z_N) = p(z_{\pi(1)}, \dots, z_{\pi(N)})$. Para conjuntos infinitos, la propiedad es análoga si se cumple para todo subconjunto finito de variables aleatorias.

6.2.4. Implementación y resultados

Una vez visto cómo funciona de manera teórica LDA, podemos comenzar a usarlo para extraer temas de los datos procesados. Para ello se hará uso de **gensim**, ya mencionada en la sección 5.3, una librería de código abierto que contiene una multitud de herramientas para el modelado temático no supervisado y otras funcionalidades PNL. En concreto, incluye un modelo de LDA llamado **LdaMulticore** para la extracción de temas y cuyos resultados pueden visualizarse fácilmente como cuadernos interactivos, que también pueden almacenarse como archivos html. El modelo requiere que se especifiquen los siguientes parámetros:

- **id2word.** Relación entre todas las palabras e IDs asignados a ellas. Esto permite definir el diccionario con el que se trabajará.
- **corpus.** Corpus de todos los documentos convertido ya en matriz de frecuencias. Este último paso se realiza gracias al método **doc2bow**, que convierte documentos, los cuales se suponen que están ya tokenizados y normalizados, en tuplas de id y número de apariciones del token.
- **num_topics.** Número de temas que se extraerán. Lo ideal sería hacer una optimización del número de temas a extraer, pero debido a que LDA es un modelo más enfocado al análisis de documentos extensos en lugar de otros cortos como los tweets o las reseñas, consideraremos que el número de temas para todos los modelos es 10.

Una vez entrenado un modelo, puede visualizarse su contenido gracias a **pyLDAvis**, que ofrece un formato interactivo para comprender de manera sencilla los resultados obtenidos mediante LDA. El cuaderno se compone de dos partes interrelacionadas, tal y como se aprecia en la figura 6.8:

- **Mapa de distancia entre temas.** Gráfico de burbujas que representa los diferentes temas hallados en el texto, representando la diferencia entre unos y otros gracias a la distancia calculada usando escalado multidimensional (MDS). Permite seleccionar temas específicos, aunque todos se agrupan bajo uno global con numeración 0.
- **Términos más relevantes.** Gráfico de barras con los términos más relevantes, los cuales se deciden en base a las métricas de relevancia indicadas y que pueden ajustarse usando el parámetro λ . Al seleccionar un tema concreto, se puede ver la frecuencia específica de los términos en la burbuja en cuestión. Además, al seleccionar un término determinado, el gráfico de burbujas se modifica para ajustar el tamaño de las mismas según dónde aparece más el token. Un ejemplo de esto puede verse en la figura 6.9, en la cual se ha seleccionado el tema 4 así como el término *walmart*.

6.2. Latent Dirichlet Allocation

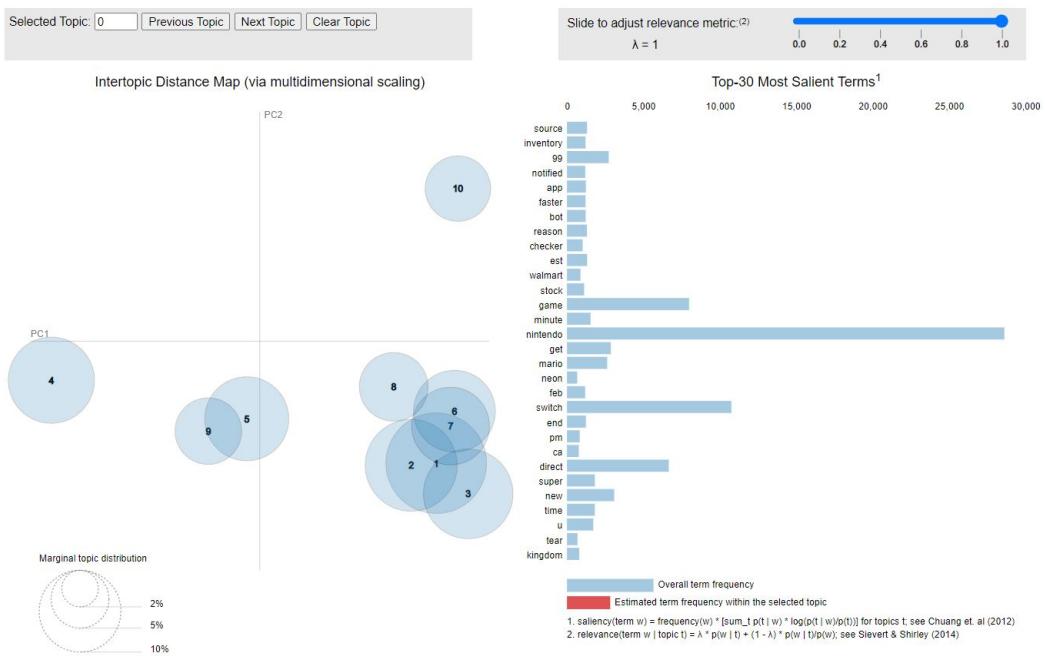


Figura 6.8: Visualización del modelo entrenado con tweets sobre Nintendo, considerando 10 temas

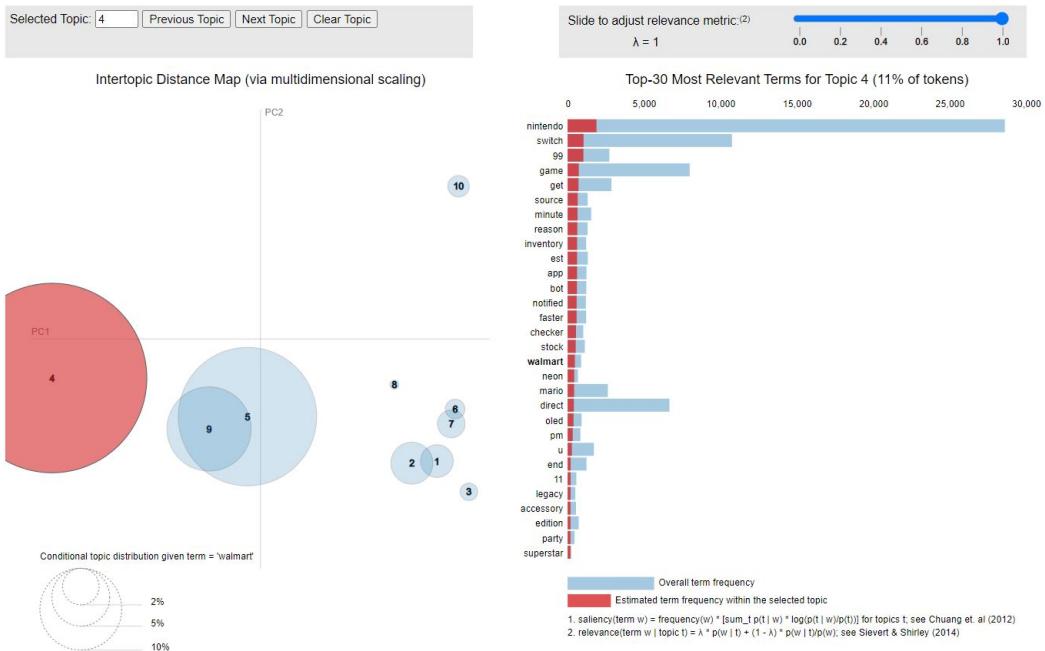


Figura 6.9: Modelo de Nintendo resaltando el cuarto tema y el término *walmart*

Gracias a esto, podemos descubrir algunos de los temas candentes en cada uno de nuestros conjuntos de datos, aunque al haber una gran cantidad de ellos, aquí sólo haremos algunas apreciaciones de éstos que pueden extraerse tras escrutar detenidamente los resultados obtenidos. Con respecto a los **tweets**, analicemos cuáles son los asuntos más comentados para cada una de las empresas:

- **Nintendo.** Tal y como ya se observaba en la nube de palabras generadas, son recurrentes los tweets relativos al Nintendo Direct, los cuales se ubican en el cuadrante inferior del gráfico de burbujas; así como las reacciones al anuncio de Walmart Canadá, las cuales se encuadran dentro del cuadrante inferior izquierdo (figura 6.9).
- **PlayStation.** Consultando los modelos se aprecian dos grandes temas: Playstation Plus, que ofrecía diferentes ventajas de contenido para sus suscriptores en juegos como Call of Duty Modern Warfare 2.0, Fortnite, ó Apex Legends; y *broadcast*, debido a tweets de jugadores que anuncian que están jugando en vivo en plataformas como Twitch.
- **Xbox.** El principal punto a destacar se pudo observar ya durante el análisis exploratorio de datos. El tema predominante comentado por la comunidad es la actualización de GTA que impedía a los hackers seguir modificando cuentas. Si bien ya se distinguía su presencia en la nube de palabras generadas, la agrupación de todos estos términos en un único tema es particularmente notoria en el gráfico de burbujas generado por LDA, tal y como puede verse en la figura 6.11. Además, también se ve que otros temas bastante comentados son juegos multijugador online en boga tales como Fortnite o Call of Duty, al igual que ocurría en PlayStation.

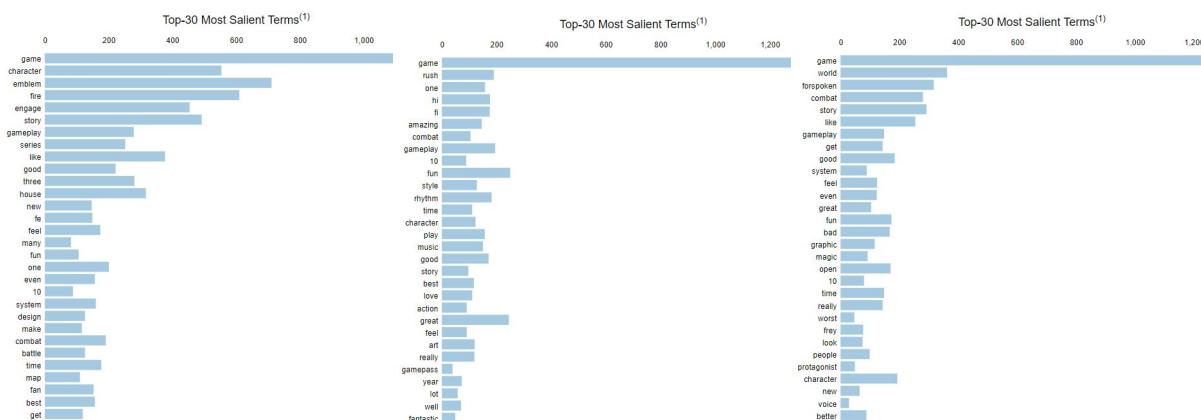


Figura 6.10: Términos más relevantes en las reseñas de cada videojuego en las consolas de cada compañía (FE Engage para Switch, Hi-Fi Rush para Xbox Series X y Forspoken para PS5; respectivamente)

En cuanto a las **reseñas** de los diferentes juegos, al ser textos más enfocados y sin tanta variabilidad con respecto a la temática que pueden abordar (pues todos ellos son valoraciones de un mismo producto), se pondrá el foco en los términos más relevantes, para ver si los resultados son coherentes con los del análisis exploratorio de los datos. Como ejemplo, veamos únicamente las reseñas de cada producto relativas a la consola perteneciente a cada una de las compañías. Tal y como se ve en la figura 6.10, tanto **Fire Emblem Engage** para **Nintendo Switch** como **Hi-Fi Rush** para **Xbox Series X** tienen una gran cantidad de términos positivos asociados como *best*, *good* o *fantastic*; hablándose en el primero del combate (*combat*) o del diseño (*design*), mientras que en el segundo destacan términos como *rhythm* o *music*. Por su parte, **Forspoken** para **PS5** también tiene asociados algunos de estos términos positivos, aunque hacen presencia otros como *bad* o *worst*, lo cual parece apuntar a cierta recepción negativa tal y como ya se dejaba entrever en la nube de palabras creada.

