Universidad de Valladolid

Escuela de Ingeniería Informática de Valladolid

Máster en Ingeniería Informática - Especialidad Big Data

Trabajo Fin de Máster

Análisis y Visualización Big Data en Eventos Deportivos

Autor:
Álvaro José Monedero Carreras

Tutores:
Aníbal Bregón Bregón
Miguel Ángel Martínez Prieto

“He who makes a beast of himself gets rid of the pain of being a man”

Avenged Sevenfold, Samuel Johnson

Agradecimientos

A mi familia, amigos y compañeros del trabajo por todo el apoyo y ánimo recibido durante esta etapa. Gracias por supuesto a mis tutores por todo el tiempo dedicado.

Resumen

Hoy en día, la utilización de las tecnologías que componen el mundo Big Data ha aumentado de forma exponencial debido a la búsqueda de agilidad e innovación por parte de las grandes empresas. Estas tecnologías proporcionan una gran capacidad de almacenamiento y procesamiento de la información, pero para poder sacar partido de los datos es necesario disponer de una correcta herramienta de visualización.

De esta forma, en este proyecto se pretenden aunar ambos escenarios, desarrollando una herramienta de extracción de información y estadísticas de resultados deportivos a través de *web scraping*, su posterior transformación y almacenamiento en base de datos, y por último, disponer de diferentes cuadros de mando desde los que poder explotar la información almacenada e indexada de una forma visual.

Con el desarrollo de este proyecto se extraerá y almacenará la información de partidos y clasificaciones de la Liga de Fútbol Profesional de España a lo largo de multitud de temporadas, permitiendo al usuario final realizar el análisis y la visualización de la información recogida.

Abstract

Nowadays, the use of Big Data technologies has grown considerably due to the search for agility and innovation by business companies. Big Data technologies provide large storage and processing capacity but, without a good data visualization strategy, all these capacities are worthless for business.

Once stablished the importance of data visualization, the aim this project is the combination of the use of Big Data technologies for the extraction, transformation and loading of data, and finally the use of a correct visualization strategy for data indexation and visualization of different data dashboards.

With de development of this project, we will get the information about matches and standings of different La Liga seasons from early 2000s until now for matches and from early 90s until now and store it. Finally, one of the main goals of the project is to highlight the importance of stablishing a good visualization strategy, so data will be indexed in a search engine and several dashboards will be created to analyze the stored information.

Índice de contenidos

Índice de figuras

Índice de tablas

1 Introducción

En los últimos años, el interés de la población por las estadísticas deportivas ha crecido considerablemente. Este aumento en la búsqueda de estadísticas deportivas puede verse reflejado en la evolución del crecimiento en apuestas deportivas tanto a nivel nacional como a nivel internacional.

De hecho, realizando una simple búsqueda en la web del término "estadísticas deportivas", podemos observar como todos los resultados destacados tienen como fin la realización de apuestas deportivas. En la *Figura 1* expuesta a continuación, podemos comprobar cuáles son los resultados de esta búsqueda:

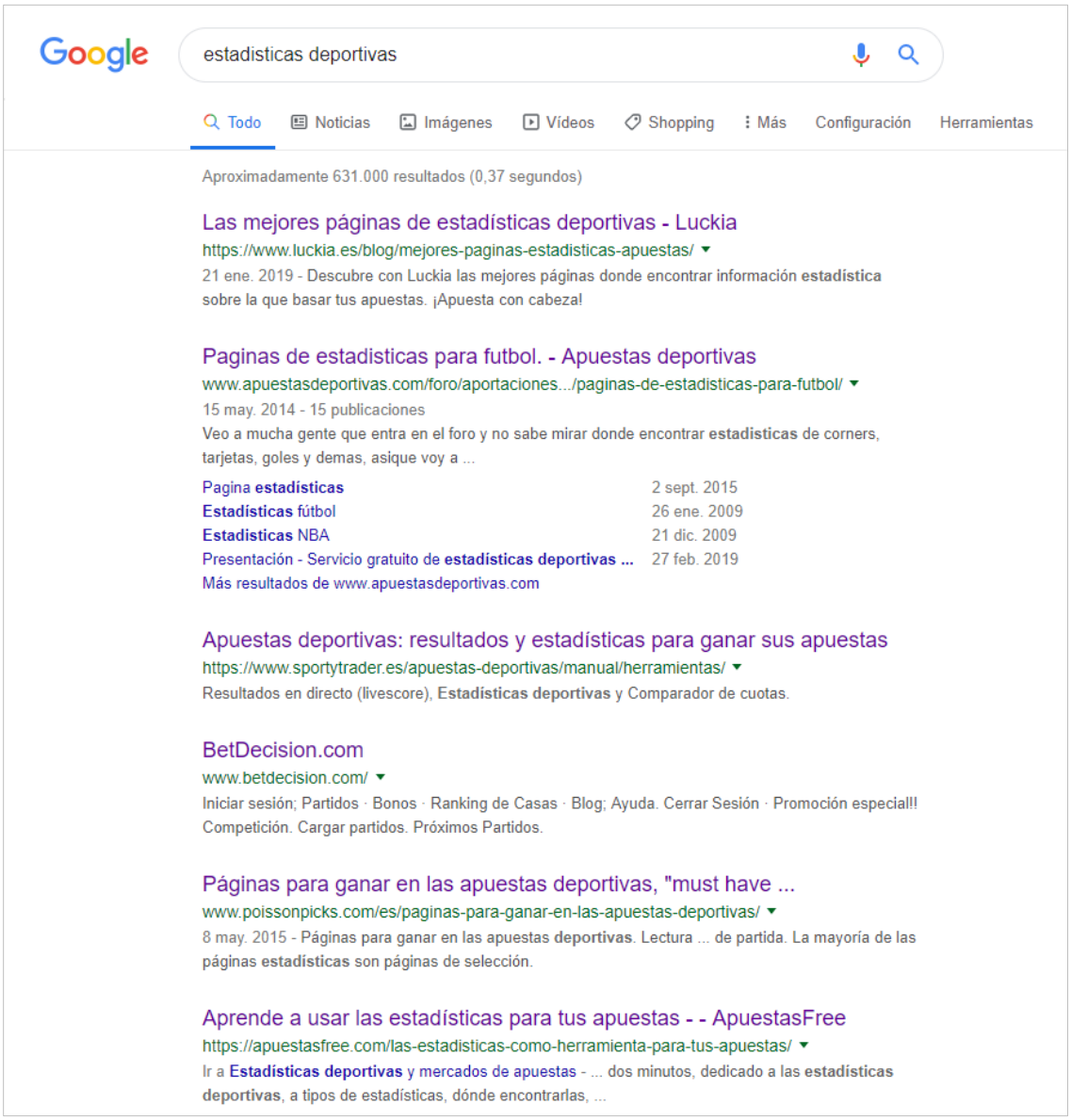


Figura 1.- Resultados web del término "estadísticas deportivas"

Este crecimiento no solo se ve reflejado en la web, si no que actualmente en los espacios publicitarios de televisión durante eventos deportivos, e incluso durante la programación normal, aparecen numerosos anuncios relacionados con casas de apuestas y eventos deportivos.

El crecimiento de las apuestas deportivas en España ha sido uno de los fenómenos más destacados del impulso de los negocios online durante los últimos años. El número de usuarios de estos portales cada vez es mayor, situando nuestro país como uno de los referentes del juego en línea en la Unión Europea. Si tenemos en cuenta cifras monetarias específicas, el juego en línea mueve en España más de trece 13.000 millones de euros con un aumento del 10% anual. Aunque las apuestas deportivas todavía no han alcanzado a los casinos online en cuanto a porcentajes de dinero jugado, se espera que en uno o dos años se conviertan en la principal forma de ocio online.

En lo que a número de jugadores se refiere, el número de jugadores habituales en 2018 en España superó los 800.000 alcanzando un aumento del 30% con respecto al año anterior. Estos datos confirman el aumento en la demanda en el sector y su impulso inminente en el mercado.

Es de gran importancia destacar que el uso sin control de la realización de apuestas deportivas puede acarrear problemas de ludopatía muy serios y que requieren de tratamiento psicológico en multitud de casos. Si bien, estas apuestas deportivas pueden ser consideradas como un problema para la sociedad actual, también suponen una oportunidad de negocio sobre un mercado que se encuentra en evidente auge.

1.1 Motivación

Una vez expuesto el interés actual en el análisis de resultados deportivos, pasamos a situar el desarrollo de este Trabajo Fin de Máster en un contexto real que nos permita justificar de forma práctica las diferentes aplicaciones de la herramienta.

El objetivo de este proyecto es la creación de un proceso de extracción web de resultados deportivos utilizando técnicas *web scraping* y su posterior transformación y carga en una base de datos capaz de soportar el procesamiento de grandes cantidades de datos. Una vez almacenada la información, ésta será indexada para posteriormente crear diversos cuadros de mando (*dashboards*) desde los que visualizar dicha información estadística.

Actualmente, podemos encontrar multitud de sitios web en internet que proporcionan información de resultados y estadísticas deportivas en directo para multitud de deportes diferentes (fútbol, tenis, baloncesto, hockey…). Obviamente, estas aplicaciones no están pensadas para facilitar el filtrado de la información por parte de los usuarios, si no que la información estadística se encuentra asociada a un evento concreto.