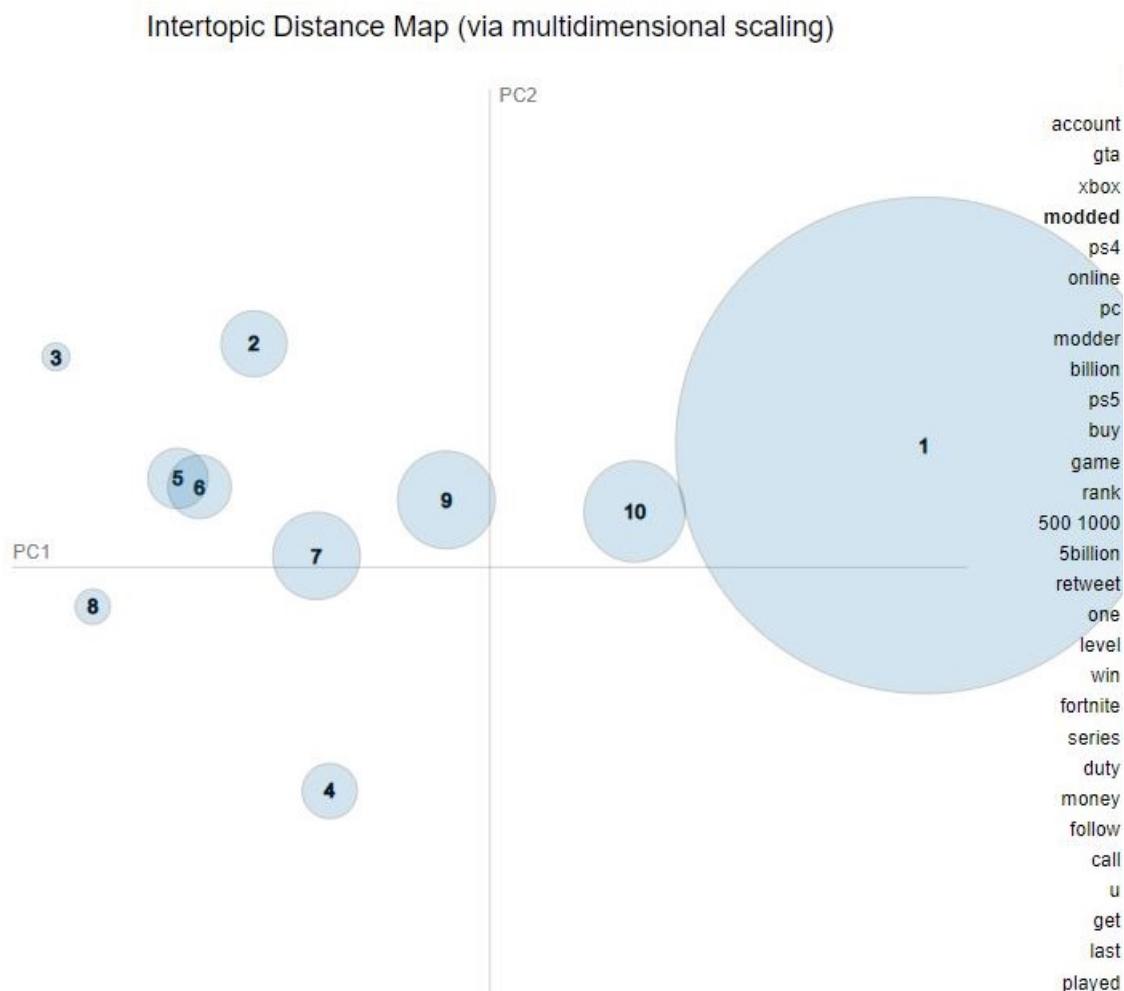


Figura 6.11: Gráfico de burbuja relativo a Xbox al resaltar el término *modded*

6.3. Biterm Topic Model

Si bien los resultados obtenidos hasta ahora han resultado favorables, sí que es cierto que los modelos clásicos ya vistos como LDA tienen algunos problemas de funcionamiento con textos cortos, como son los tweets o las reseñas. La razón fundamental radica en que los modelos temáticos convencionales capturan de manera implícita los patrones de coocurrencia de palabras a nivel de documento para revelar los temas, por lo que se ven afectados por la escasez de datos en documentos cortos. Para poner fin a esta problemática surge el modelo conocido como Biterm, diseñado específicamente para el modelado de temas en textos cortos [77].

Biterm Topic Model (BTM) utiliza los patrones agregados de todo el corpus para aprender temas y resolver el problema de la escasez de patrones de coocurrencia de palabras a nivel de documento que tienen los modelos convencionales. La idea detrás de esto es análoga a la de los 2-gramas ya vistos, con la única diferencia de que se consideran pares de palabras **no ordenados** que coocurran. El proceso de generación de un corpus de BTM es similar al de LDA, aunque únicamente recurriendo a distribuciones de Dirichlet:

1. Para cada tema z , escoger una distribución de palabras por temas $\varphi_z \sim \text{Dir}(\beta)$.
2. Elegir una distribución de temas $\theta \sim \text{Dir}(\alpha)$ para toda la colección.
3. Para cada biterm b del conjunto de biterms B :
 - a) Elegir un tema $z \sim \text{Multinomial}(\theta)$.
 - b) Escoger dos palabras $w_i, w_j \sim \text{Multinomial}(\varphi_z)$.

Vemos que el conjunto de biterms B funciona a efectos prácticos como el diccionario V que se usa en el modelo LDA, aunque aquí sus elementos son tuplas $b = (w_i, w_j)$, calculándose la probabilidad conjunta de cada biterm como:

$$P(b) = \sum_z P(z)P(w_i|z)P(w_j|z) = \sum_z \theta_z \varphi_{i|z} \varphi_{j|z}$$

En base a esto, la extracción de temas de un documento se basa en el uso de la siguiente probabilidad:

$$\begin{aligned} P(z|d) &= \sum_b P(z|b)P(b|d) = \\ &= \sum_b \frac{P(z)P(w_i|z)P(w_j|z)}{\sum_z P(z)P(w_i|z)P(w_j|z)} \frac{C(b, d)}{\sum_b C(b, d)} = \sum_b \frac{\theta_z \varphi_{i|z} \varphi_{j|z}}{\sum_z \theta_z \varphi_{i|z} \varphi_{j|z}} \frac{C(b, d)}{\sum_b C(b, d)} \end{aligned}$$

cuyo cálculo puede realizarse gracias al uso de la regla de la probabilidad condicional y la estimación a partir de frecuencias relativas (de manera análoga al caso visto para N-gramas), siendo $C(b, d)$ la frecuencia de aparición del biterm b en el documento d .

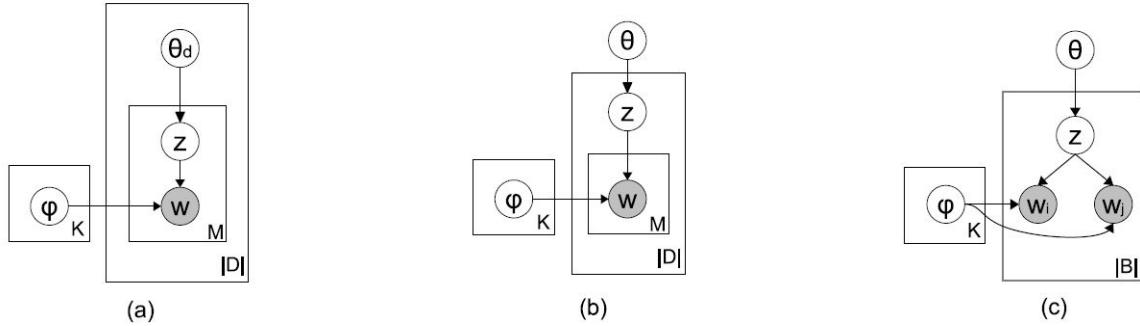


Figura 6.12: Representación gráfica de (a) LDA, (b) mezcla de unigramas y (c) BTM. Por claridad, no se representan los hiperparámetros fijados α y β .

A diferencia del LDA y la mezcla de unigramas, BTM modela el procedimiento de generación de biterms en una colección, en lugar de documentos. Esta comparativa entre los modelos clásicos y BTM se pone de manifiesto gracias a la figura 6.12. **LDA** genera primero una distribución de temas a nivel de documento para poder escoger entre éstos al considerar cada palabra del documento y realizar dicha asignación, lo cual provoca una alta dependencia de los términos presentes en los documentos y provoca dificultades para textos cortos, en los cuales la escasez de información puede dificultar este aprendizaje de temas. Por su parte, la **mezcla de unigramas** soluciona parcialmente este problema al generar la distribución de temas al nivel del corpus, aunque toma como base que todas las palabras de un mismo documento tratan el mismo tema, lo cual supone una reducción simplista ya que hasta en textos escuetos puede abordarse más de un tema. **BTM** solventa esto al dividir los documentos en biterms, conservando la correlación entre las palabras así como los diferentes temas que pueda haber dentro de un mismo documento.

Para su **implementación**, y dado que el trabajo original [77] únicamente da un serie de scripts para trabajar, se optó por tomar una librería que adaptese dicho modelo para su uso en Python. En un primer momento se intentó usar la implementación **bitemr** datada de 2019 [68], debido a las similitudes de ésta con la usada para LDA, pues incluso los resultados finales se muestran como un cuaderno pyLDAvis. No obstante, al tratar de instalarla se dieron varios problemas durante la compilación de los paquetes, por lo que se decidió cambiar de librería a una más reciente (a fecha de 2021): **bitemrplus**.

6.3.1. Entrenamiento del modelo y métricas

Bitemrplus es una versión *cythonizada*, de la implementación de [77] que también es capaz de calcular métricas como la perplejidad o la coherencia semántica [63]. Así, para cada uno de los temas que se va a analizar es necesario crear un modelo específico y entrenarlo. Al crear un modelo usando esta librería, deben definirse al menos los siguientes parámetros:

- **X.** Matriz de frecuencias de términos y documentos (*term-document matrix*).
- **vocabulary.** Vocabulario empleado en el corpus dado, en formato de lista de palabras
- **seed.** Semilla aleatoria utilizada para la creación del modelo.
- **T.** Número de temas que extraerán del modelo.
- **M.** Número de palabras más usadas para el cálculo de la coherencia.

Una vez generado el modelo específico, debe entrenarse para que se adecúe al conjunto de datos a analizar. Para ello, basta especificar el número de **iteraciones** (*iterations*, cuántas veces se actualizarán los parámetros del algoritmo) y los biterms con los que se entrenará, los cuales se pasan como una lista generada a partir del método **get_biterms**, el cual lo proporciona a través de una lista que contenga todos los documentos del corpus vectorizados.

Como se ha podido observar, para la creación del modelo es necesario especificar de antemano el número de temas que hay en todo el corpus. No obstante, en la mayoría de los casos, un proceso de extracción de temas se realiza al no conocerse el número de tópicos presentes en todo el corpus. Es por ello que, a la hora de fijar este parámetro, debe realizarse primero una estimación del número ideal de temas a extraer. Para ello, basta generar modelos y entrenarlos variando únicamente el número de temas considerados y acabar escogiendo el que presente menor entropía. La **entropía** puede definirse como una medida de la falta de información existente [11]. Por lo tanto, y siguiendo el principio de máxima entropía, al considerar la entropía como información negativa lo ideal será minimizar esta métrica. Es por ello que la búsqueda del número ideal de temas concluirá con la cantidad que de el menor valor de entropía. En concreto, *bitermplus* realiza este cálculo usando la entropía de Renyi [37].

La medida de **perplejidad** (*perplexity*) permite cuantificar el grado de precisión con que un modelo probabilístico predice una muestra. En PNL, se corresponde con el grado de incertidumbre que tiene un modelo al predecir un texto. Su cálculo efectivo se corresponde con la probabilidad inversa de los datos obtenidos de un conjunto de pruebas $W = w_1w_2...w_N$, normalizados por el número de palabras. Para un modelo de 2-gramas, esto sería:

$$PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i|w_{i-1})}}$$

Por tanto, minimizar la perplejidad equivale a maximizar la probabilidad del conjunto de pruebas según el modelo lingüístico [35].

La última métrica que ofrece *bitermplus* para la evaluación de modelos es la **coherencia semántica**. Esta medida puede interpretarse como la relación lógica sin contradicción entre los elementos (conceptos, proposiciones y temas) que componen un texto.

Por lo tanto, gracias a ella podemos evaluar la calidad de los temas detectados en base a las coocurrencias de términos en los documentos, al ser el grado en que un tema está respaldado "por un conjunto de textos (corpus de referencia). Para ello, es necesario una lista de los términos más frecuentes para cada tema, pues éstos se representan de cara al usuario a partir de dicha lista; $V^{(z)} = (v_1^{(t)}, \dots, v_M^{(t)})$. Vemos que la extensión de ésta ya se definió gracias al parámetro M . Así, el cálculo de la coherencia semántica se realiza como:

$$C(t, V^{(z)}) = \sum_{m=2}^M \sum_{l=1}^{m-1} \log \frac{C(v_m^{(t)}, v_l^{(t)}) + 1}{C(v_l^{(t)})}$$

6.3.2. Resultados

Para la visualización de resultados, *bitemrplus* hace uso de la librería desarrollada por el propio Terpilowski: **tmpplot**, un paquete de Python para el análisis y la visualización de los resultados del modelado de temas. Proporciona una interfaz de informe interactivo que tomando prestado mucho de LDavis/pyLDavis y se basa en él ofreciendo una serie de métricas para calcular distancias de temas y una serie de algoritmos para calcular coordenadas de dispersión de temas [64]. Este resultado final se genera como una consola interactiva, lo que imposibilita su guardado para consultas futuras. Dicha consola se compone de tres gráficos, de los cuales no puede omitirse ninguno o resultará en problemas a la hora de visualizar el gráfico:

- **Mapa de distancia entre temas.** Análogo al visto en LDA, aunque en esta ocasión el tópico 0 no representa al conjunto de todos ellos, si no que es uno concreto. La distancia entre temas puede calcularse de diversas maneras en vez de únicamente usando MDS. No obstante, para que la comparativa entre LDA y BTM sea coherente, debe fijarse dicha distancia.
- **Términos más relevantes.** Equivalente al usado en la visualización de LDA, cuya relevancia también puede ajustarse en función del parámetro λ .
- **Documentos más relevantes por tema.** Lista de los documentos con mayor presencia en cada tema, pudiendo mostrar una horquilla de entre 0 y 100 de éstos.

Cabe destacar que la interactividad se ha visto reducida, pues al pulsar en los gráficos de burbujas o en los propios términos, ya no se resaltan tal y como se producía en las páginas de LDA. En su lugar, cada tema a inspeccionar debe elegirse manualmente a través de un selector. Los tres gráficos pueden verse en la figura 6.13, donde se han omitido los parámetros relativos a la selección de tema y ajuste de los diferentes gráficos.

Los resultados obtenidos son muy similares a los que se consiguieron usando LDA. Debido a este hecho, y aunado con la reducción de funcionalidades de los cuadernos de visualización, no se ha realizado un estudio tan pormenorizado de los resultados obtenidos en este apartado y principalmente se ha corroborado la correspondencia de los mismos con los ya presentes en LDA.

Respecto a los **resultados en sí**, vemos que el número ideal de temas calculados en base a la entropía ronda los 10 para los datos de tweets, mientras que para reseñas se reduce a simplemente 7 (lo cual es lógico al ser textos con una variabilidad de temas mucho menor). Si bien es cierto que los valores de entropía resultan adecuados, tanto los resultados de perplexidad como de coherencia son bastante malos. Esto puede ser problema de los propios datos, que deban ser cribados de manera más exhaustiva, o bien errores de la implementación de la propia librería. No obstante, debido a las limitaciones temporales del proyecto se mantendrán estos resultados obtenidos, dejando cualquier posible mejora como trabajo para futuras iteraciones del trabajo.

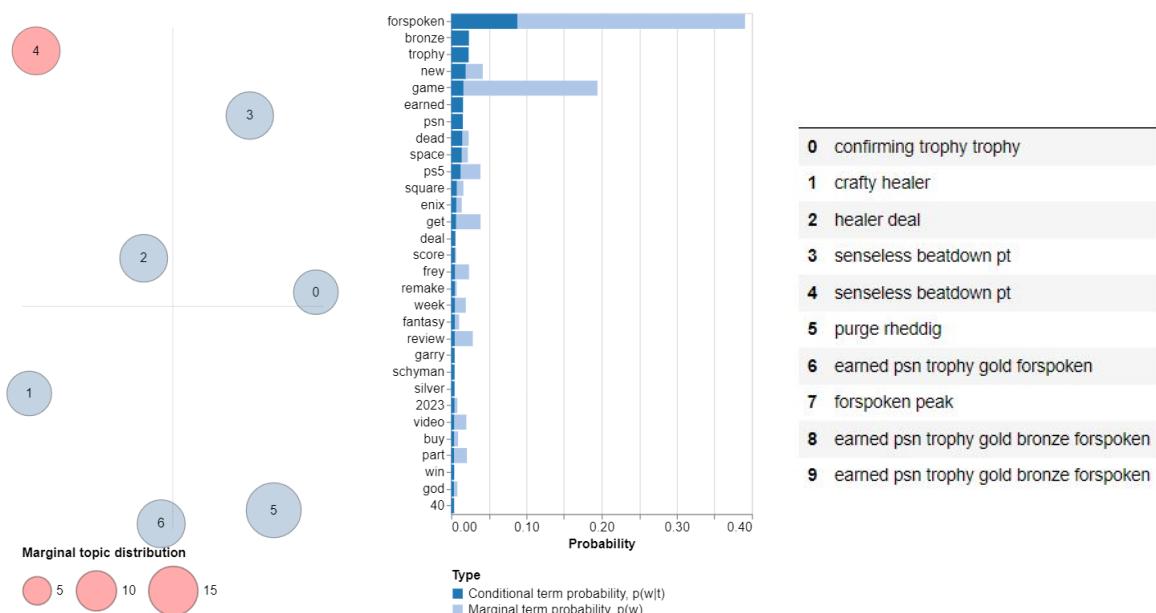


Figura 6.13: Consola de visualización de resultados del modelo relativo a los tweets que mencionan el videojuego Forspoken, seleccionando el tema 4 para analizar.

A pesar de todo, los temas obtenidos siguen resultando coherentes y continúan la línea que ya se definió con los modelos de LDA, pues por ejemplo en los tweets de Xbox siguen obteniéndose temas enfocados al parche de cuentas baneadas de GTA, o la predominancia del Nintendo Direct en la mayoría de los tweets que mencionan a Nintendo.

Por mencionar un tema que no se comentase antes, podemos destacar que, en los tweets que hacen referencia a Forspoken, se distingue un tema de manera evidente del resto que trata sobre mensajes de jugadores informando a través de Twitter de que han obtenido un logro en el videojuego. Esto puede verse en la figura 6.13, donde en los términos más relevantes encontramos tokens como "*trophy*", "*bronze*" o "*earned*"; siendo a su vez los documentos más relevantes de ese tema mensajes que evidencian el asunto del tema.

Capítulo 7

Sentiment analysis

El **análisis de sentimientos** hace referencia al proceso de clasificar los comentarios u opiniones presentes en un texto en categorías como "*positivo*" o "*negativo*", habitualmente existiendo también una categoría intermedia entre ambas catalogada como "*neutral*". En el contexto de *data science* y aprendizaje automático, el análisis de sentimientos también se denomina *opinion mining* o, en terminología de marketing, *Voice of the Customer* (VoC). Puede ser una herramienta muy útil para comprobar la afinidad hacia marcas, productos o dominios. Aplicado a los medios sociales, permite obtener una visión general de la opinión pública sobre temas específicos. No obstante, también tiene sus limitaciones y no debe considerarse como un reflejo completamente fidedigno de la realidad.

El análisis de sentimientos suele enfocarse como un problema de clasificación dentro del campo del aprendizaje automático supervisado, al basar la designación de la categoría otorgada en la información que contiene cada texto. Este es un problema no trivial con múltiples desafíos, estando gran parte de ellos relacionados con la manera en que funciona el propio lenguaje, ya que una misma palabra puede tener connotaciones distintas en función del contexto en que se utilice. Un ejemplo de esto puede verse dentro de las propias redes sociales, pues la palabra "*monstruo*" podría tener connotaciones negativas para referirse a alguien despreciable, aunque en la jerga juvenil puede emplearse para ensalzar a una persona resaltando su valía; "*Tío, eres un monstruo*". Esta tarea de detección de sentimientos alcanza cotas aún más elevadas de dificultad cuando en los textos se emplea humor o sarcasmo, alterando completamente el significado de la información [17].

Hoy en día, lo habitual es mezclar este enfoque con el uso de métodos basados en léxicos. Un **léxico** es el conjunto de palabras que conforma un determinado lecto o modalidad lingüística y, por extensión, también se denomina así a los diccionarios que los recogen. En el contexto del análisis de sentimientos, hace referencia a una colección de palabras y frases a las que se asignan puntuaciones de sentimiento en función de su significado y contexto. Un buen léxico debe abarcar una amplia gama de vocabulario y expresiones, así como términos específicos del ámbito y argot. También debe actualizarse periódicamente para reflejar los cambios y tendencias en el uso del lenguaje.

Por lo tanto, las técnicas empleadas en la actualidad, que mezclan ambos enfoques, suelen acabar recurriendo a clasificadores probabilísticos para asignar puntuaciones de positivo, negativo o neutro a una frase, documento o entidad; tal y como puede verse en el ejemplo 7.1, donde el valor *compound* hace referencia al sentimiento general del texto conocido como **polaridad**, cuyo valor oscila entre -1 y 1. Para textos cortos como los presentados en este trabajo (tweets y reseñas), las limitaciones de longitud de los documentos tratados no ofrecen distinción entre análisis de frases y el documento en su totalidad.

```
best class gift timerra fire emblem engage
{ 'compound': 0.7964, 'neg': 0, 'neu': 0.386, 'pos': 0.523}
```

Código 7.1: Ejemplo de asignación de puntuaciones de sentimiento a un texto

Para este proyecto, se realizará una clasificación del sentimiento asociado a tweets y reseñas a partir de modelos preentrenados disponibles en la red, comparando los resultados obtenidos entre los diferentes modelos. Para el caso de las críticas de videojuegos, también se realizará una comparativa de dichos resultados con otros baremos como la subjetividad del texto o la utilidad de la valoración según el criterio de otros usuarios de la comunidad. Además, se generará un modelo propio para calcular el sentimiento asociado a éstas usando un clasificador probabilístico.

7.1. Modelos preentrenados

El uso de **modelos de clasificación preentrenados** supone un gran punto de partida para iniciarse en esta tarea, pues han sido entrenados exprofeso por profesionales para desarrollar esta labor, dando acceso a una amplia variedad de funcionalidades con solamente unas pocas líneas de código. No obstante, debe tenerse en cuenta que al desarrollarse éstos normalmente para propósitos generales, están sometidos a las limitaciones de flexibilidad de las cuales los modelos propios no adolecen, al ser el mismo científico de datos quien designa los conjuntos de entrenamiento para dotar de la funcionalidad deseada a su modelo. En esta sección se verán tres modelos diferentes para la clasificación del sentimiento general de textos, comparándolos entre sí.

7.1.1. BERT

BERT son las siglas de *Bidirectional Encoder Representations from Transformers*, y se trata de un modelo preentrenado de última generación para tareas de PLN. Fue desarrollado por Google en 2018 y lo primero que hace es usar un embebimiento como los presentados en la sección 6.2.2 para representar las palabras como vectores. Una vez hecho esto, se pasan a una red neuronal profunda con múltiples capas de transformadores para codificar el significado y el contexto de palabras y frases desde ambas direcciones. BERT se puede refinar para tareas específicas, como el análisis de sentimientos, añadiendo una capa de clasificación sobre el modelo preentrenado y entrenándolo con datos etiquetados [31].

Existen variantes de este modelo como **RoBERTa** (*Robustly optimized BERT approach*), que usa BERT como base modificando algunos de sus hiperparámetros, eliminando el objetivo de preentrenamiento de la siguiente frase y entrenando con *mini-batches*¹ y tasas de aprendizaje mucho mayores.

Para este proyecto se ha empleado un modelo de RoBERTa entrenado con unos 58 millones de tweets y refinado para el análisis de sentimientos de tweets escritos en inglés, siguiendo el marco de referencia TweetEval [9]. A grandes rasgos, su funcionamiento consiste en analizar individualmente cada uno de los textos, dando una puntuación entre -4 y 4 para cada una de las categorías existentes (positivo, neutro o negativo), indicando la intensidad de cada uno de estos sentimientos en el texto. Estas puntuaciones luego se normalizan y se convierten en un porcentaje final para cada una de las categorías.

```
making progress finally let play fire emblem engage
[-3.4069092 1.1576403 2.1524656]
[0.00280364 0.26920322 0.7279932 ]
```

Código 7.2: Puntuaciones y porcentajes (negativo, neutro y positivo; respectivamente) asignadas por RoBERTa a un tweet.

El principal **inconveniente** de este modelo reside en su **velocidad**, pues la asignación de una puntuación hace uso de tensores para analizar el texto, ralentizando bastante el cálculo final. Además, y al contrario de lo que ocurre con los otros modelos que se verán, no se incluye una funcionalidad para calcular una puntuación agregada que indique el sentimiento general que transmite el texto. Es por ello que, teniendo este fin en mente, se han considerado dos métricas para poder evaluar este sentimiento final:

- **maxROBERTA.** Se elige como sentimiento predominante la categoría cuyo porcentaje es mayor; etiquetando 0, 1 ó 2. No obstante, este cribado resulta algo crudo y no tiene en cuenta el mayor peso de las categorías positiva (2) y negativa (0) respecto a la neutra (1), por lo muchos tweets se clasifican en esta última.
- **compoundROBERTA.** Retomando las puntuaciones otorgadas por el modelo, podemos normalizar éstas para calcular una función que dilucide el sentimiento final en base a estos valores:

$$f(pos, neu, neg) = \begin{cases} 0, & \text{si } neu > |pos - neg| \\ \frac{pos+neu-neg}{\sqrt{(pos+neu-neg)^2}}, & \text{en cualquier otro caso.} \end{cases}$$

donde *pos*, *neu* y *neg* son las puntuaciones no normalizadas del sentimiento positivo, neutro y negativo presente en el texto. Así, en caso de que neutro fuera el sentimiento predominante (pues su puntuación nunca es negativa), le damos un valor de 0. En caso contrario, normalizamos las puntuaciones y vemos si el sentimiento obtenido es positivo (1) o negativo (-1). La idea de esta función es emular la de polaridad que ofrecen los otros modelos preentrenados.

¹Al entrenar un modelo de *machine learning*, el tamaño del **batch** o lote es un hiperparámetro que define el número de muestras con las que hay que trabajar antes de actualizar los parámetros internos del modelo. Un conjunto de entrenamiento puede dividirse en uno o más *batches* [16].

7.1.2. VADER

VADER (*Valence Aware Dictionary and sEntiment Reasoner*) es una herramienta de análisis de sentimientos basada en léxicos y reglas que se adapta específicamente a los sentimientos expresados en los medios sociales [33]. Para usarlo, basta recurrir al paquete NLTK e importar el modelo **SentimentIntensityAnalyzer**, cerciorándose antes de que el equipo dispone del léxico de VADER descargado. Una vez instanciado el modelo, basta aplicarlo a los textos de manera individual.

Su principal **ventaja** es que es muy fácil de usar y calcular, así como bastante veloz. Además, aparte del porcentaje específico de sentimiento manifestado en cada tweet, también da una puntuación compuesta que permite clasificarlo como positivo, neutro o negativo con mayor precisión. Este valor es lo que se denomina **polaridad**, siendo una puntuación normalizada que oscila entre -1 y 1 e indica el sentimiento positivo o negativo subyacente en cada texto.

Dado que VADER ha sido entrenado específicamente para detectar el sentimiento asociado a textos provenientes de medios sociales, se usará para analizar tanto los tweets como las reseñas extraídas. En este caso, se considerarán como neutros aquellos textos que den una puntuación de 0 , y valorándose como positivos o negativos en los otros casos (según si el resultado es positivo o negativo, respectivamente).

7.1.3. TextBlob

TextBlob es una librería de Python para el procesamiento de datos textuales. Proporciona una API sencilla para sumergirse en tareas comunes de procesamiento del lenguaje natural como el análisis de sentimientos, la detección de la subjetividad subyacente a un texto o la traducción [40]. Al calcular el sentimiento, TextBlob toma la media de todo el texto, por lo que sólo se tiene en cuenta el significado común de una palabra en todo el texto. Esta tarea se realiza mediante el uso de la base de datos WordNet y a partir de un modelo de NLTK entrenado sobre un corpus de reseñas de películas. Por este motivo, se empleará sólo para calcular el sentimiento asociado a las reseñas de videojuegos obtenidas de Metacritic.