Un ejemplo práctico de esta limitación en la forma en la que se presenta la información sería el siguiente:

- Disponemos de la información para el partido *Real Valladolid – F.C. Barcelona*. Para dicho partido contamos con la información de goles, faltas, tarjetas, saques de esquina…

- El problema viene a la hora de querer comparar o agregar la información, puede ser muy interesante para el análisis disponer de toda la información estadística anterior agregada a nivel de temporada y además compararla con la información recogida durante temporadas pasadas.

Existen otro tipo de sitios web que son capaces de proporcionar algunas estadísticas de forma agregada y que permiten establecer diferentes filtros (equipo, jugadores, competición…) en la información. En este caso nos encontramos con una nueva limitación, ya que estos sitios web muestran la información filtrada en formato de tabla, lo que reduce la interpretación por parte del usuario.

De esta forma, la motivación para el desarrollo de este Trabajo Fin de Máster es la de dar solución a las limitaciones expuestas anteriormente y crear una herramienta capaz de recoger toda la información estadística relacionada a los partidos y clasificaciones deportivas de forma centralizada, y proveer de un sistema de visualización para estos datos ya que, sin una correcta visualización de la información, ésta no sirve de utilidad.

Además, otro punto muy a tener en cuenta en la motivación de este trabajo es la utilización de herramientas Big Data en el mundo del deporte profesional. Por ejemplo, durante un partido de fútbol se generan en torno a 8 millones de eventos y datos, lo que supone la necesidad de disponer de profesionales especializados en el manejo de grandes volúmenes de datos. Este va a ser un aspecto fundamental en el desarrollo del proyecto, ya que se hará uso de estas tecnologías Big Data durante prácticamente todo el desarrollo del trabajo, desde el almacenamiento a la indexación y la visualización de la información.

1.2 Objetivos del proyecto

El objetivo general de este proyecto es la realización de un proceso de extracción, transformación y carga (ETL) de información, junto con su posterior etapa de visualización, parte fundamental en cualquier proceso de análisis de datos.

Como objeto de análisis se ha escogido el ámbito deportivo, concretamente La Primera División de Fútbol de España. Es importante destacar que una vez completado el desarrollo del proyecto, éste será fácilmente adaptable a otras competiciones futbolísticas e incluso a otros deportes, lo que implicaría el procesamiento y almacenamiento de grandes cantidades de información.

El flujo de proceso que se ha seguido para el desarrollo del proyecto es el indicado en la *Figura 2*:

Figura 2.- Flujo de proceso del proyecto

- **Extracción de datos (web scraping)**

En esta fase se realiza la conexión con la url desde la que se va a obtener la información y, a través de la utilización de técnicas de web scraping se simula la actividad de navegación que realizaría un usuario por la web, almacenando la información en diferentes variables de forma que pueda ser transformada antes de su almacenamiento en base de datos. Así, se recogerá toda la información proveniente de las clasificaciones de la competición, y toda la información y estadísticas generadas a lo largo de un partido. Esta fase se expondrá de forma más exhaustiva en secciones posteriores.

- **Transformación de datos**

En el mismo momento en el que la información es recogida es transformada, eliminando datos que no son de interés, publicidad, codificación HTML… para ser almacenada en estructuras de datos. Este planteamiento permite contar con un tiempo de ejecución más eficiente que si realizásemos la transformación una vez almacenada en el data warehouse.

Una vez transformada, toda la información relevante se estructura en diccionarios Python que posteriormente serán ingestados en nuestro servidor MongoDB. Esta fase se expondrá de forma más exhaustiva en secciones posteriores.

- **Ingesta de datos**

Tras haber tratado, limpiado y estructurado la información que queremos almacenar, desde Python se establece la conexión con la base de datos y la colección MongoDB en la que queremos almacenar la información. Tal y como se ha expuesto anteriormente, MongoDB es una base de datos documental cuya estructura de documentos es equivalente a la de un diccionario Python, lo que permite una perfecta integración entre ambas tecnologías. Expondremos esta fase de forma más exhaustiva en secciones posteriores.

- **Indexación de datos**

Durante la fase de indexación el proceso realizado es el de migrar toda la información documental almacenada en MongoDB al motor de búsqueda ElasticSearch. Para realizar este proceso se utiliza la herramienta Apache Nifi descrita anteriormente. Desde Apache Nifi tendremos que definir la conexión con la colección de documentos que deseemos, aplicar una pequeña transformación y unir el flujo de datos con el índice de ElasticSearch donde deseamos almacenar la información. Al igual que en las fases anteriores, expondremos este flujo de trabajo con más detalle en secciones posteriores.

- **Visualización**

Por último, una vez disponemos de toda la información replicada e indexada en ElasticSearch, utilizaremos Kibana para acceder y representar la información en forma de visualización. Kibana es un software de visualización integrado dentro de Elastic, por lo que es un software extremadamente eficiente comparado con otras herramientas de visualización que podemos encontrar en el mercado.

El objetivo en esta fase es la creación de diferentes métricas, filtros, gráficos y cuadros de mando donde se pueda explotar toda la información que ha sido recogida. Esta fase se expondrá con más detalle en secciones posteriores.

1.3 Tecnologías utilizadas

Una vez expuestas la motivación en el desarrollo del proyecto y los objetivos principales del mismo, pasamos a definir las diferentes tecnologías que han sido utilizadas para el desarrollo de este Trabajo Fin de Máster:

Figura 3.- Tecnologías utilizadas

- **Python 3.7.3 -** https://www.python.org/

Python, en su versión 3.7.3, ha sido utilizado como lenguaje de programación para implementar la ingesta y la transformación de datos en este proyecto. A través de Python se realiza la conexión con la web para que, mediante técnicas de *web scraping*, los datos que deseamos obtener sean recogidos y posteriormente almacenados en base de datos.

Python es un lenguaje de programación interpretado que se encuentra enfocado hacia la legibilidad del código. La ventaja principal de Python sobre otros lenguajes de programación, *Java* o *C++*, es que es posible desarrollar código de una forma más rápida e intuitiva, mejorando la productividad y reduciendo el número total de líneas de código en la implementación. Otro aspecto a tener en cuenta es la portabilidad del lenguaje, permitiendo una correcta ejecución en máquinas Mac, Linux o Windows. Por último, una de las principales ventajas de Python es la comunidad que lleva consigo, lo que implica un alto grado de cuidado del lenguaje y una democratización en el desarrollo y la aplicación de versiones, así como el gran número de sitios web que proporcionan ayuda sobre errores, configuración y guías para el desarrollo.

Si nos fijamos en la evolución de este lenguaje a lo largo del tiempo, podemos ver que su popularidad ha ido siempre en aumento, siendo utilizado para el desarrollo de aplicaciones web, aplicaciones de escritorio, servidores de red, inteligencia artificial… Al principio, la popularidad de Python fue creciendo debido a su facilidad de aprendizaje y a la simplicidad del código,, pero posteriormente este crecimiento se disparó gracias a la explosión de campos como la inteligencia artificial, el aprendizaje automático o el análisis de datos.

Otra de las principales ventajas de Python es el amplio abanico de APIs y bibliotecas que podemos encontrar en la red. En este aspecto, la biblioteca que ha presentado un mayor crecimiento ha sido *Pandas,* usada para el tratamiento de datos. A continuación, en la *Figura 4,* podemos ver un gráfico con la evolución en la popularidad de las principales bibliotecas del lenguaje (Pandas, Django, Numpy, Matplotlib y Flask):

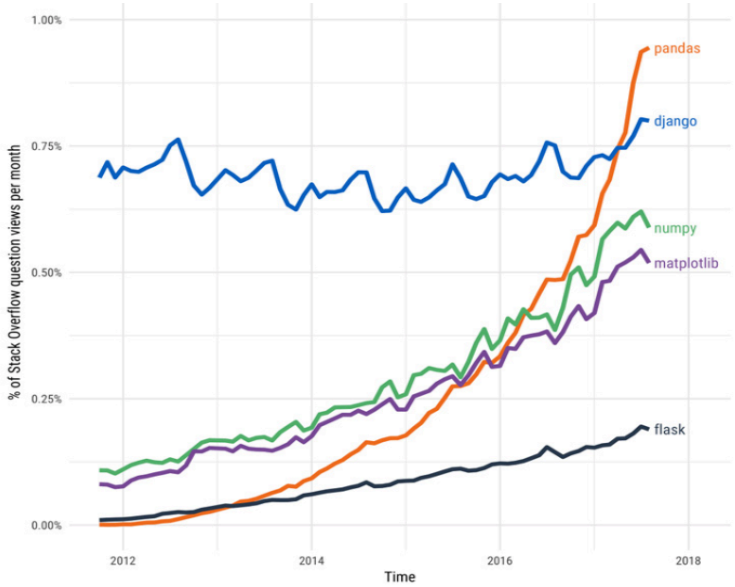


Figura 4.- Evolución en la popularidad de las bibliotecas en función de preguntas en Stack Overflow

La extracción de la información del proyecto se realizará mediante técnicas de web scraping. Web scraping es una técnica utilizada para acceder a información de sitios web. Esta técnica consiste en simular la navegación de un usuario real por la web a través del código fuente de dichos sitios web. En lo que se refiere a las técnicas de web scraping que podemos encontrar actualmente, podemos destacar el framework Scrapy y las bibliotecas Urllib, Requests, Selenium, BeautifulSoup y LXML. Para el desarrollo de este proyecto la biblioteca que se ha utilizado para realizar el web scraping ha sido BeautifulSoup.