Una de las ventajas de TextBlob es la posibilidad de hacer uso de otras funcionalidades adicionales aparte de la propia detección de sentimientos, como es el cálculo del grado de subjetividad presente en un texto. La **subjetividad** se utiliza para determinar si el texto analizado expresa una opinión o no. Matemáticamente, se representa como un valor entre 0 y 1 . Cuanto mayor sea el valor, más cercano a una opinión se considera el texto. A pesar de que todos los textos considerados son reseñas de videojuegos, y por ende tienen asociado cierto grado de subjetividad intrínseco, se querrá ver si las críticas que hacen uso de valoraciones más personales tienen alguna influencia en la polaridad del texto final (tal y como se ve en la figura 7.11).

7.2. Modelos específicos

La detección de sentimiento de un texto se realiza tras haber entrenado previamente clasificadores de sentimiento. Hasta ahora, sólo hemos utilizado clasificadores preentrenados con textos generales o específicos. La principal ventaja a la hora de entrenar un modelo propio es que se pueden decidir los conjuntos de entrenamiento para que sean lo más similares a los datos reales que se acabarán usando, obteniendo el modelo mejores resultados. En esta sección, también entrenaremos nuestro propio clasificador para identificar si una reseña es positiva, negativa o mixta. Realizamos esta tarea únicamente sobre las críticas de videojuegos, ya que todos los datos están etiquetados de antemano en una de las tres categorías, por lo que el entrenamiento de los modelos puede realizarse fácilmente.

7.2.1. Naive Bayes

Para realizar la tarea de clasificación, haremos uso de un **clasificador Naive Bayes multinomial (NBC)**, que es un clasificador de Bayes cuya idea básica es simplificar las hipótesis de cómo las características interactúan entre sí. La idea subyacente detrás de esto es la de una **bolsa de palabras** o *bag of words*, que consiste en considerar el texto de un documento como un conjunto de palabras no ordenado en el cual se ignora la posición de éstas, manteniéndose únicamente la frecuencia de aparición de los términos. En la figura 7.1 se puede observar un ejemplo de este proceso a partir de una reseña en inglés de una película [35].

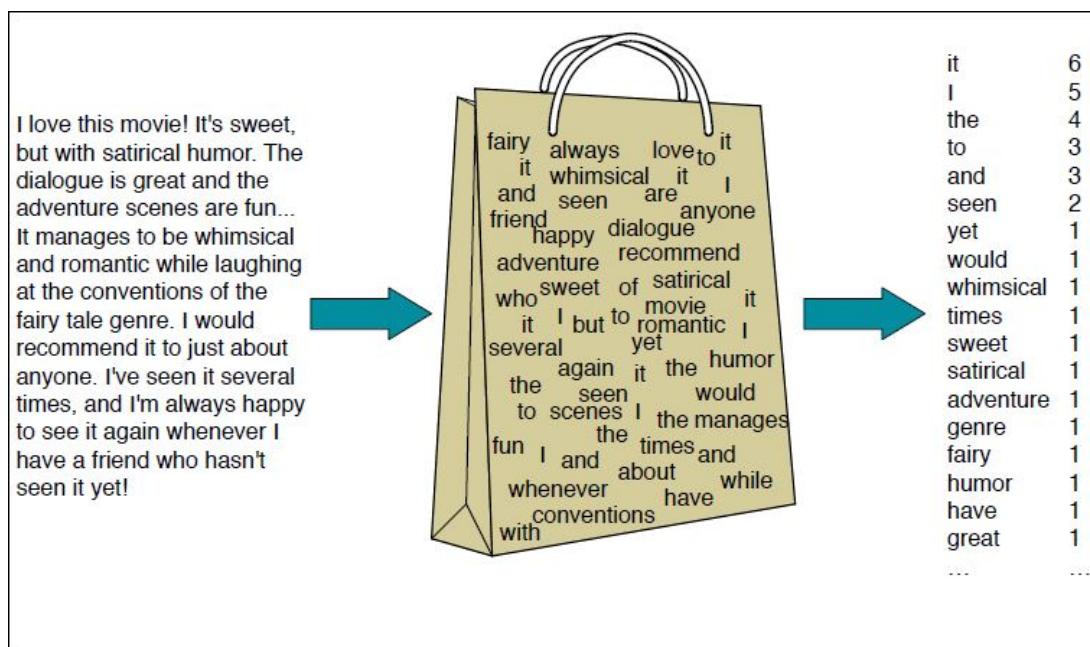


Figura 7.1: Idea intuitiva detrás del clasificador de Bayes y la bolsa de palabras

Naive Bayes es un clasificador probabilístico, lo que significa que para un documento d , de todas las clases $c \in C$, el clasificador devuelve aquella clase \tilde{c} que tiene la máxima probabilidad posterior dada la probabilidad del documento. Para este cálculo, se emplea la **inferencia Bayesiana**, cuya idea es como la que se usaba en los N-gramas: simplificar los cálculos aplicando la regla de la probabilidad condicionada (descartando la probabilidad $P(d)$ al ser una constante) y asumiendo que las diferentes características que representan un documento (f_1, f_2, \dots, f_n) son independientes entre sí (**hipótesis Naive Bayes**).

$$\begin{aligned}\tilde{c} &= \arg \max_{c \in C} P(c|d) = \arg \max_{c \in C} \frac{P(d|c)P(c)}{P(d)} = \arg \max_{c \in C} P(d|c)P(c) = \\ &= \arg \max_{c \in C} P(f_1, f_2, \dots, f_n|c)P(c) = \arg \max_{c \in C} P(c) \prod_{i=1}^n P(f_i|c)\end{aligned}$$

En concreto, las características f_i son todas las posibles posiciones w_i que podría ocupar una palabra a lo largo del texto. Además, al igual que ocurría con los N-gramas, estos cálculos se realizan en formato logarítmico para evitar problemas de *underflow*, por lo que la ecuación final resulta:

$$\tilde{c} = \arg \max_{c \in C} \log P(c) + \sum_{i \in \text{posiciones}} \log P(w_i|c)$$

Por lo tanto, la predicción final de la clase se realiza como una función lineal a partir de un conjunto de datos de entrada. Los clasificadores que utilizan una combinación lineal de las entradas para tomar una decisión de clasificación se denominan **clasificadores lineales**.

7.2.2. Entrenamiento del modelo

Una vez definido el modelo, es necesario entrenarlo a partir de los datos que se tienen para que pueda realizar las labores de clasificación deseadas. Sin embargo, no se usan todos los datos para el entrenamiento del modelo. Lo ideal es destinar alrededor del 60 u 80% para entrenar el modelo (**conjunto de datos de entrenamiento**), quedando el resto disponible para evaluar la calidad del modelo obtenido (**conjunto de datos de prueba**). Además de esto, es recomendable **estratificar** (dividir los datos en grupos o estratos según sus características) los datos para que su distribución final sea lo más similar posible al de las futuras muestras que se predecirán. Una vez definidos estos parámetros, también es necesario concretar otros como el tamaño del *batch* o el número de épocas².

No obstante, cabe preguntarse cómo se usan estos datos para entrenar el modelo. Retomando la expresión vista anteriormente, lo más común es estimar las probabilidades $P(c)$ y $P(f_i|c)$ mediante una estimación de máxima verosimilitud: usar las frecuencias relativas de los datos de entrenamiento.

²El número de **épocas** es un hiperparámetro que define el número de veces que el algoritmo de aprendizaje trabajará a través de todo el conjunto de datos de entrenamiento, actualizándose los parámetros internos del modelo. Una época se compone de uno o varios *batches* [16].

Así, si N_c es el número de documentos de clase c en el conjunto de datos de entrenamiento y N_d el número total de documentos, podemos estimar $\tilde{P}(c) = \frac{N_c}{N_d}$.

La estimación de características se realiza de manera análoga, siendo éstas la frecuencia de aparición en la bolsa de palabras para documentos de clase c . Además, para evitar probabilidades nulas, se repite el mismo truco que el usado en N-gramas, añadiendo 1 como valor por defecto para cada conteo³:

$$\tilde{P}(w_i|c) = \frac{C(w_i, c) + 1}{\sum_{w \in V} (C(w, c) + 1)} = \frac{C(w_i, c) + 1}{\sum_{w \in V} C(w, c) + |V|}$$

donde $C(w_i, c)$ es la fracción de veces que la palabra w_i aparece entre todas las palabras de todos los documentos de clase c y V es el vocabulario del corpus.

7.2.3. Evaluación del modelo

Tras haber entrenado el modelo, es necesario evaluar su rendimiento para comprobar la efectividad del mismo. Para ello, se hace uso el conjunto de datos de prueba y se extraen diversas métricas de los resultados obtenidos, los cuales se calculan en función de si la labor de clasificación se ha realizado o no de manera correcta:

- **True Positive (TP)**: Valores predichos correctamente (predicción correcta)
- **False Positive (FP)**: Valores predichos incorrectamente (predicción incorrecta)
- **True Negative (TN)**: Valores rechazados correctamente (predicción correcta)
- **False Negative (FN)**: Valores rechazados incorrectamente (predicción incorrecta)

Aunque estos valores resultan especialmente intuitivos en el caso binario, para la situación en la que hay n-clases basta realizar este análisis clase por clase. Estos resultados se pueden resumir mediante la técnica conocida como **matriz de confusión**, la cual proporciona información sobre los aciertos y errores del modelo de clasificación, así como del tipo de errores que comete. La figura 7.2 muestra la matriz de confusión generada tras evaluar los resultados obtenidos del modelo capaz de clasificar reseñas de usuarios. En él, vemos que no se ha predicho ninguna de ellas como "*Mixed*", debido a que tampoco se consta de una gran cantidad de datos para entrenar y, tal y como se verá en la sección 7.3.3, las críticas de usuarios suelen estar altamente polarizadas.

Retomando la evaluación del modelo, en base a estos valores pueden obtenerse diversas métricas que informan sobre la validez del modelo obtenido:

- **Accuracy**. Frecuencia con la que el método evaluado realiza la predicción correcta. Se calcula como la suma de las predicciones verdaderas dividida por el número total de predicciones:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Recall**. Fracción de casos positivos que se predijeron como positivos:

$$\text{Recall} = \frac{TP}{TP + FN}$$

³Esto se conoce como **Laplace smoothing** o suavizado de Laplace.

- **Precision.** Representa la exactitud del método. Se calcula como la proporción de casos que se predijeron como positivos y que efectivamente lo fueron, dividida por el número total de casos que se predijeron como positivos:

$$Precision = \frac{TP}{TP + FP}$$

- **F-score.** Métrica que combina *recall* y *precision* para determinar la precisión de la prueba. Para problemas de n-clases, suele hallarse esta puntuación para cada una de ellas. Se calcula como:

$$F\text{-score} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

- **Support.** Número de observaciones que se predicen en una clase determinada.

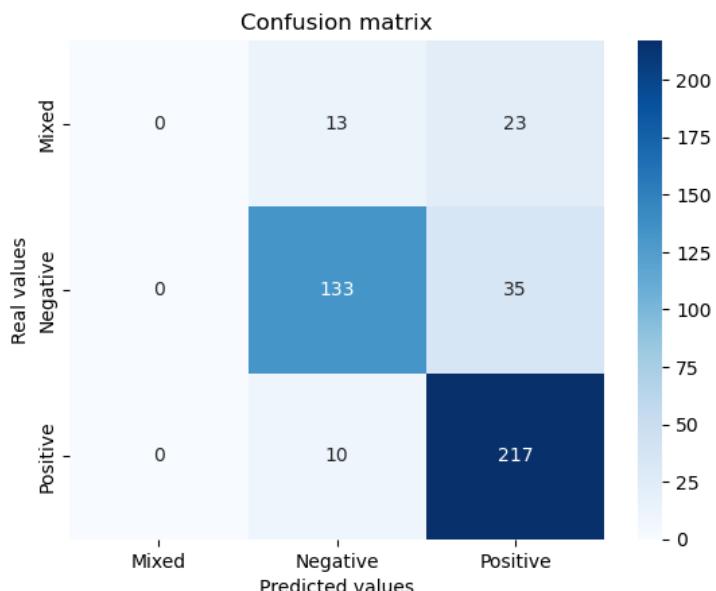


Figura 7.2: Matriz de confusión del modelo entrenado con reseñas de usuarios para predecir la categoría de una crítica

Además de todas estas métricas, también se suele hacer uso de la **validación cruzada** o *cross-validation*, una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que sean independientes de la partición entre datos de entrenamiento y prueba. La manera más habitual para ponerlo en práctica es el método conocido como ***K-folds***, en la cual los datos de entrada se dividen en *K* partes, una de las cuales se reserva para las pruebas y la otra $K - 1$ para el entrenamiento. Este proceso se repite *K* veces y se promedian las métricas de evaluación. Esto ayuda a determinar la capacidad de generalización de un modelo a nuevos conjuntos de datos.

Con respecto a los **resultados obtenidos en los cuadernos**, vemos que el principal problema ha sido la escasez de datos para el entrenamiento, así como el sesgo de los mismos. Si ya comentábamos la escasez de reseñas de usuarios de valoración intermedia en la figura 7.2, las críticas de prensa especializada adolecen del mismo problema para las valoraciones negativas. De todas formas, en ambos casos se han obtenido modelos finales con una precisión de entre el 70 % y el 80 %, por lo que puede considerarse que ambos clasificadores dan un rendimiento aceptable.

7.3. Resultados

Una vez visto el marco teórico sobre el que se sustentan las labores de análisis de sentimiento, ya se puede llevar a cabo el escrutinio de los propios resultados obtenidos. Debido a la gran cantidad de gráficas y estadísticas generadas, únicamente se comentarán algunas de ellas, pudiéndose consultar el resto en los propios cuadernos incluidos como contenido adjunto. Además, en esta sección sólo se hablará de los resultados obtenidos con los **modelos preentrenados**, pues ya se habló de los clasificadores creados y su precisión en el apartado 7.2.3.

EL análisis se divide en función de los datos considerados, distinguiendo los tweets de las reseñas. En estas últimas también se realiza la distinción de si son críticas escritas por prensa especializada o por la propia comunidad pues, tal y como podía verse en estudios como el de Pellarolo (figuras 3.3 y 3.4), existe una diferencia palpable entre las valoraciones que dan ambos colectivos a la hora de dar puntuaciones.

7.3.1. Tweets

En el caso de los tweets, únicamente podemos comparar los resultados obtenidos por las tres métricas definidas con los modelos RoBERTa y VADER. A pesar de que RoBERTa ha sido entrenado específicamente para la detección de sentimiento en tweets, el no gozar de una métrica definida para realizar tareas de clasificación penaliza bastante su rendimiento con respecto a lo que ofrece el modelo VADER.

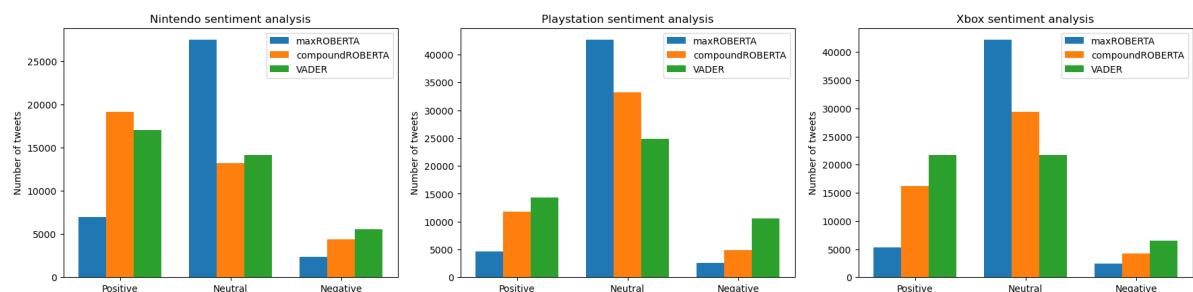


Figura 7.3: Sentimiento detectado según las métricas definidas para RoBERTa y VADER en los tweets que mencionan a Nintendo, Playstation y Xbox

Como puede verse en las figuras 7.3 y 7.4, la métrica maxROBERTA tiende a clasificar casi todos los textos como neutros. La puntuación creada en este estudio, compoundROBERTA, ofrece mejores resultados, y que además se asemejan a los obtenidos mediante VADER. En cualquier caso, las tres compañías tienen una cantidad equivalente de tweets tanto positivos como neutros, siendo los negativos bastante inferiores, por lo que se puede deducir que las tres marcas tienen una percepción positiva por parte de la comunidad. En el caso de sus respectivos lanzamientos, tanto Hi-Fi Rush como Fire Emblem Engage reciben mayoritariamente comentarios positivos, mientras que Forspoken parece haber tenido una acogida más templada por parte de los usuarios, pues tiene aproximadamente la mitad de tweets positivos mencionándolo con respecto a sus competidores.

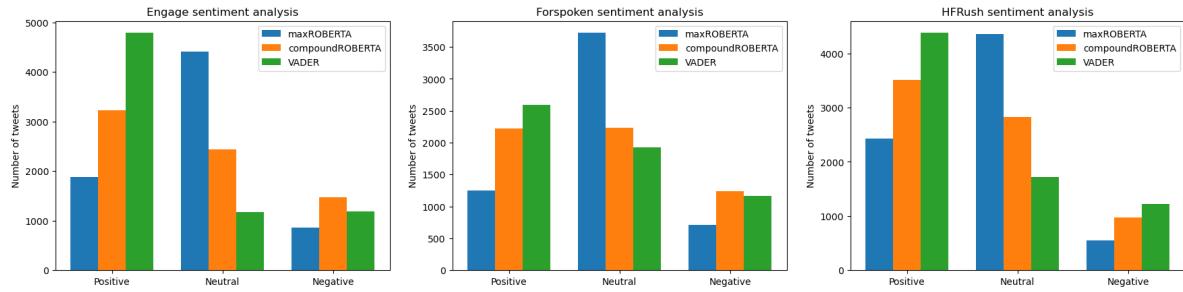


Figura 7.4: Sentimiento detectado según las métricas definidas para RoBERTa y VADER en los tweets que mencionan a Fire Emblem Engage, Forspoken y Hi-Fi Rush

Además, puede comprobarse la efectividad de los modelos de clasificación obteniendo una muestra aleatoria de los tweets considerados. Para ello, se ha seleccionado como mínimo un tweet de cada categoría designada por la métrica maxROBERTA, al ser la que al parecer detecta únicamente aquellos tweets más polarizados. Si vemos la figura 7.5, prácticamente todos los tweets han sido categorizados adecuadamente considerando su texto. No obstante, sí se observa alguna discrepancia como es el caso del tweet acerca de Hi-Fi Rush, el cual ambas métricas de ROBERTA han detectado como negativo pero VADER lo cataloga como positivo. Lo cual se explica por el lenguaje agresivo del tweet, que no deja muy claro el sentimiento predominante del texto, aunque una vez analizado con atención se ve que tiene connotaciones positivas al alabar el producto por su calidad.

	text	ROBERTAmaxSentiment	ROBERTAcompSentiment	VADERsentiment	tweet
15440	fuck nintendo tho greedy assholes	0	-1.0	-0.7003	Nintendo
37540	Check out my broadcast from my PlayStation 4! #PS4live (The Last of Us™ Remastered) live at https://t.co/CiBZEurt5M	1	0.0	0.0000	Playstation
21840	Will be raiding an non affiliate in a few hrs come say hi come watch us win \n#twitchstreamer #StreamersConnected #streamer #xbox #fornite #nonaffiliate #SupportSmallStreams #support #hype #love \n https://t.co/pXk0zK4JyH	2	1.0	0.5859	Xbox
4311	While I really appreciate the weapon triangle returning in Engage and it also changing how you interact especially early game. It's so strange to me when I see relatively new FE people talk about how they were pissed off 3H didn't have	2	1.0	-0.5186	Engage
3766	Finally have a little time to dive into #Forspoken\nThis is fun! Not sure what some people were complaining about: #Frey is not amused. https://t.co/WhRkXRoxaP	2	1.0	0.7458	Forspoken
5146	hi-fi rush is a goddamn banger\nno early access, no gb bugs, got released the day it got announced, music slaps, free on the gamepass (or 30 euros), fun gameplay, characters and story (fuck corporations)	0	-1.0	0.1531	HFRush

Figura 7.5: Muestra de algunos tweets y la clasificación otorgada por cada modelo preentrenado.

7.3.2. Reseñas de prensa especializada

El principal problema de las críticas de prensa para los títulos recogidos es la escasez de datos para poder realizar comparaciones adecuadas. Además, casi todas ellas no suelen dar una puntuación que pueda catalogarse como negativa, lo cual resultó también un problema cuando se entrenó el clasificador de la sección 7.2.3. Por este motivo, el análisis de los gráficos de subjetividad o los gráficos de caja de cómo se distribuyen las reseñas respecto a la polaridad se comentarán en el siguiente apartado, en la cual ya existen más datos para dar dicha comparativa.

No obstante, sí que puede realizarse otra comparativa. La principal ventaja de las reseñas es que incluyen una **puntuación numérica** de cada producto. Si se normaliza dicho valor, puede emplearse para comparar la puntuación otorgada con el propio texto de las críticas, comprobando si se corresponde esta nota con lo que transmite el texto. Para ello, y dado que las valoraciones de la prensa dan una puntuación numérica entre 0 y 100, debe realizarse el siguiente **escalado** para así compararlo con la polaridad de los textos (valor entre -1 y 1):

$$f(x) = \left(\frac{2x}{100} - 1 \right) \in [-1, 1] \text{ para } x \in [0, 100]$$

La figura 7.6 muestra estos gráficos para los productos de cada compañía en sus propias plataformas. Al igual que pasaba con RoBERTa, TextBlob parece que tiene cierta tendencia a optar por valores intermedios, sin polarizar en exceso la puntuación final. En el otro extremo se encuentra VADER que, al compararlo con las notas normalizadas dadas, parece exagerar los sentimientos presentes en el texto, aunque su aproximación resulta más certera que la de TextBlob. En cualquier caso, Forspoken parece ser el único producto de los tres que ha obtenido valoraciones mediocres o no tan positivas, al contrario de lo que ocurre con los otros dos videojuegos, que han obtenido críticas muy favorables.

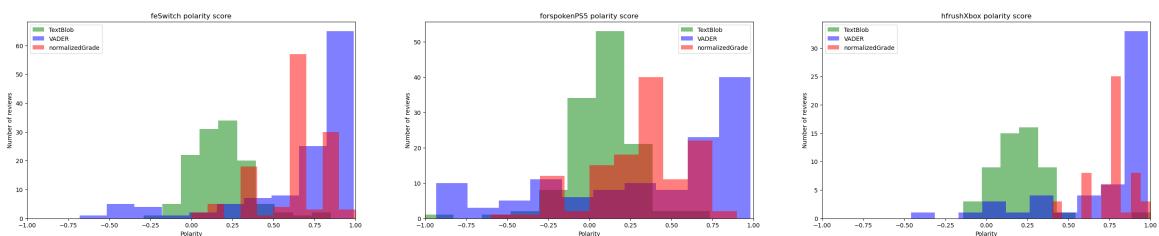


Figura 7.6: Comparativa entre la polaridad del texto y la nota dada por prensa especializada a Fire Emblem Engage (Switch), Forspoken (PS5) y Hi-Fi Rush (Xbox Series X)

El comportamiento poco certero de VADER se debe a que ha sido entrenado para reconocer el sentimiento de textos provenientes de medios sociales. Las reseñas de prensa, que tienen un tono más formal que dichos textos, le resultan más difíciles de ubicar y es por ello que comete dichos errores. No obstante, para reseñas de usuarios sí que ofrece un funcionamiento adecuado, tal y como muestra la figura 7.8.

En último lugar, considerando las valoraciones de subjetividad aportadas por TextBlob (figura 7.7), vemos que se obtienen valores intermedios en prácticamente todos los casos. Sin embargo, la escasez de reseñas no permite sacar conclusiones de los datos de los que se dispone actualmente, por lo que esta valoración se pospondrá hasta que se haga este mismo escrutinio con la opinión de los usuarios, las cuales son más cuantiosas y permitirán obtener resultados más claros tanto en este gráfico como al efectuar la comparativa entre los valores de polaridad y subjetividad detectados.

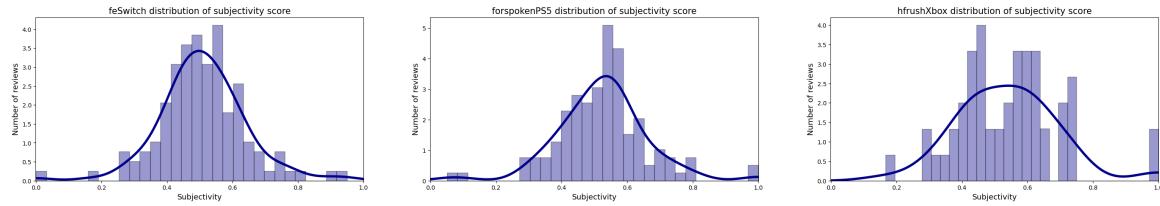


Figura 7.7: Distribución de la subjetividad detectada en las críticas de prensa especializada de Fire Emblem Engage (Switch), Forspoken (PS5) y Hi-Fi Rush (Xbox Series X)

7.3.3. Reseñas de usuarios

La comparación de polaridad en el caso de usuarios se realiza de manera análoga, aunque en esta ocasión la función de escalado debe adaptarse a las notas de los usuarios (en lugar de ser sobre 100 son sobre 10):

$$f(x) = \left(\frac{2x}{10} - 1 \right) \in [-1, 1] \text{ para } x \in [0, 10]$$

En la figura 7.8 se observa de manera aún más marcada que en el caso de las críticas de prensa la mala acogida que ha tenido el videojuego Forspoken, mientras que Hi-Fi Rush ha sido reconocido de manera uniforme como un gran producto.

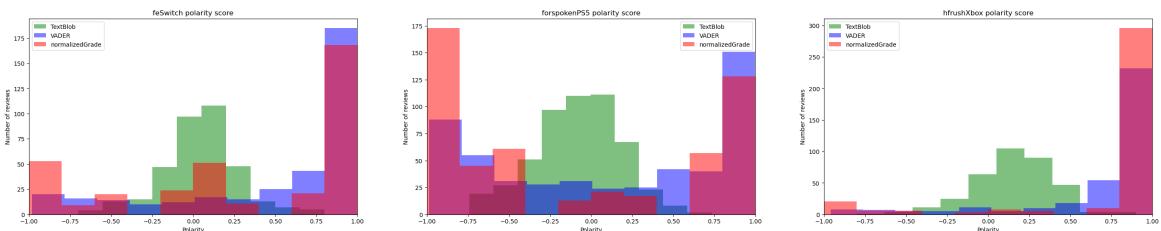


Figura 7.8: Comparativa entre la polaridad del texto y la nota dada por usuarios a Fire Emblem Engage (Switch), Forspoken (PS5) y Hi-Fi Rush (Xbox Series X)

Dichos resultados son aún más evidentes en la figura 7.9, pues las críticas negativas de Forspoken tienen valores más bajos de polaridad, mientras que las positivas de Hi-Fi Rush muestran puntuaciones más altas en este aspecto.