Otro aspecto que destacar de este lenguaje durante el desarrollo del proyecto es la facilidad en la conexión, en este caso, con MongoDB. A través de muy pocas líneas de código se establece la conexión con la colección de datos en la que deseamos almacenar la información. Una vez establecida la conexión la inserción de documentos cuenta con un tiempo de inserción muy bajo.

- **MongoDB Community 4.0.8 -** https://www.mongodb.com/

MongoDB, en su versión 4.0.8, ha sido utilizado como el principal almacén de datos para el proyecto. Una vez se ha recogido y transformado la información de la web, desde el programa Python se establece la conexión con MongoDB para insertar los diccionarios de datos que han sido generados.

El principal motivo por el cual se ha seleccionado esta tecnología como almacén de datos es la forma en la que se guarda la información. MongoDB es una base de datos documental, es decir, en lugar de almacenar datos en tablas como se ha venido haciendo tradicionalmente, almacena la información en forma de documentos (*BSON* – Binary JSON). Esta forma de estructurar la información en documentos es prácticamente equivalente a los diccionarios de Python, por lo que suponía la sinergia perfecta entre la fase de extracción y transformación y la fase de ingesta en el almacén de datos.

MongoDB se define en su sitio web como la base de datos más productiva del mercado, destacando la multitud de objetos de uso diferentes, la estructuración de la información en documentos y su carácter distribuido pensado tanto para el desarrollo de aplicaciones como para trabajar desde la nube. Esta base de datos distribuida cuenta con el apoyo de varias de las principales compañías en el mercado como Google, Facebook, Electronic Arts, Ebay…

La estructuración de la información en documentos similares a JSON supone un cambio en el paradigma de las bases de datos, dejando de lado las filas y las columnas en las tablas, y siendo esta forma mucho más expresiva y potente que el modelo tradicional. Estas estructuras soportan la inclusión de arrays y de objetos anidados como valores, uno de los aspectos claves para la realización de este proyecto debido a la forma en la que se recoge la información (diccionarios Python). A su vez, también permite esquemas dinámicos en los que no todos los documentos tienen por qué tener el mismo número de campos.

Otro de los aspectos a tener en cuenta es la potencia en sus consultas, permitiendo filtrar y ordenar por cualquier campo del documento, sin importar lo anidado que dicho campo se encuentre. También permite la realización de agregaciones y distintos tipos de búsquedas (*geo-based search*, *graph search* & *text search).*

Por último, MongoDB permite toda la funcionalidad de las bases de datos relacionales tradicionales como transacciones ACID y soporte para la realización de joins en consultas, un aspecto muy a tener en cuenta a la hora de escoger una base de datos no relacional.

- **Robo 3T 1.2 -** https://robomongo.org/

Robo 3T, en su versión 1.2, es un software gratuito que proporciona una interfaz gráfica que permite la conexión y la administración de bases de datos MongoDB. Es muy útil para leer la información almacenada en las colecciones de la base datos, así como para crear y eliminar bases de datos, colecciones e incluso documentos.

En la *Figura 5* se muestra una imagen con la interfaz gráfica de la aplicación Robo 3T, donde podemos observar las diferentes bases de datos instanciadas para la conexión, las diferentes colecciones asociadas a dicha base de datos y los documentos de los que se compone.

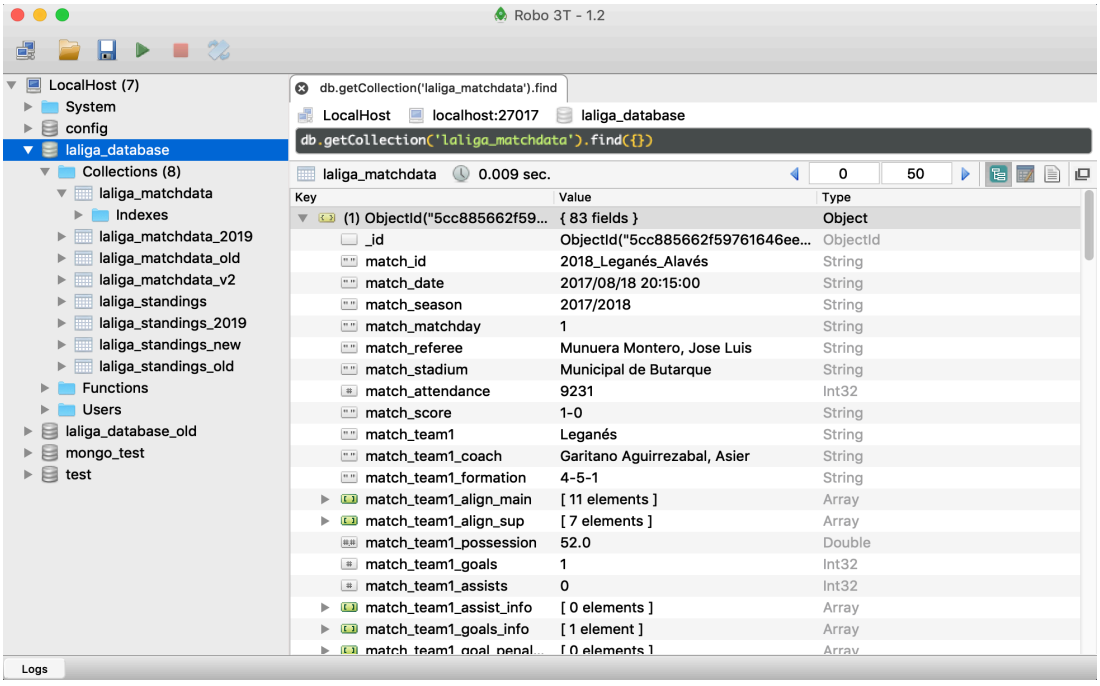


Figura 5.- Interfaz gráfica Robo 3T

- **Apache Nifi 1.9.1 -** https://nifi.apache.org/

Apache Nifi, en su versión 1.9.1, se ha utilizado en este proyecto para establecer un flujo continuo de información entre el almacén de datos (MongoDB) y el indexador de datos para su posterior visualización (ElasticSearch).

Apache Nifi es una plataforma integrada de procesamiento y logística de datos en tiempo real, permitiendo el movimiento de datos entre diferentes sistemas de forma rápida, sencilla y segura. De esta forma, este software es capaz de recoger datos desde diferentes fuentes, procesarlos y transformarlos y volcarlos posteriormente en otra fuente.

La principal ventaja de Apache Nifi es su simplicidad, ya que dispone de una interfaz web desde la que podemos diseñar de forma intuitiva y configurar de forma visual el flujo de datos. También nos permite iniciar y parar el proceso de carga de forma manual, así como establecer unos parámetros de inicio y fin. Otra característica de la que dispone Apache Nifi es la posibilidad de monitorizar el estado del flujo, pudiendo establecer diferentes ficheros para recoger posibles logs y errores en el proceso.

Por último, otra característica importante es que permite la ejecución en paralelo de múltiples instancias de Nifi, lo que puede ser de gran utilidad en caso de disponer de diferentes procesos ETL para una misma aplicación.

En la *Figura 6*, podemos ver una imagen mostrando como se vería un flujo de datos en la interfaz de Apache Nifi:

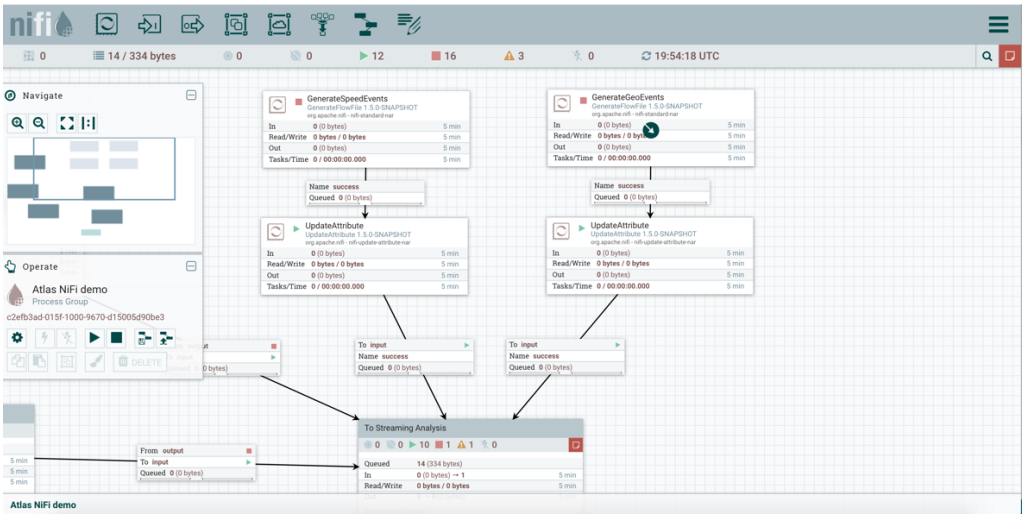


Figura 6.- Interfaz gráfica Apache Nifi

- **ElasticSearch 6.7.0 -** https://www.elastic.co/

ElasticSearch es un motor de búsqueda y análisis de código abierto, distribuido y de tipo REST que ha sido desarrollada con el fin de realizar búsquedas con bajos tiempos de respuesta para grandes cantidades de datos, la versión de la herramienta que se ha utilizado es la 6.7.0.

Las características que mejor definen el propósito de ElasticSearch son la distributividad y la escalabilidad implementando alta disponibilidad, permitiendo realizar búsquedas extremadamente rápidas que respalden las aplicaciones de análisis de datos. Además de las dos características expuestas anteriormente podemos destacar:

- Orientación a documentos. Utiliza JSON para indexar la información.
- No utiliza esquemas, aunque pueden definirse si es necesario.
- Dispone de una API REST muy completa que permite explotar prácticamente todas sus funcionalidades.
- Permite realizar búsquedas tanto estructuradas como no estructuradas.

Una vez se han expuesto las diferentes características de la herramienta, las principales ventajas con las que cuenta son:

- Rapidez. Mediante el uso de índices invertidos distribuidos, ElasticSearch consigue encontrar las mejores coincidencias para la búsqueda de texto incluso en conjuntos de datos muy grandes.
- API sencilla. ElasticSearch dispone de una API muy completa junto a una interfaz HTTP simple y la utilización de documentos JSON sin esquemas, facilitando todo el proceso.
- Integración con Kibana y Logtash. ElasticSearch viene integrado con estas dos herramientas, ampliando considerablemente su zona de influencia proporcionando funcionalidades de visualización y transformación.
- Actualizaciones de índice en tiempo real. Agregar un nuevo documento al índice de la herramienta puede tardar menos de un segundo, lo que proporciona a las aplicaciones un tiempo de respuesta muy rápido.
- Soporte para diferentes lenguajes de desarrollo. Hay una amplia lista de lenguajes cliente para los desarrolladores que quieran incluir ElasticSearch como Java, PHP, Python, JavaScript, Node.js…

- **Kaizen 2.71.69 -** https://www.elastic-kaizen.com/

Al igual que se ha utilizado Robo 3T como interfaz gráfica para realizar ciertas tareas en MongoDB, el software Kaizen, en su versión 2.71.69, ha sido utilizado como interfaz gráfica para ElasticSearch. Una de las principales ventajas de la herramienta es que es multiplataforma, permitiendo su ejecución tanto en Mac, como Linux, como Windows.

Esta aplicación cuenta con las siguientes características:

- Exploración de los documentos indexados (*Figura 7*).
- Administración de índices y alias.
- Conexión múltiple a diferentes servidores.
- Realización de tareas de backup y restauración de índices.
- Ejecución de consultas REST de forma directa (*Figura 8*).
- Consulta de estadísticas en el clúster de datos (*Figura 9*).

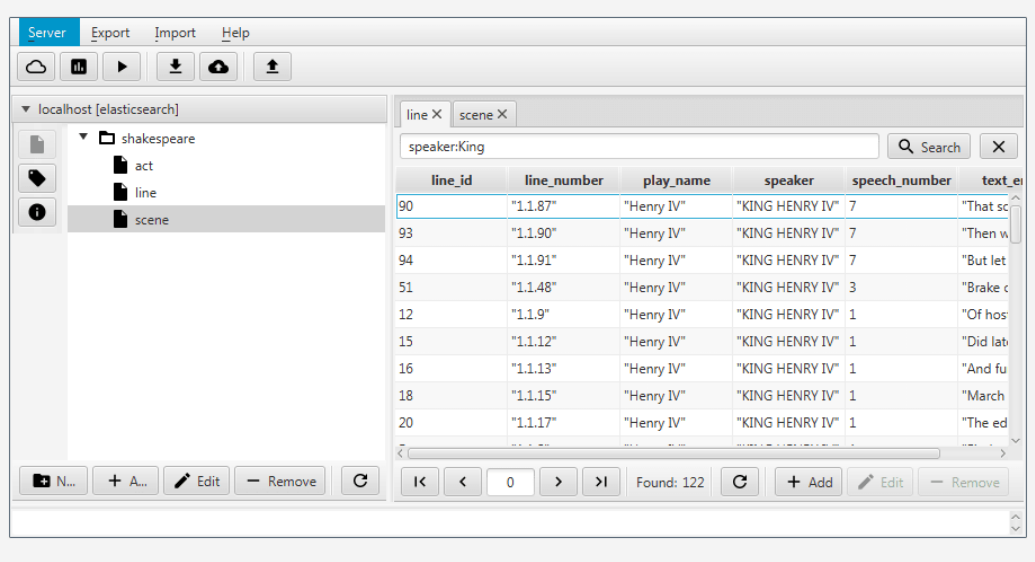


Figura 7.- Interfaz gráfica Kaizen - Exploración de documentos

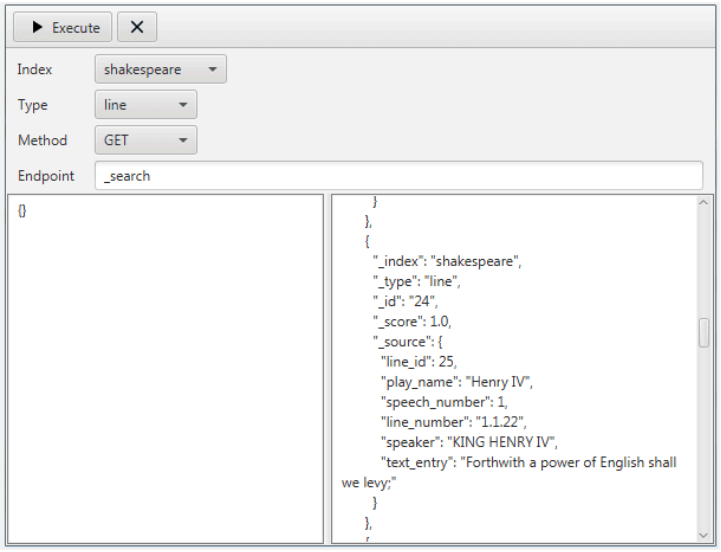


Figura 8.- Interfaz gráfica Kaizen - Ejecución de consultas REST

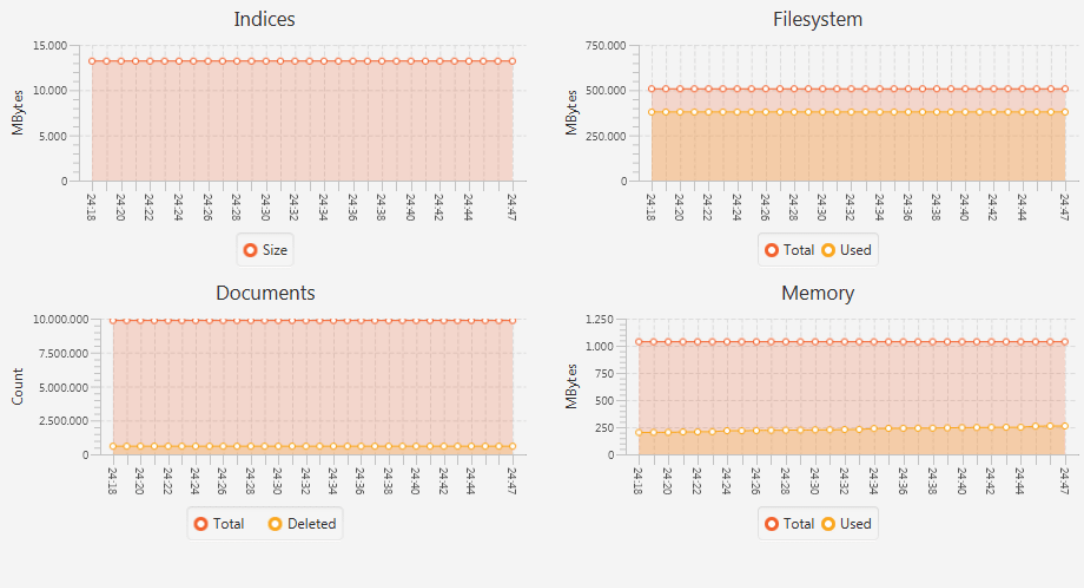


Figura 9.- Interfaz gráfica Kaizen - Estadísticas del clúster

- **Kibana 6.7.0 -** https://www.elastic.co/products/kibana

Por último, Kibana, en su versión 6.7.0, es la herramienta con la que se ha desarrollado la parte de visualización de este proyecto. Kibana permite visualizar los documentos indexados en ElasticSearch y navegar por la información que proporcionan. La herramienta incluye multitud de los gráficos frecuentemente más utilizados en visualización (histogramas, gráficos de líneas, gráficos de tarta…), pero a parte de estas funcionalidades Kibana ofrece otras funciones más avanzadas:

- Representación de información geoespacial (*Figura 10*).

Figura 10.- Kibana - Representación geoespacial

- Análisis de series temporales (*Figura 11*).

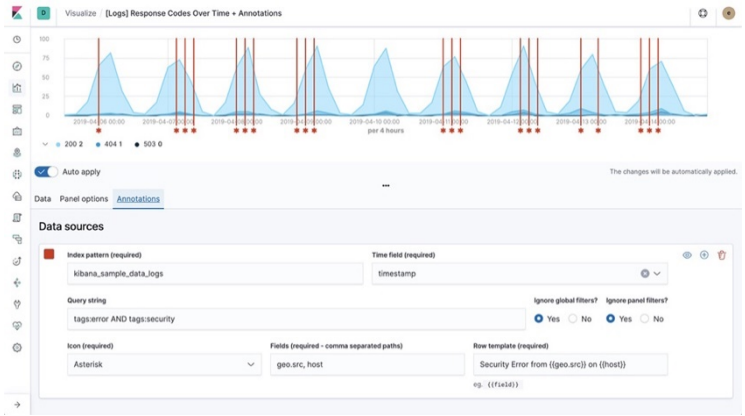


Figura 11.- Kibana - Series temporales

- Análisis de relaciones mediante grafos (*Figura 12*).