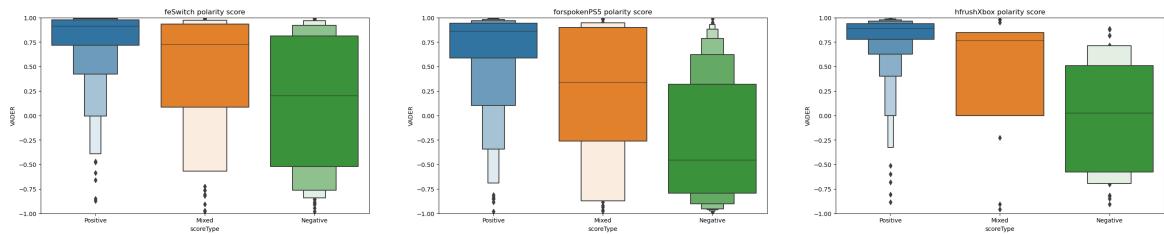


Figura 7.9: Gráficos de caja que indican la polaridad de los textos según el tipo de reseña

Retomando el **análisis de la subjetividad** de la sección previa, vemos que en esta ocasión sí disponemos de datos suficientes para poder construir histogramas de distribución de la subjetividad adecuados (figura 7.10). De nuevo, los valores generales se encuentran bastante centrados, aunque para Hi-Fi Rush se ha detectado un poco más de subjetividad en las valoraciones de lo habitual.

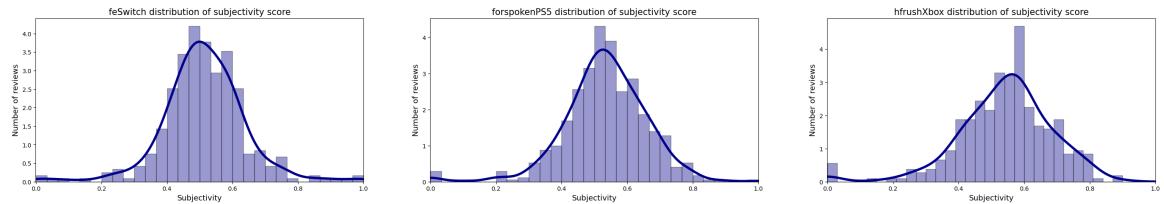


Figura 7.10: Distribución de la subjetividad detectada en las críticas de usuarios de Fire Emblem Engage (Switch), Forspoken (PS5) y Hi-Fi Rush (Xbox Series X)

Con respecto a la influencia de una valoración subjetiva respecto a la polaridad del texto, vemos en la figura 7.11 que la gran mayoría de las críticas se ubican en la zona central del gráfico, con valores intermedios tanto para polaridad como para subjetividad. Esto se debe a que la métrica de polaridad usada es la proporcionada por TextBlob, la cual suele dar valores no excesivamente polarizados (figura 7.8), por lo que podría ser conveniente repetir este análisis con los valores de polaridad dados por VADER. No obstante, se ha preferido mantener este análisis para que ambas métricas fuesen las proporcionadas por el mismo modelo.

A pesar de todo, puede observarse que Forspoken tiene un mayor número de valoraciones subjetivas con menor polaridad, mientras que Hi-Fi Rush ha obtenido críticas de mayor polaridad aunque manteniendo unos niveles de subjetividad por encima de la media. Por su parte, Fire Emblem Engage conserva la mayoría de sus reseñas en un espectro intermedio.

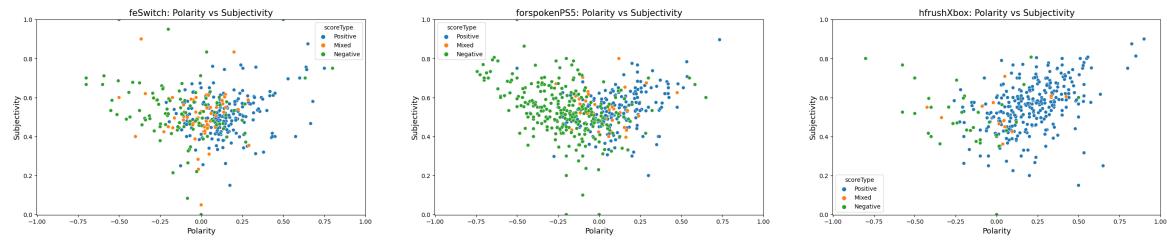


Figura 7.11: Comparativa entre los valores de polaridad y subjetividad detectados por TextBlob en las reseñas de usuarios de Fire Emblem Engage (Switch), Forspoken (PS5) y Hi-Fi Rush (Xbox Series X)

Además, las valoraciones de usuarios incluyen una métrica adicional de utilidad o *helpfulness*, que indica si una crítica le ha resultado útil al resto de la comunidad. Por ejemplo, para el caso del videojuego Fire Emblem Engage vemos que aquellas reseñas que han resultado más útiles a la comunidad son catalogadas como intermedias o negativas, pero que mantienen unos valores centrados de polaridad (figura 7.12). Esto se debe a que ésta es una saga de videojuegos con una amplia comunidad de seguidores, por lo que una parte de las reseñas positivas son comentarios de los fans alabando el juego, lo cual supone información no especialmente relevante para un jugador neutral que quiera informarse sobre el producto para decidir si lo adquirirá o no.

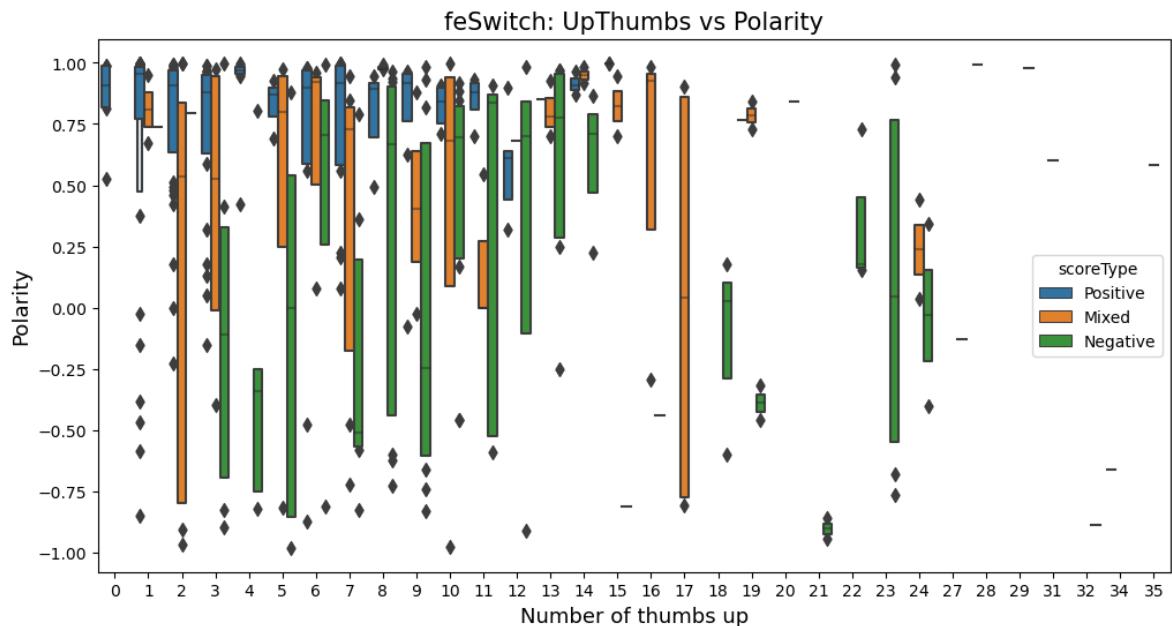


Figura 7.12: Comparativa entre la utilidad de las reseñas del videojuego Fire Emblem Engage y la polaridad detectada por VADER, así como la categoría en la que se encuadran

Capítulo 8

Conclusiones y trabajo futuro

En esta sección se extraerán conclusiones tanto de los productos y marcas analizados a lo largo de este trabajo, como del aprendizaje que ha obtenido el estudiante gracias al desarrollo del proyecto. También se incluyen posibles mejoras y líneas de trabajo alternativas en caso de que se realizasen futuras iteraciones del estudio final obtenido.

8.1. Conclusiones

8.1.1. Conclusiones del estudio

Una vez culminado el desarrollo del proyecto, únicamente resta corroborar si se han ido cumpliendo los objetivos marcados al inicio del mismo. Analicemos punto por punto si éste ha sido el caso:

- **OBJ-1:** Se han generado un total de siete cuadernos, así como una memoria final, que cubren todos los pasos y productos finales que compondrían el desarrollo de un proyecto de *data science* aplicando PNL.
- **OBJ-2:** Si bien la relevancia de los temas detectados se encontraba supeditada a la actualidad del momento, el proceso de extracción de temas también se ha llevado a cabo de forma satisfactoria (capítulo 6).
- **OBJ-3:** Tras consultar los datos recogidos en redes sociales y examinar las diferentes reseñas, hemos visto que Fire Emblem Engage y Hi-Fi Rush han tenido una recepción mayoritariamente positiva, al contrario de lo que ha pasado con Forspoken. De este último tampoco se puede decir que haya provocado un rechazo generalizado, pero sí ha tenido una recepción más tibia por parte de prensa y comunidad, destacando sus flaquezas por encima de sus virtudes.

Faltaría solamente el **OBJ-4**, el cual puede evaluarse ahora al haber pasado ya casi seis meses desde el lanzamiento de los tres títulos y ya se dispone de información sobre la acogida general de cada uno de estos productos. No obstante, debe tenerse en cuenta que, debido a políticas de las empresas, sólo una pequeña parte de estos datos se dan al público general, por lo que la principal fuente de información son tanto declaraciones de los propios estudios como páginas que recogen datos de ventas en plataformas como Steam (SteamDB).

En el caso de **Fire Emblem Engage**, el cual fue lanzado únicamente para Nintendo Switch, alcanzó en marzo de 2023 las 1.61 millón de copias vendidas en todo el mundo [30]. Estos resultados pueden considerarse un éxito si se tiene en cuenta que la anterior entrega de la saga, Three Houses, se había convertido en el juego más vendido de la franquicia tras alcanzar en diciembre de 2021 (dos años y medio después de su lanzamiento) los 3.82 millones de copias vendidas en todo el mundo. Esto corrobora los buenos datos que se han obtenido sobre el juego a lo largo del estudio.

Por su parte, **Forspoken** no ha gozado de la misma suerte. En el informe de beneficios publicado en febrero de 2023 por Square Enix, empresa matriz de Luminous Productions, se da respuesta a las malas críticas y los comentarios negativos que ha recibido el juego, mencionando además que las ventas del mismo han acabado resultando "*deslucidas*" [76]. Debido a esto, Luminous Productions ha sido reintegrada dentro de Square Enix en mayo de este mismo año, mientras sigue trabajando en parches para mejorar errores presentes en el juego, así como el último DLC de Forspoken que salió en junio [66]. No obstante, éste parece seguir adoleciendo de los mismos problemas que lastraban al juego base.

Este fracaso se hace aún más patente al consultar sus ventas en PC registradas en la plataforma Steam, donde ni siquiera a logrado entrar en el top 10 de juegos más vendidos. Especialmente sangrante queda la comparativa con **Hi-Fi Rush**, juego que prácticamente no gozó de ningún acto promocional ni campaña de marketing asociada, pero aún así ha logrado colarse como el octavo juego más vendido en Steam la semana de su lanzamiento [38]. Este hecho resulta especialmente grave si consideramos que el desarrollo de Forspoken costó el doble que el de Hi-Fi Rush [8], estando este último disponible para usuarios de Xbox Game Pass sin coste adicional, el cual también permite jugar en ordenador al título.

Ante estos datos, cabe preguntarse si el modelo de desarrollo convencional de las grandes superproducciones de videojuegos puede empezar a resultar improductivo, debido al alto riesgo y costes de desarrollos asociados a estos proyectos. Tal y como reflejan las ventas de ambos juegos, así como las impresiones recogidas a lo largo de este estudio, podría resultar más rentable un cambio de paradigma a desarrollos más contenidos, que primen la calidad de los títulos antes que la duración de los mismos, lo cual desembocaría en productos más asequibles para el usuario final (pues Forspoken costaba de salida 70\$ mientras que Hi-Fi Rush estaba disponible por menos de la mitad de precio [65]). Lo mismo puede aplicarse a los modelos de distribución convencionales pues, ante casos como el de Hi-Fi Rush, vemos que las declaraciones hechas por Phil Spencer, CEO de Microsoft, sobre el beneficio en las ventas finales de los títulos que salen en Game Pass han acabado resultando ciertas [73].