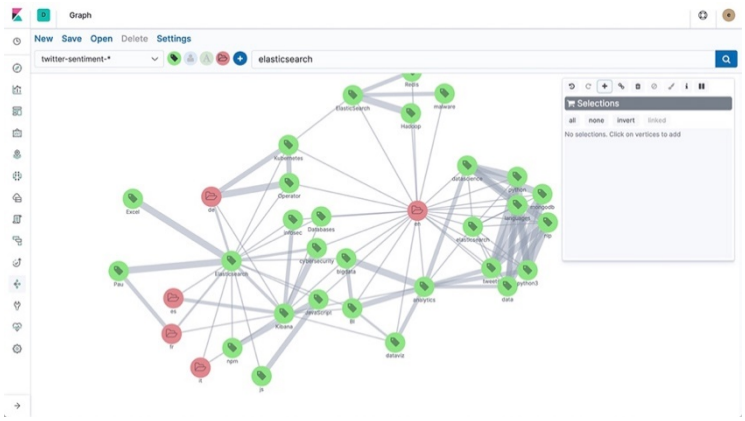


Figura 12.- Kibana - Relación mediante grafos

- Exploración de anomalías mediante aprendizaje automatizado (*Figura 13*).

Figura 13.- Kibana - Exploración de anomalías

- Personalización de gráficos y estilos a través de Canvas y CSS (*Figura 14*).

Figura 14.- Kibana – Personalización

- Herramientas de desarrollo:
  - *Consola* para realizar consultas directamente con ElasticSearch (*Figura 15*).

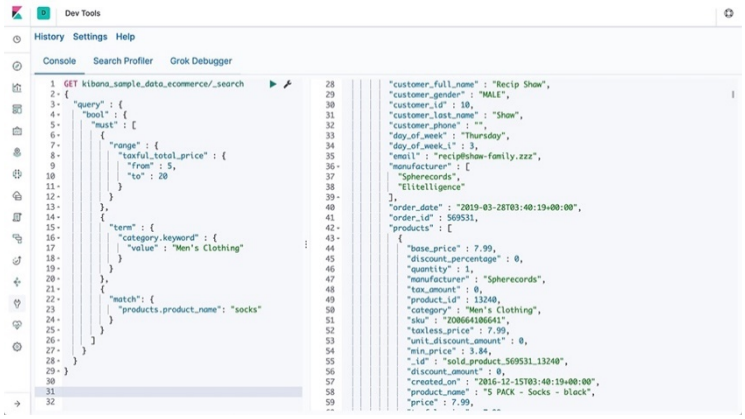


Figura 15.- Kibana - Consola

  - *Search Profiler* para comprobar en qué parte del procesamiento es donde más se ralentiza cada consulta (*Figura 16*).

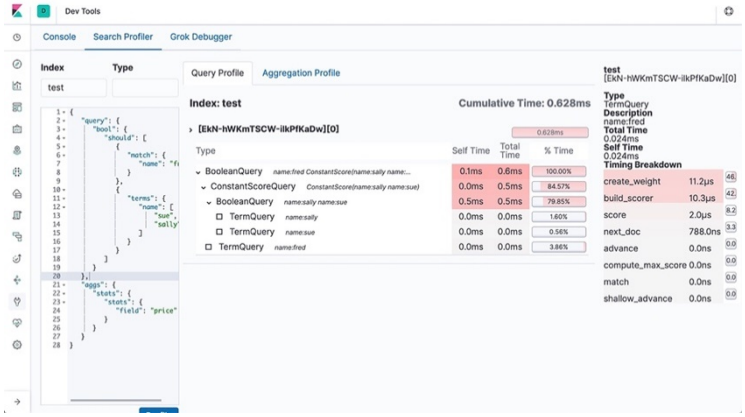


Figura 16.- Kibana - Search Profiler

  - *Grok Debugger* para la creación de patrones complejos de transformación con Logstash (*Figura 17*).

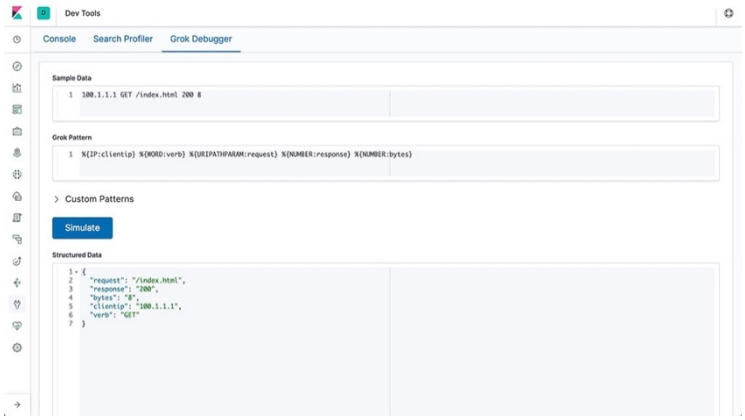


Figura 17.- Kibana - Grok Debugger

2 Estado del arte

A lo largo de este capítulo se hará un recorrido por la evolución temporal y la situación actual de las tecnologías Big Data, se expondrá la necesidad de disponer de una correcta estrategia de visualización en cualquier proceso de análisis de datos, y, por último, se expondrán los principales sitios web que han sido identificados como *generadores* de información deportiva.

2.1 Big Data

Actualmente, la principal medida para el éxito en las empresas ya no la estabilidad en los procesos de información, si no que aparecen nuevos retos a cumplir, como la agilidad y la innovación, para poder mantener un flujo de evolución constante. La respuesta al cumplimiento de estos objetivos son las tecnologías Big Data.

El término Big Data se refiere a grandes volúmenes de información que pueden ser *recogidos* y *almacenados* y que son difíciles de *analizar* y *procesar* de forma tradicional como hojas de cálculo o bases de datos relacionales.

Este aumento en la cantidad de información recogida y almacenada viene dado gracias a la evolución de las capacidades informáticas que tenemos a nuestra disposición. Hoy en día empresas e instituciones invierten en sensores y herramientas que monitorizan todo lo que ocurre dentro y fuera de su entorno, lo que conlleva la creación de grandes bases de datos, que si se procesan y analizan de una forma adecuada pueden hallarse hábitos, tendencias y patrones que hasta ahora se encontraban ocultos.

Los inicios del Big Data pueden situarse en el año 2003, con la publicación del artículo "*The Google File System*" por parte de Google. En este artículo se presentaba un sistema de archivos *distribuido* y *escalable* para grandes aplicaciones distribuidas que precisan de un uso intensivo de los datos. Posteriormente, en el año 2006, apareció *Hadoop*, un framework software *open-source* que permite la ejecución de aplicaciones distribuidas con miles de nodos y *petabytes* de datos sobre el que se han desarrollado grandes avances. A día de hoy contamos con multitud de diferentes tecnologías, por lo que tendremos que tener en cuenta sus diferentes características y las funcionalidades que proporcionan.

En la *Figura 18*, mostramos en una imagen la lista con las diferentes tecnologías Big Data que podemos encontrar en el mercado (*año 2018*) clasificadas según su propósito, como podemos apreciar, la lista de tecnologías es prácticamente innumerable.

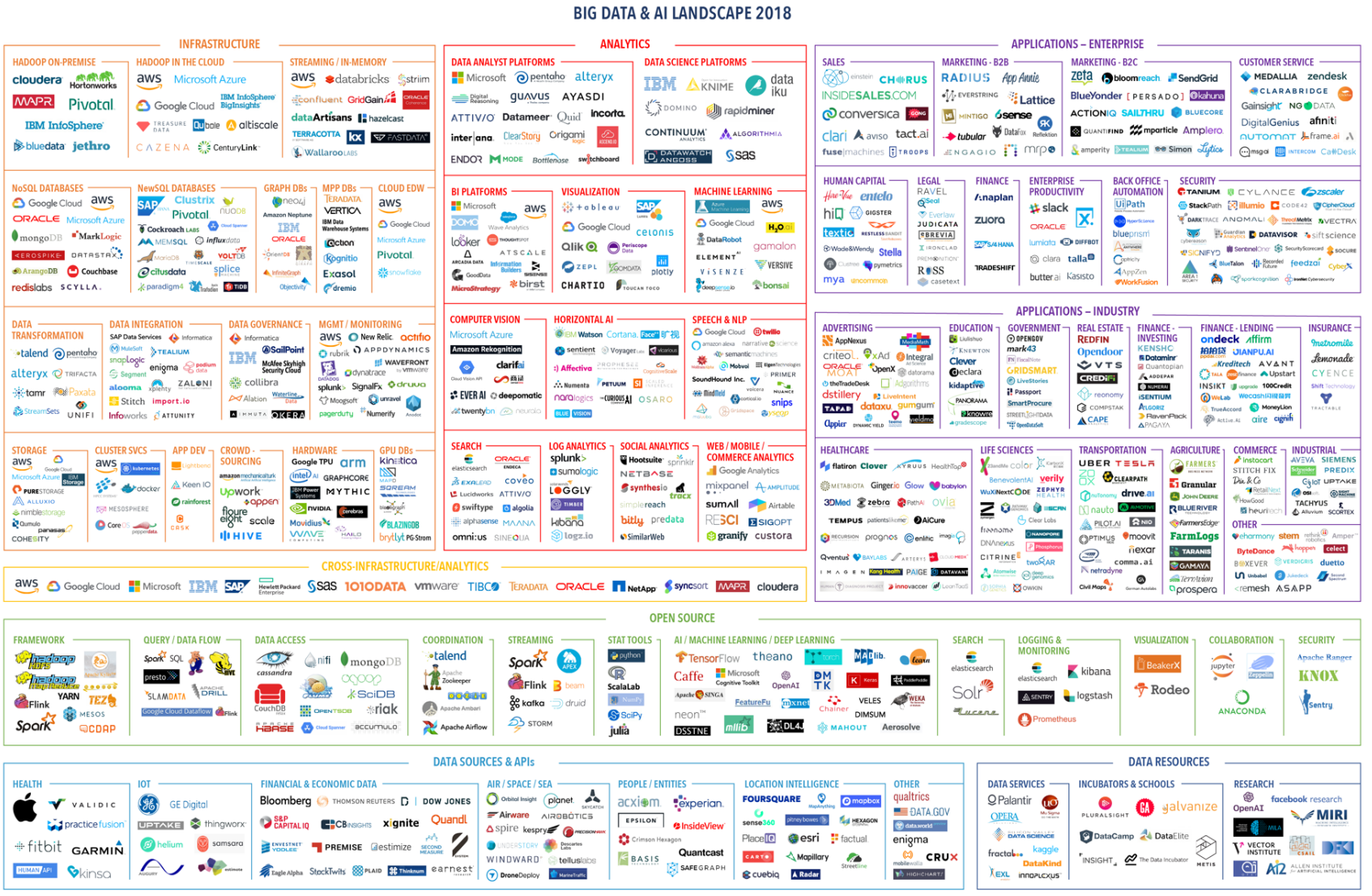


Figura 18.- Distribución tecnologías Big Data - https://mattturck.com/bigdata2018/

Como podemos ver en la *Figura 18*, las tecnologías Big Data han evolucionado de forma extremadamente rápida. Esto ha permitido que incluso pequeñas empresas puedan beneficiarse de la utilización de estas tecnologías gracias a Internet y a las plataformas en la nube.

Gracias a las plataformas en la nube desaparece la necesidad de construir una infraestructura hardware propia y de disponer de diferentes arquitectos de datos para su desarrollo y mantenimiento, ya que actualmente es posible realizar una conexión de forma remota a estos entornos a través de internet. Actualmente, gracias a la inteligencia artificial, la nube y el internet de las cosas, es posible que incluso la complejidad de las infraestructuras Big Data sea accesible para cualquier tipo de usuario y de organización.

A continuación, poniendo en contexto práctico lo expuesto anteriormente, destacamos algunas de las corrientes actuales que están permitiendo evoluciones en el mundo Big Data:

- Crecimiento de las redes IoT (*Internet of Things*)

  Hoy en día, cada vez más usuarios utilizan sus dispositivos *smartphones* para controlar diferentes aspectos de nuestra casa, y esto es gracias al IoT. Con la aparición de dispositivos inteligentes como Google Assistant, Alexa, Cortana… la inversión por parte de las compañías en el desarrollo de esta tecnología ha supuesto un aumento considerable.

  De esta forma, la inversión en el desarrollo de estas tecnologías abrirá las puertas a diferentes formas de recogida de datos, junto a los procesos de administración y análisis de los mismos. Por el momento, la respuesta de la industria a estos nuevos paradigmas está siendo apostar por nuevos dispositivos capaces de recoger, analizar y procesar la información antes de almacenarla, esta nueva forma de abordar el proceso se conoce como *Edge Computing.*

- Inteligencia artificial accesible

  Hoy en día se ha extendido notablemente el uso de la inteligencia artificial como método de mejora de los procesos en grandes y pequeñas organizaciones. Los programas dotados de inteligencia artificial son capaces de ejecutar diferentes tareas de forma más rápida y precisa que los seres humanos, disminuyendo el número de errores y mejorando el flujo de actividad de la organización. Con la aparición de estos programas se permite que el ser humano pueda centrarse en tareas más críticas para el negocio y así mejorar la calidad del servicio ofrecido.

  Ahora todo el mundo puede tener acceso a máquinas prediseñadas que incluyen software de inteligencia artificial, siendo el punto clave la correcta integración de estas nuevas herramientas en los procesos de negocio.

- Análisis predictivo

Desde su aparición, se han venido utilizando las tecnologías y las herramientas Big Data y de análisis para procesar grandes cantidades de datos y así poder determinar por qué ocurren ciertas casuísticas. Actualmente, la realización de análisis predictivo sobre Big Data permite predecir que casuísticas pueden ocurrir en el futuro.

Este tipo de estrategia es muy efectiva a la hora de analizar la información almacenada en el sistema para predecir el comportamiento del cliente a la hora de realizar la compra, el movimiento del tráfico en la ciudad, el clima…

- Migración de información analógica a la nube

La información que todavía no ha sido digitalizada es conocida como dark data. Se espera que este conjunto de información sea migrado a plataformas en la nube de forma que puedan aplicarse análisis predictivos que proporcionen beneficios para el negocio de las organizaciones.

- Computación cuántica

El procesamiento, análisis e interpretación de grandes cantidades de datos puede ser un proceso lento si se lleva a cabo con las tecnologías actuales. Gracias a la computación cuántica podrían procesarse billones de registros de información en pocos minutos, reduciendo el tiempo del proceso de forma drástica.

La computación cuántica se encuentra todavía en los primeros pasos de su desarrollo y actualmente se están llevando a cabo experimentos que ayuden en la investigación práctica y teórica en diferentes industrias. Además, se espera que próximamente grandes compañías como Google, IBM y Microsoft comiencen a realizar pruebas en computadores cuánticos para poder realizar la integración en sus procesos de negocio.

- Ciberseguridad más inteligente y estricta

Debido a diferentes ciberataques y brechas de seguridad en organizaciones durante los últimos años, las compañías han realizado un gran esfuerzo e inversión centrándose en el fortalecimiento de la confidencialidad de la organización. Otro aspecto que ha derivado sus esfuerzos en la ciberseguridad han sido las tecnologías *IoT* (*Internet of Things*), tratando de asegurar la confidencialidad de la información recogida por estos dispositivos.

Así, para prevenir y asegurar la confidencialidad de la información, gracias al Big Data pueden utilizarse análisis de datos para predecir y detectar amenazas en la seguridad. Las tecnologías Big Data pueden integrarse dentro de una estrategia de ciberseguridad, estableciendo un sistema de control de *logs* sobre amenazas pasadas, permitiendo prevenir y mitigar el impacto de futuras amenazas y brechas de seguridad.

- Soluciones *Open Source*

Existen multitud de soluciones open source relacionadas con el Big Data que han evolucionado a lo largo del tiempo. Esta evolución conlleva notables mejoras en la velocidad de procesamiento de la información, así como la posibilidad acceder y responder la información en tiempo real (*streaming*).

- *Edge computing*

Edge computing se define como el procesamiento de la información en el mismo lugar en el que es captada. De esta forma, la información se procesa lo más próxima posible a la fuente captadora, en lugar de procesarla de forma masiva en un almacén de datos centralizado (*data warehouse*).

En la actualidad existen compañías que aplican esta dinámica para obtener resultados más rápidos en el procesamiento. Por ejemplo, la gama de móviles *Pixel* de Google cuenta con chips dedicados para el aprendizaje automático (*machine learning*), procesando la información de imágenes directamente desde el propio dispositivo, lo que permite un procesamiento más rápido y de mayor calidad que el resto de dispositivos móviles.

- *Chatbots* inteligentes

Un chatbot es un programa informático con el que es posible mantener una conversación, tanto para solicitar información como para llevar a cabo ciertas tareas. Así, gracias a la evolución en inteligencia artificial, los chatbots son utilizados frecuentemente por las organizaciones en servicios de atención al cliente para procesar diferentes peticiones por parte de los usuarios.

Las tecnologías Big Data suponen un aspecto clave en la evolución de los chatbots, permitiendo una experiencia de usuario más placentera gracias al procesamiento de grandes cantidades de información para proporcionar respuestas relevantes en función de las palabras clave introducidas por el usuario. A su vez, los chatbots son capaces de almacenar y analizar información a partir de las conversaciones con los usuarios, estableciendo así un proceso de mejora continua.

Como podemos comprobar, la evolución en las tecnologías Big Data ha sido inmensa desde sus inicios con Google File System o Hadoop, pero esta evolución no se ha estancado si no que sigue un progreso constante en donde cada poco tiempo aparecen nuevos retos y avances para estas tecnologías.

2.2 Visualización

Podemos distinguir varias razones por las que debemos considerar la representación visual de la información como una de las partes fundamentales del Big Data así como de cualquier otro tipo de análisis cuyo centro de atención sea el dato.

El proceso de visualización permite al usuario que el proceso de análisis del dato sea mucho más sencillo e intuitivo, proporcionando una dimensión extra para comprender la información con la que se trabaja. De esta forma, podemos afirmar que cualquier proceso de análisis del dato comienza con la extracción de información y termina con la visualización de dicha información.

Así, de entre las diferentes razones por las que debemos considerar como fundamental la incorporación del proceso de visualización a nuestra estrategia de análisis de datos podemos destacar:

- La mente humana procesa mucho mejor la información visual.