8.1.2. Conclusiones personales

Analizando este trabajo como una asignatura más del grado, creo que este proyecto me ha permitido crecer sobremanera como profesional. Si bien a lo largo de la carrera se plantean diversos trabajos para que vayamos adquiriendo competencias y habilidades en nuestro campo, creo que ninguno de ellos me ha resultado tan útil como este proyecto. En mi opinión, uno de los principales motivos por los que este trabajo me ha sido de tanta utilidad es la libertad de la que he gozado a lo largo del desarrollo, tanto por parte de la empresa como de la universidad, estando ambas instituciones encarnadas en las figuras de mis tutores, que siempre han resuelto las dudas que me iban surgiendo y me animaban a continuar en las diferentes líneas de desarrollo que se me ocurrían para el estudio. Esto se hace patente en temas como las temáticas elegidas para estudiar, pues analizar temas que me apasionan (como es la industria de los videojuegos o el fútbol) ha provocado que muchos días haya podido trabajar con más ilusión de lo que seguramente lo habría hecho en caso de haber escogido cualquier otro tema. Además, el haber sido a efectos prácticos el responsable último del desarrollo creo que también me ha ayudado mucho, pues he tenido que gestionar y organizar todas las tareas desde un inicio para poder adaptarme a los plazos marcados, lo cual tengo claro que resultará clave en mi futuro profesional.

Si bien no había cursado la asignatura de *Sistemas Inteligentes*, la cual es la más cercana a este trabajo al presentar los conceptos básicos de Aprendizaje Automático, también es cierto que no se me puede considerar un completo neófito en este campo, al haber desarrollado mis prácticas de empresa en esta rama. De hecho, dicha etapa ha sido la que me ha permitido adquirir las nociones fundamentales sobre las cuales he podido indagar más a lo largo de este trabajo. En concreto, me ha resultado muy interesante el trabajo que se realiza en *data science* y seguramente trate de proseguir mi formación en este campo, sobre todo usando técnicas de PNL, las cuales he visto que tienen una fuerte base matemática que creo entender mejor gracias a la formación que he adquirido en el grado de Matemáticas.

8.2. Trabajo futuro

En última instancia, se dejan algunos puntos sobre los que podría mejorarse el trabajo en futura iteraciones del mismo, así como ideas de desarrollo alternativas aprovechando todos los datos recabados durante el desarrollo del proyecto:

- **Realizar un estudio análogo aprovechando los datos recogidos por Twitter relativos a los clubes de la Premier League.** Como temática alternativa, podría abordarse este problema para aprovechar la gran cantidad de datos provenientes de Twitter relativos a dichos equipos y sus fichajes. Ese estudio alternativo podría abarcar otras ramas de *data science* distintas, como podría ser el propio análisis de rendimiento de los jugadores fichados antes y después de que cambiaseen de escuadra.

- **Obtener datos sociales provenientes de otras fuentes.** Como podrían ser Instagram, Reddit u otros medios sociales. Esto permitiría ampliar la variedad de la información obtenida obteniendo mejores resultados.
- **Implementar una base de datos no relacional para la gestión de la información recabada.** Debido al alcance reducido del proyecto y al producirse todo el desarrollo de manera local, se ha trabajado siempre con los propios datos en sí, teniendo siempre un par de copias de seguridad de los mismos. No obstante, en un proyecto más ambicioso podría recurrirse a una base de datos no relacional para almacenar las diferentes colecciones de información de las que se dispone.
- **Pulir la limpieza y el cribado de los datos.** A pesar de que ya se ha realizado esta tarea en varias ocasiones, siempre puede mejorarse el filtro realizado. Por ejemplo, podrían eliminarse los términos usados para las búsquedas en Twitter para así no obtener resultados redundantes.
- **Ajustar el número de temas de LDA a la cantidad ideal de éstos,** de la misma forma en que se hizo con el modelo BTM. De hecho, podrían aprovecharse dichos valores para ajustar la cantidad de tópicos a buscar.
- **Obtener resultados con mejor visualización para el modelo BTM.** Uno de los principales puntos débiles del estudio es la dificultad para consultar los resultados obtenidos por el modelo biterm, por lo que sería conveniente buscar otra librería que lo implemente y migrar el código a dichas funcionalidades.
- **Análisis de sentimientos aprovechando el uso de mayúsculas y emoticonos.** Algunas de las principales herramientas de comunicación en redes sociales es el uso de mayúsculas para enfatizar palabras o emoticonos para transmitir sentimientos. No obstante, ambos fueron cribados durante el proceso de limpieza, por lo que no se ha podido medir su impacto.
- **Recabar más datos para entrenar los clasificadores de sentimiento de reseñas.** Bastaría con recabar reseñas de diversos lanzamientos, al igual que hace Pellarolo en su trabajo [52].
- **Comparar los valores de subjetividad percibidos con TextBlob con los de polaridad detectados por VADER** pues estos últimos, al menos en el caso de críticas de usuarios, reflejaban de manera bastante fidedigna las puntuaciones otorgadas.

Parte III

Apéndices

Apéndice A

Manual de Instalación

En esta sección se muestran los pasos necesarios para emular el entorno de desarrollo en el que se ha llevado a cabo el proyecto, para así poder ejecutar el desarrollo final del mismo. El proceso se ha llevado a cabo en un ordenador que tiene Windows 10 como sistema operativo, aunque para otros sistemas los pasos a seguir son prácticamente equivalentes salvo ligeras modificaciones en función del sistema operativo que se use.

1. **Instalar la plataforma de desarrollo Anaconda.** Para ello, basta descargar y ejecutar el instalador proporcionado en [6], manteniendo las opciones por defecto que ofrece. **Anaconda** es una distribución gratuita y de código abierto de los lenguajes de programación Python y R, utilizada para ciencia de datos y Aprendizaje Automático. Proporciona un entorno completo y listo para usar que incluye una gran cantidad de paquetes, bibliotecas y herramientas que son utilizados comúnmente en el ámbito de la ciencia de datos, así como una interfaz intuitiva que facilita su manejo.
2. **Descargar todos los paquetes y librerías requeridos por los diversos cuadernos.** Una vez completado el paso previo, únicamente hay que buscar entre los programas disponibles *Anaconda Prompt* y ejecutarlo como **administrador**, pues a lo largo del proceso se desinstalarán algunos de los paquetes que incluye Anaconda por defecto. Al abrirlo, surgirá una consola de comandos desde la que se pueden modificar los distintos módulos descargados en el sistema. La descarga de nuevos paquetes se realiza mediante el mandato

```
pip install <nombre_del_paquete>
```

El listado A.1 muestra todos los comandos necesarios para realizar dicha descarga, por lo que basta copiar y ejecutar dichos mandatos uno por uno. El principal problema reside en instalar el módulo **bitermplus**, el cual exige disponer de una versión de Microsoft Visual Studio C++ igual o superior a la 14.0, para lo cual basta con seguir los pasos dados en [22].

```
$ pip install tweepy
$ pip install langdetect
$ pip install wordcloud
$ pip install pyLDAvis --user
$ pip install bitermplus
$ pip install tmplot
$ pip install tomotopy
$ pip install textblob
```

Código A.1: Comandos para instalar todos los paquetes requeridos por el proyecto

3. **Abrir la interfaz que da acceso a los cuadernos de desarrollo.** El acceso a los cuadernos de desarrollo se puede realizar de manera sencilla buscando el programa *Jupyter Notebook*. Al ejecutarlo, debe seleccionarse un navegador sobre el que acceder a la visualización de los documentos (durante el desarrollo se ha usado Google Chrome como opción predeterminada). Tras haber elegido esto, basta dirigirse a la ubicación donde esté almacenado el proyecto e ir ejecutando los cuadernos desde el inicio (el primero no funciona, pero como se guarda la imagen del núcleo pueden verse los resultados de cuando el cuaderno se ejecutó de manera exitosa), pues éstos incluyen hipervínculos para avanzar al siguiente de manera ordenada. De hecho, debido a que se conservan los resultados de la última ejecución realizada, puede consultarse los cuadernos tal cual están, pues además algunos de estos procesos requieren de un tiempo de ejecución bastante elevado. En caso de que quiera probarse en la propia máquina, la ejecución de cada una de las celdas se puede hacer gracias al botón *Run* (figura A.1).



Figura A.1: Cabecera de un archivo ejecutado por Jupyter Notebook

Apéndice B

Contenido adjunto

En esta sección se detallan los contenidos de los archivos adjuntos entregados a la par que la memoria del proyecto.

- **Cuadernos de desarrollo.** *Notebooks* de Jupyter en los que se han implementado las diversas fases del estudio. A petición del cliente final, están escritos en inglés. Son un conjunto de 7 cuadernos numerados que representan las diversas fases de las que ha constado el proyecto:

1. *RestAPI*. Extracción de datos provenientes de Twitter. Debido a los cambios de las políticas de acceso que ha sufrido la empresa en los últimos meses, **ya no puede ejecutarse** al no disponerse de una cuenta oficial que permita descargar datos de Twitter. Además, en caso de que pudiese ejecutarse tampoco se obtendrían los mismos datos, pues únicamente pueden recuperarse tweets con un plazo de hasta una semana de antigüedad. Se explica en la sección 4.1.
2. *WebScraping*. Extracción de datos provenientes de Metacritic. Abarca la sección 4.2 de este documento.
3. *DataStructure*. Cuaderno para visualizar y entender mejor la estructura de los datos con los que se trabaja. Se corresponde con la sección 4.3 de la memoria.
4. *DataCleaning*. Capítulo 5 de la memoria. Los datos obtenidos previamente se limpian y se convierten al formato que se usará en cuadernos posteriores.
5. *TopicModelling*. Análisis exploratorio de los datos y modelado de temas a partir de LDA (las dos primeras secciones del capítulo 6).
6. *Biterm*. Se corresponde con la sección 6.3, la que falta del capítulo relativo al *topic modeling*. Identificación de tópicos mediante el modelo BTM y la implementación bitermplus. No obstante, al tratar de hacerlo funcionar en un ordenador diferente al del empleado durante el desarrollo, la **visualización de modelos da errores** por culpa del módulo tmplot, debido a que el paquete bitermplus usa una versión anterior del mismo. Esto supone un motivo extra para migrar el desarrollo a otra implementación de BTM en futuras iteraciones del proyecto.

7. *SentimentAnalysis*. Detección del sentimiento asociado a los diferentes textos de los que se dispone, correspondiente al capítulo 7 de este documento.
- **data**. Conjunto de reseñas obtenidas de la página web Metacritic en el segundo cuaderno. Sigue la estructura de datos comentada en 4.3.2.
 - **Furbo**. Información extraída de Twitter gracias al primer cuaderno, en relación a los equipos pertenecientes al *Big Six* de Inglaterra, así como de sus fichajes más recientes.
 - **Modelos**. Modelos obtenidos en los cuadernos 5 y 6 para la extracción de temas (LDA y BTM), así como los clasificadores entrenados durante el último cuaderno para identificar el tipo de reseña según el texto de la misma. En el caso de LDA, también se incluyen los archivos de visualización interactivos html, a los cuales se puede acceder desde el propio cuaderno en que se generaron.
 - **Videojocs**. Información extraída de Twitter gracias al primer cuaderno, en relación a las tres principales compañías de la industria del videojuego, así como de sus lanzamientos a finales de enero de 2023. También se incluyen la versión cribada tanto de reseñas como tweets, generada mediante el cuarto cuaderno, así como el conjunto de tweets etiquetados durante el último cuaderno usando el modelo RoBERTa.
 - **Wordcloud**. Carpeta con las imágenes generadas durante el análisis exploratorio de datos (quinto cuaderno).

Bibliografía

- [5] Shikah J. Alsunaidi et al. “Applications of Big Data Analytics to Control COVID-19 Pandemic”. En: *Sensors* 21, 2282 (marzo 2021).
- [9] Francesco Barbieri et al. “TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification”. En: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, nov. de 2020, págs. 1644-1650. DOI: 10.18653/v1/2020.findings-emnlp.148. URL: <https://aclanthology.org/2020.findings-emnlp.148>.
- [11] Christian Beck. “Generalised information and entropy measures in physics”. En: *Contemp. Phys* 50 (4) (2009), págs. 495-510.
- [13] Steven Bird, Edward Loper y Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.
- [14] David M. Blei, Andrew Y. Ng y Michael I. Jordan. “Latent Dirichlet Allocation”. En: *Journal of Machine Learning Research* 3 993-1022 (2003).
- [15] Marco Bonzanini. *Mastering Social Media Mining with Python*. Packt, 2016.
- [17] Siddhartha Chatterjee y Michal Krystyanczuk. *Python Social Media Analytics*. Packt, 2017.
- [26] Anastasia Giachanou y Fabio Crestani. “Like it or not: A survey of Twitter sentiment analysis methods”. En: *ACM Computing Surveys* 49, 2, artículo 28 (junio 2016).
- [33] C.J. Hutto y Eric Gilbert. “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text”. En: *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI* Junio (2014).
- [35] Daniel Jurafsky y James H. Martin. *Speech and Language Processing*. Borrador (tercera versión), 2020.
- [36] Ilker Kocabas, bekir Taner Dinçer y Bahar Karaoglan. “Investigation of Luhn's claim on information retrieval”. En: *Turkish Journal of Electrical Engineering and Computer Sciences* 19, N°3 (2011).
- [37] Sergei Koltcov. “Application of Rényi and Tsallis entropies to topic modeling optimization”. En: *Physica A: Statistical Mechanics and its Applications* 512 (2018), págs. 1192-1204.

Bibliografía

- [44] Hugo A. Mitre-Hernández, Lemus-Olalde Cuauhtémoc y Edgar Ortega-Martínez. “Estimación y control de costos en métodos ágiles para desarrollo de software: un caso de estudio”. En: *Ingeniería Investigación y Tecnología XV* (número 3) (julio-septiembre 2014), págs. 403-418.
- [53] Jesús Cordobés Puertas. *Tema 2 - El ambiente externo*. 2023.
- [57] Ken Schwaber y Jeff Sutherland. *The Scrum Guide*. 2020.
- [77] Xiaohui Yan et al. “A Biterm Topic Model for Short Texts”. En: *Association for Computing Machinery* (2013).