  El proceso evolutivo del ser humano ha conectado el cerebro de una forma en la que la información visual resulta más intrigante y significativa que cualquier otra fuente de información. Así, se establece que únicamente es necesaria una fracción de segundo para retener, comprender y responder a la información que recibimos de forma visual.

  Cualquier proceso de nuestra mente necesita de visualización, pudiendo establecer una fuerte correlación entre la percepción visual y la retención del pensamiento. En este aspecto, se indica que es unas diez veces mas probable recordar información visual que la información textual.

- Mediante la visualización del dato podemos encontrar patrones en la información que se encontraban "ocultos".

  A través de la unión entre arte y estadística, la visualización del dato de una forma estética permite encontrar patrones con diferentes gamas de colores y ver cual es la relación de conexión entre dichos patrones.

  La visualización del dato contiene multitud de gráficos diferentes como gráficos de colores, grafos, animaciones, mapas, diagramas, pictografías, infografías… Mediante estos elementos los usuarios pueden encontrar patrones que hasta el momento se encontraban ocultos, correlaciones, vacíos y tendencias existentes en los datos. En definitiva, la etapa de visualización permite procesos de análisis de datos simples y rápidos.

- Una imagen vale más que mil palabras.

Otro de los aspectos con los que podemos encontrarnos a la hora de realizar procesos de análisis Big Data es con la sobrecarga de información causada por la explotación del dato. En este sentido, el objetivo de la visualización es reducir la información a su esencia y representarla en imágenes de forma simple y elegante.

Las tecnologías de visualización permiten concentrarse en los aspectos que realmente son importantes para el usuario, lo que es imprescindible en entornos en los que la cantidad de datos es muy grande y el tiempo para su exploración es muy corto.

Es importante hacer referencia a las palabras de *David McCandless,* considerando la visualización como el mapa de la información*: visualizando la información creamos un escenario que podemos explorar con la vista como si de un mapa se tratase, de esta forma, cuando nos encontremos perdidos en la información es útil disponer de un mapa de dicha información.*

- Una visualización interactiva proporciona una mejor perspectiva al usuario.

El mayor avance en las tecnologías de visualización ha sido la creación de visualizaciones y gráficos interactivos. Hoy en día, a través de herramientas avanzadas de visualización es posible modelar y remodelar los datos para seguir la estrategia que mejor funcione.

Otro aspecto que trae consigo la visualización interactiva es la capacidad de realizar informes y visualizaciones por niveles de información. Esta capacidad se conoce como Drill Down y permite al usuario acceder a diferentes niveles de la información a través de la interacción en visualizaciones.

El conjunto de esta serie de características establece el proceso de visualización como etapa imprescindible en cualquier proceso de análisis de datos.

2.3 Herramientas de estadísticas deportivas en internet

Antes de entrar a desarrollar el análisis, diseño e implementación del proyecto, es conveniente realizar un estudio previo sobre las diferentes herramientas y aplicaciones dirigidas al análisis y visualización de eventos deportivos.

Como ya se ha mencionado anteriormente, el alcance definido para este proyecto es el análisis de estadísticas para La Primera División de Fútbol de España por lo que para este estudio previo nos centraremos en aquellas herramientas centradas en el ámbito futbolístico. Así, las herramientas más significativas que se han encontrado por la red son las siguientes:

- **SofaScore (**www.sofascore.com**)**

SofaScore es una de las páginas de resultados deportivos más importantes en el ámbito web. Esta web ofrece resultados en directo para más de 500 ligas, copas y torneos de fútbol en todo el mundo. SofaScore proporciona actualizaciones en directo, resultados, estadísticas, clasificaciones…

SofaScore facilita actualizaciones rápidas y precisas sobre el tiempo de partido, resultados en cada momento, a la media parte y al final del partido, goleadores y asistentes, tarjetas, substituciones, estadísticas de partido y transmisión en directo. Además, también ofrece contenido multimedia con los videos y noticias de los momentos más destacados para las ligas más importantes del panorama (España, Italia, Alemania, Francia e Inglaterra).

Centrándonos en La Liga española, a través de SofaScore podemos visualizar la clasificación de equipos permitiéndonos seleccionar la temporada, pero no el número de jornada.

En lo que se refiere a la información para los partidos, esta aplicación divide la información en tres pestañas: en la primera muestra un pequeño gráfico con el número de ocasiones de cada equipo y muestra en orden cronológico eventos como sustituciones, goles o tarjetas; en la segunda nos proporciona información sobre diferentes métricas y estadísticas muy completas (posesión del balón, tiros, tiros a puerta, tiros fuera, paradas, saques de esquina, fueras de juego, faltas, tarjetas, tiros al palo…); y por último incluye las alineaciones de cada equipo del partido.

Por último, otra sección a destacar es la página de equipo, dónde se indican los últimos partidos jugados por el equipo, el próximo partido del equipo, número de goles marcados y recibidos y la información de los principales goleadores del equipo.

A continuación, se muestran las diferentes secciones de la aplicación descritas anteriormente:

- *Figura 19*: clasificación de equipos en la competición.
- *Figura 20*: información asociada a un partido concreto.
- *Figura 21*: información general de un equipo.

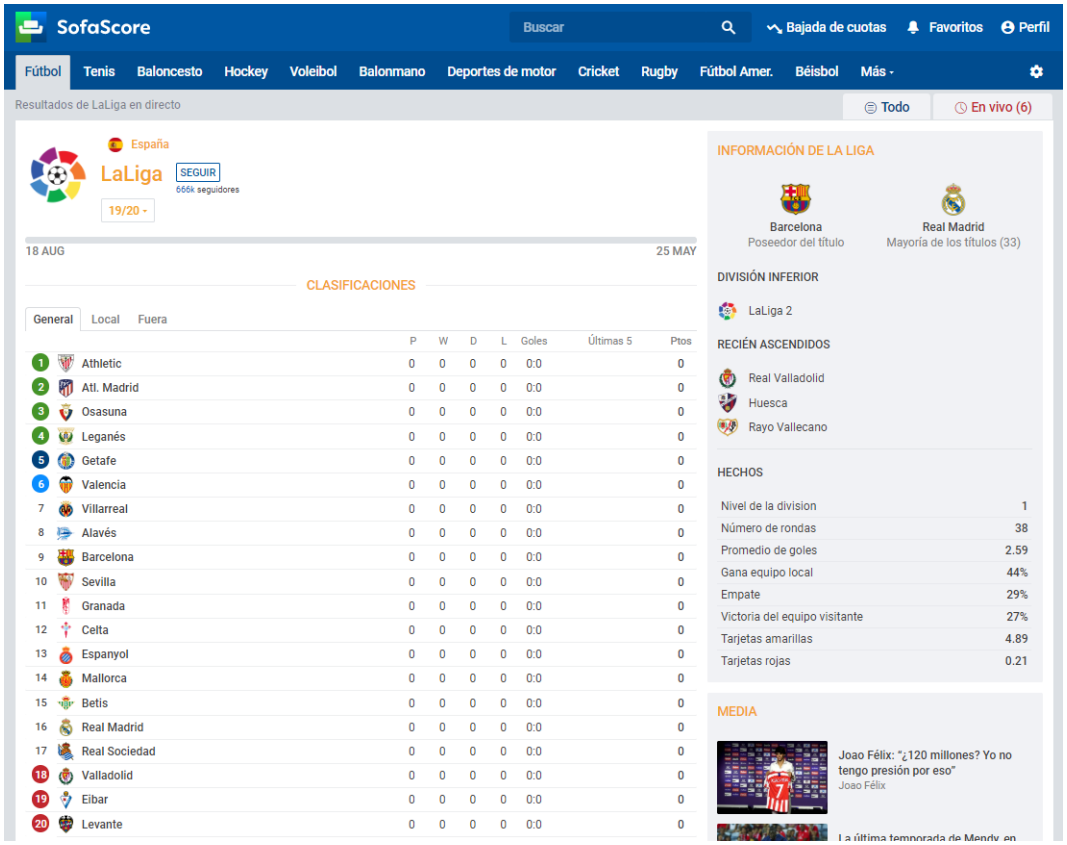


Figura 19.- SofaScore - Clasificación de equipos

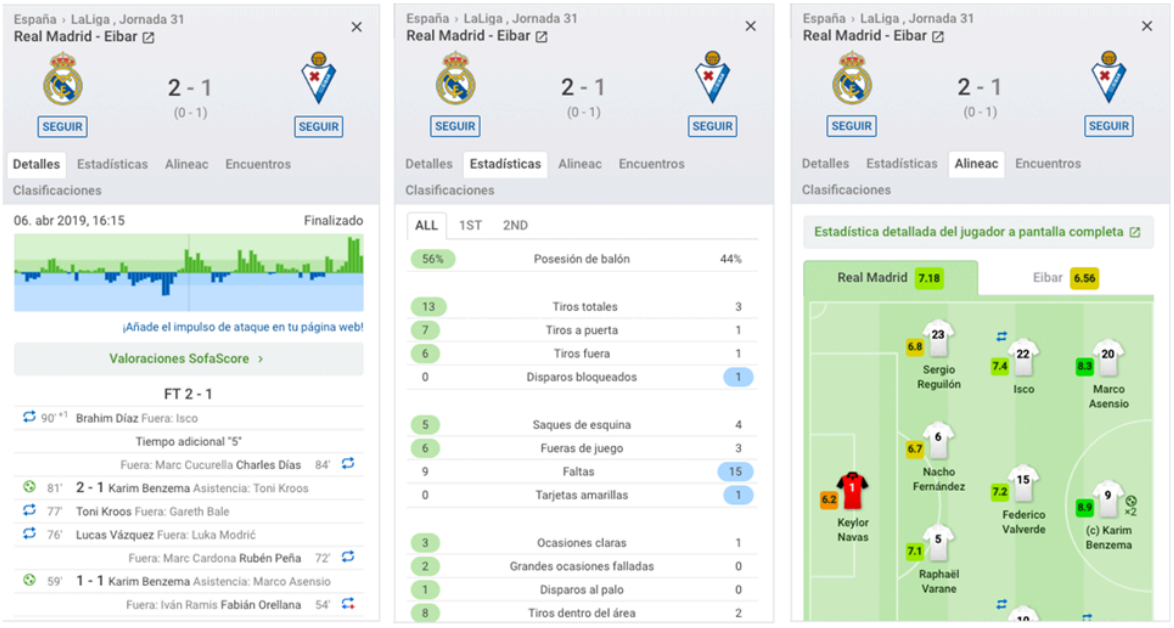


Figura 20.- SofaScore - Información del partido (Eventos principales / Estadísticas / Alineaciones)