Webgrafía

- [1] *About The Nielsen Company.* Nielsen. Accedido el 18/05/2023. URL: <https://web.archive.org/web/20090215003017/http://nielsen.com/about/index.html>.
- [2] *About us.* Nielsen. Accedido el 18/05/2023. URL: <https://www.nielsen.com/about-us/about/>.
- [3] *About us.* Nielsen IQ. Accedido el 18/05/2023. URL: <https://nielseniq.com/global/en/about-us/>.
- [4] María Fernanda Aguirre. *Realiza estimaciones ágiles y precisas gracias al Planning Poker.* appvizer. Accedido el 29/05/2023. URL: <https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/planning-poker>.
- [6] *Anaconda Distribution: Free Download.* Anaconda. Accedido el 15/11/2022. URL: <https://www.anaconda.com/download>.
- [7] Louie Andre. *53 Important Statistics About How Much Data Is Created Every Day.* Finances Online. Accedido el 19/05/2023. URL: <https://financesonline.com/how-much-data-is-created-every-day/>.
- [8] David Arroyo. *Hi-Fi Rush costó la mitad que Forspoken y ya le supera en ventas.* Meristation. Accedido el 29/06/2023. URL: <https://as.com/meristation/noticias/hi-fi-rush-costo-la-mitad-que-forspoken-y-ya-le-supera-en-ventas-n/>.
- [10] *Beautiful Soup Documentation.* Accedido el 19/03/2023. URL: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>.
- [12] *Big Data en el fútbol: El caso Brentford FC.* acadef. Accedido el 05/06/2023. URL: <https://www.acadef.es/big-data-en-el-futbol-el-caso-brentford-fc/>.
- [16] Jason Brownlee. *Difference Between a Batch and an Epoch in a Neural Network.* Machine Learning Mastery. Accedido el 15/11/2022. URL: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>.
- [18] Brian Dean. *How Many People Use Twitter in 2023?* backlinko. Accedido el 08/06/2023. URL: <https://backlinko.com/twitter-users#twitter-users>.
- [19] Clément Delteil. *Unsupervised Sentiment Analysis With Real-World Data: 500,000 Tweets on Elon Musk.* Towards AI. Accedido el 12/04/2023. URL: <https://pub.towardsai.net/unsupervised-sentiment-analysis-with-real-world-data-500-000-tweets-on-elon-musk-3f0653135558>.

Webgrafía

- [20] *Developer Platform Documentation.* Twitter. Accedido el 02/02/2023. URL: <https://developer.twitter.com/en/docs>.
- [21] Claire Drumond. *Guía de la metodología scrum: qué es, cómo funciona y cómo empezar.* TechCrunch. Accedido el 20/05/2023. URL: <https://www.atlassian.com/es/agile/scrum>.
- [22] *error: Microsoft Visual C++ 14.0 or greater is required.* Microsoft. Accedido el 06/05/2022. URL: <https://learn.microsoft.com/en-us/answers/questions/136595/error-microsoft-visual-c-14-0-or-greater-is-requir>.
- [23] Javier Escribano. *El rumor del lanzamiento de Advance Wars 1+2 en Switch esta semana es falso.* Hobby Consolas. Accedido el 13/04/2023. URL: <https://www.hobbyconsolas.com/noticias/rumor-lanzamiento-advance-wars-12-switch-semana-falso-1196730>.
- [24] Matija Ferjan. *Xbox Game Pass Subscribers: How Many Game Pass Subscribers are There in 2023?* Headphones addict. Accedido el 04/06/2023. URL: <https://headphonesaddict.com/xbox-game-pass-subscribers/>.
- [25] Jason Foster. *Data Skills Are Mission-Critical: How To Bridge The Skills Gap.* Forbes. Accedido el 04/06/2023. URL: <https://www.forbes.com/sites/forbesbusinesscouncil/2022/11/15/data-skills-are-mission-critical-how-to-bridge-the-skills-gap/>.
- [27] Enes Gokce. *NLP Capstone Project.* Accedido el 09/05/2023. URL: https://github.com/EnesGokceDS/Amazon_Reviews_NLP_Capstone_Project.
- [28] Enes Gokce. *Sentiment Analysis on Amazon Reviews.* Towards Data Science. Accedido el 09/05/2023. URL: <https://towardsdatascience.com/sentiment-analysis-on-amazon-reviews-45cd169447ac>.
- [29] David Gómez. *Social Media no traduce Redes Sociales.* bienpensado. Accedido el 19/05/2023. URL: <https://bienpensado.com/que-es-social-media-y-su-diferencia-con-las-redes-sociales/>.
- [30] Aimee Hart. *Fire Emblem Engage hits 1.61 million sales worldwide.* Gayming. Accedido el 29/06/2023. URL: <https://gaymingmag.com/2023/05/fire-emblem-engage-hits-1-61-million-sales-worldwide/>.
- [31] *How do you compare and contrast BERT with other deep learning approaches for sentiment analysis?* Linkedin. Accedido el 28/06/2023. URL: <https://www.linkedin.com/advice/0/how-do-you-compare-contrast-bert-other-deep-learning>.
- [32] Owen Hughes. *Employers are desperate for data scientists as demand booms.* ZD-NET. Accedido el 04/06/2023. URL: <https://www.zdnet.com/article/employers-are-desperate-for-data-scientists-as-demand-booms/>.
- [34] Manish Singh Ivan Mehta. *Twitter to end free access to its API in Elon Musk's latest monetization push.* TechCrunch. Accedido el 02/02/2023. URL: <https://techcrunch.com/2023/02/01/twitter-to-end-free-access-to-its-api/>.

- [38] Neville Lahiru. *Hi-Fi Rush Outperforms Forspoken in Steam Sales*. Gamerant. Accedido el 29/06/2023. URL: <https://gamerant.com/hi-fi-rush-forspoken-steam-sales-outperformed/>.
- [39] Alejandro Manzanares Loreto. *¿Cuál es la diferencia entre Data Science y Data Analytics?* Hack a boss. Accedido el 02/06/2023. URL: <https://www.hackaboss.com/blog/cual-es-la-diferencia-entre-data-science-y-data-analytics>.
- [40] Steven Loria. *TextBlob: Simplified Text Processing*. Accedido el 11/05/2023. URL: <https://textblob.readthedocs.io/en/dev/>.
- [41] Fran G. Matas. *Las 21 consolas más vendidas de la historia*. Vandal. Accedido el 04/06/2023. URL: <https://vandal.elespanol.com/reportaje/las-20-consolas-mas-vendidas-de-la-historia>.
- [42] Abby McCain. *26 STUNNING BIG DATA STATISTICS [2023]: MARKET SIZE, TRENDS, AND FACTS*. ZIPPIA. Accedido el 03/06/2023. URL: <https://www.zippia.com/advice/big-data-statistics/>.
- [43] Rachel Meltzer. *These Are the Top Industries Hiring Data Analysts Right Now*. CF Blog. Accedido el 04/06/2023. URL: <https://careerfoundry.com/en/blog/data-analytics/top-industries-hiring-data-professionals/>.
- [45] Andreas Mueller. *WordCloud for Python documentation*. Accedido el 12/04/2023. URL: https://amueller.github.io/word_cloud/.
- [46] Fran Méndez. *¿Cómo el Big Data ayudó a Obama a ganar?* Forbes. Accedido el 05/06/2023. URL: <https://forbes.es/start-ups/7560/como-el-big-data-ayudo-a-obama-a-ganar/>.
- [47] *NielsenIQ se convierte en una empresa independiente*. Business Wire. Accedido el 18/05/2023. URL: <https://www.businesswire.com/news/home/20210308005771/es>.
- [48] *NielsenIQ's alternatives and competitors*. CBInsights. Accedido el 03/06/2023. URL: <https://www.cbinsights.com/company/nielseniq/alternatives-competitors>.
- [49] *NLP Natural Language Processing : Introducción*. DataScientest. Accedido el 02/06/2023. URL: <https://datascientest.com/es/nlp-introduccion>.
- [50] Matthias Orgler. *What is the optimal sprint length in Scrum?* Hackernoon. Accedido el 22/05/2023. URL: <https://hackernoon.com/what-is-the-optimal-sprint-length-in-scrum-368e966f3243>.
- [51] Prasad Patil. *What is Exploratory Data Analysis?* Towards Data Science. Accedido el 16/06/2023. URL: <https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15>.
- [52] Martín Pellarolo. *metacritic*. Accedido el 17/03/2023. URL: <https://github.com/martinpella/metacritic>.
- [54] Radim Rehurek. *Gensim*. Accedido el 05/02/2023. URL: <https://pypi.org/project/gensim/#data>.

Webgrafía

- [55] Shahmeer Sarfaraz. *GTA Online Update 1.66 Fixes Major Exploit Causing Player Bans*. Exputer. Accedido el 13/04/2023. URL: <https://exputer.com/news/games/gta-online-update-1-66-exploit/>.
- [56] Bruce Schoenfeld. *El arma secreta del Liverpool: el análisis de datos*. The New York Times. Accedido el 05/06/2023. URL: <https://www.nytimes.com/es/2019/05/29/espanol/liverpool-champions.html>.
- [58] *Story Point based Cost Estimation*. G DATA. Accedido el 02/06/2023. URL: <https://www.gdatasoftware.com/blog/story-point-based-cost-estimation>.
- [59] Keith Stuart. *Interview: the science and art of Metacritic*. The Guardian. Accedido el 13/06/2023. URL: <https://www.theguardian.com/technology/gamesblog/2008/jan/17/interviewtheartofmetacriti>.
- [60] *Sueldos para el puesto de Analista en España*. glassdoor. Accedido el 02/06/2023. URL: https://www.glassdoor.es/Sueldos/analista-sueldo-SRCH_K00,8.htm?clickSource=searchBtn.
- [61] *Sueldos para el puesto de Data Scientist en España*. glassdoor. Accedido el 02/06/2023. URL: https://www.glassdoor.es/Sueldos/data-scientist-sueldo-SRCH_K00,14.htm?clickSource=searchBtn.
- [62] Paul Tassi. *'Horizon Forbidden West: Burning Shores' Shows Metacritic Must Curb Review Bombing*. Forbes. Accedido el 13/06/2023. URL: <https://www.forbes.com/sites/paultassi/2023/04/23/horizon-forbidden-west-burning-shores-shows-metacritic-must-curb-review-bombing/>.
- [63] Maksim Terpilowski. *Bitermplus*. Accedido el 09/05/2023. URL: <https://bitermplus.readthedocs.io>.
- [64] Maksim Terpilowski. *Tmplot*. Accedido el 10/05/2023. URL: <https://pypi.org/project/tmplot/>.
- [65] Mintu Tomar. *Xbox Smash Hit Hi-Fi Rush Proves Having a \$70 Price Tag Is Not the Winning Formula for Modern Games*. Essentially Sports. Accedido el 29/06/2023. URL: <https://www.esSENTIALLYsports.com/esports-news-xbox-smash-hit-hi-fi-rush-proves-having-a-70-price-tag-is-not-the-winning-formula-for-modern-games/>.
- [66] John Tones. *El cierre del estudio de 'Forspoken' deja claro algo: la industria actual del videojuego no perdona errores*. Xataka. Accedido el 29/06/2023. URL: <https://www.xataka.com/videojuegos/cierre-estudio-responsable-forspoken-deja-clara-cosa-industria-actual-videojuego-no-perdona-errores>.
- [67] John Tones. *Microsoft compra Bethesda por 7.500 millones de dólares y se queda con franquicias como 'DOOM', 'Fallout' o 'Wolfenstein'*. Xataka. Accedido el 04/06/2023. URL: <https://www.xataka.com/videojuegos/microsoft-da-empujon-a-su-cartera-exclusivos-compra-bethesda-editora-franquicias-como-doom-fallout-wolfenstein>.

- [68] Markus Tretzmüller. *Biterm*. Accedido el 08/05/2023. URL: <https://pypi.org/project/biterm/>.
- [69] *Tweepy Documentation*. Accedido el 02/02/2023. URL: <https://docs.tweepy.org/en/stable>.
- [70] *Twitter API Rate Limits*. Twitter. Accedido el 02/02/2023. URL: <https://developer.twitter.com/en/docs/twitter-api/rate-limits>.
- [71] *Twitter API (subscriptions plans)*. Twitter. Accedido el 02/06/2023. URL: <https://developer.twitter.com/en/products/twitter-api>.
- [72] *Twitter API v2 data dictionary*. Twitter. Accedido el 02/02/2023. URL: <https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model/tweet>.
- [73] Rebekah Valentine. *Phil Spencer: Game Pass leads to more game sales*. Games Industry. Accedido el 29/06/2023. URL: <https://www.gamesindustry.biz/phil-spencer-game-pass-leads-to-more-game-sales>.
- [74] Duong Vu. *Generating WordClouds in Python Tutorial*. Datacamp. Accedido el 12/04/2023. URL: <https://www.datacamp.com/tutorial/wordcloud-python>.
- [75] *What is exploratory data analysis?* IBM. Accedido el 16/06/2023. URL: <https://www.ibm.com/topics/exploratory-data-analysis>.
- [76] Leah J. Williams. *Square Enix says Forspoken sales were ‘lacklustre’*. GAMES hub. Accedido el 29/06/2023. URL: <https://www.gameshub.com/news/news/square-enix-forspoken-sales-lacklustre-2609303/>.
- [78] *¿Cuál es el coste de la empresa al contratar a un trabajador?* KENJO Blog. Accedido el 02/06/2023. URL: <https://blog.kenjo.io/es/cual-es-el-coste-de-la-empresa-al-contratar-a-un-trabajador>.