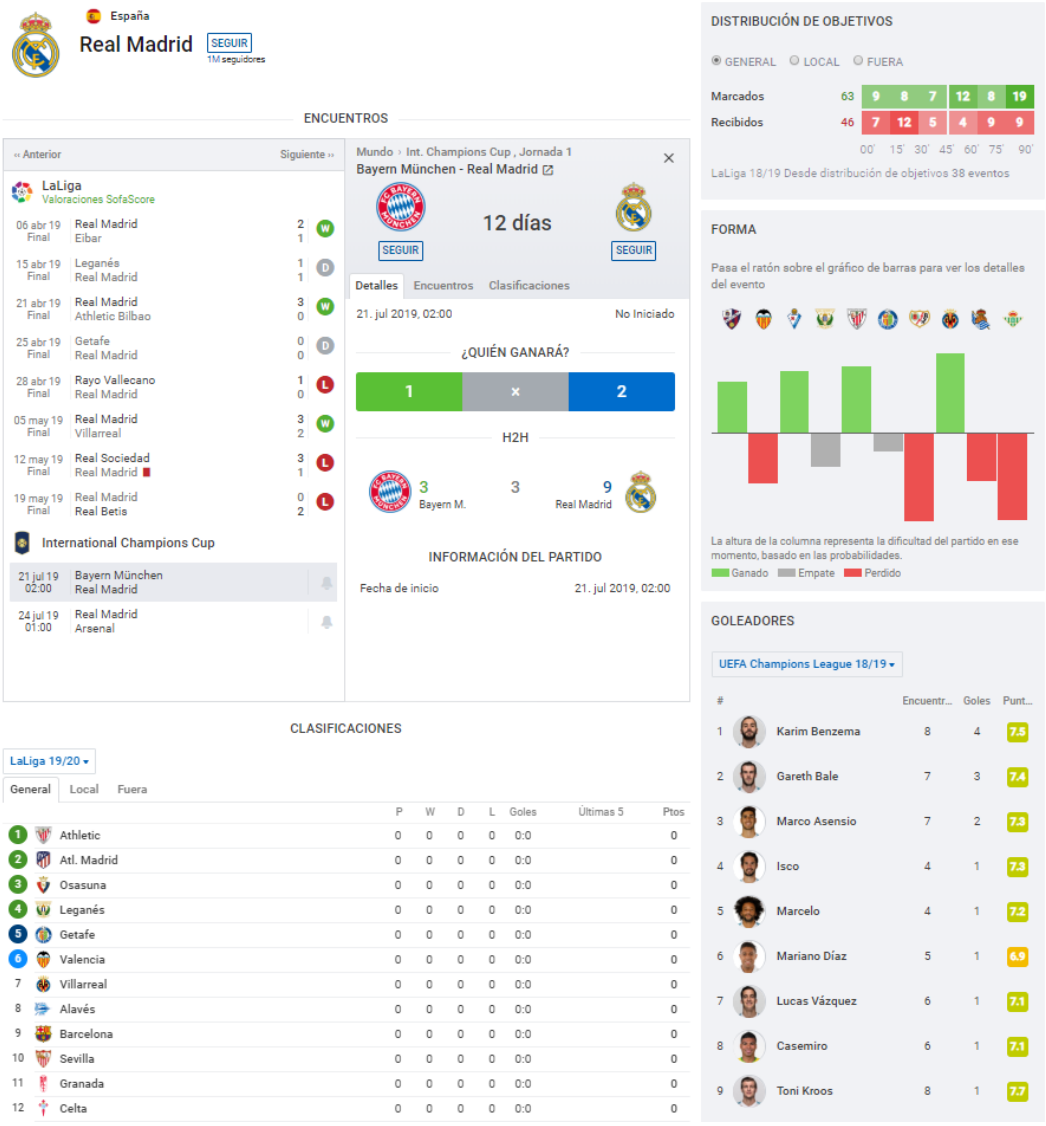


Figura 21.- SofaScore - Información de equipo

- **FlashScore (**www.flashscore.com**)**

FlashScore, al igual que la aplicación anterior, es una aplicación web de resultados en directo que incluye diferentes deportes (fútbol, tenis, baloncesto, hockey, rugby…). Esta aplicación cuenta con una interfaz más sencilla y menos vistosa que SofaScore, centrando la importancia en los resultados y en el dato y no en la visualización de estos.

De la misma forma que en la aplicación anterior, centrándonos en La Liga española podemos distinguir tres secciones diferentes: clasificaciones, partidos y equipos.

En el apartado de clasificaciones, se muestra la clasificación para la temporada actual, sin posibilidad de filtrar por temporada o por jornada, quedando así algo limitada.

Una vez accedemos a un partido concreto, abre una nueva pestaña con información estructurada de forma muy similar a SofaScore, con la diferencia de que se establece una mejor diferenciación entre los eventos del partido y aporta algo de visualización en la pestaña de estadísticas. Además de los contenidos mencionados en la aplicación anterior, FlashScore incluye también información de apuestas, información de partidos pasados. Fotos, noticias y todos los comentarios con los principales eventos del partido.

Por último, la información general para un equipo es mucho mas pobre que la proporcionada por SofaScore, mostrando únicamente los resultados de los últimos partidos, los próximos partidos e información sobre la plantilla. Otro apartado en esta sección incluye información sobre los traspasos que ha realizado dicho club (compras, ventas, cesiones…), información que no se encuentra en muchas aplicaciones del estilo.

A continuación, mostramos una serie de figuras donde podemos ver las secciones descritas anteriormente:

- *Figura 22*: clasificación de equipos en la competición.
- *Figura 23*: información asociada a un partido concreto.
- *Figura 24*: información general de un equipo e información de traspasos.

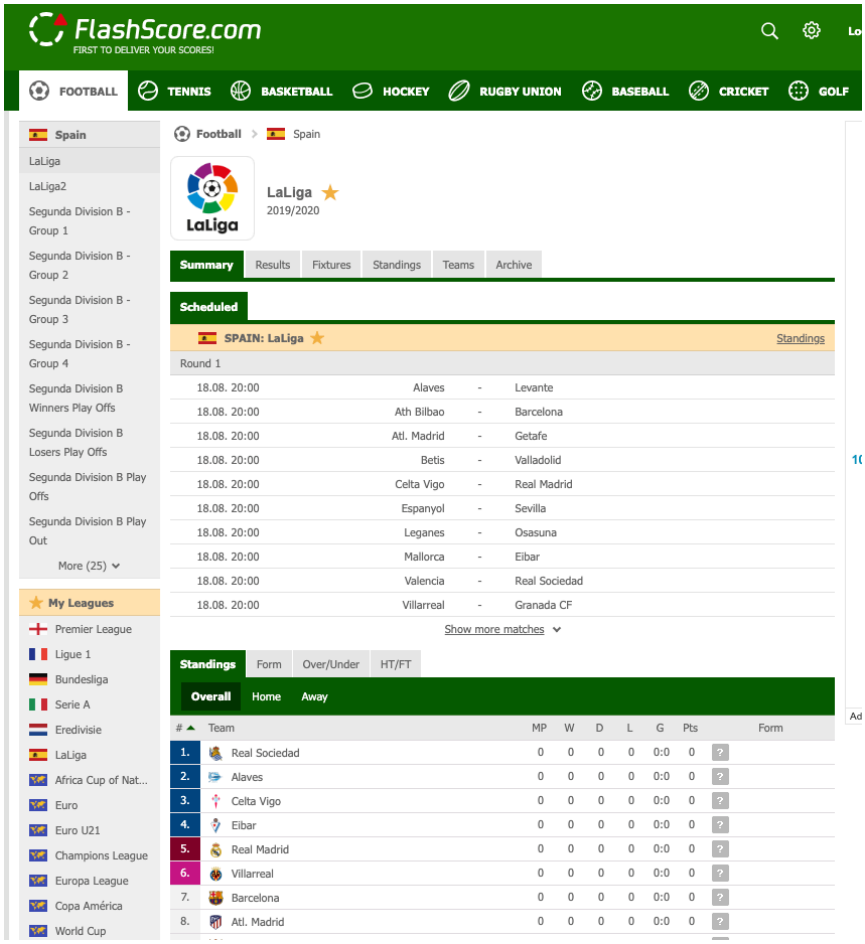


Figura 22.- FlashScore - Clasificaciones y partidos para la jornada actual

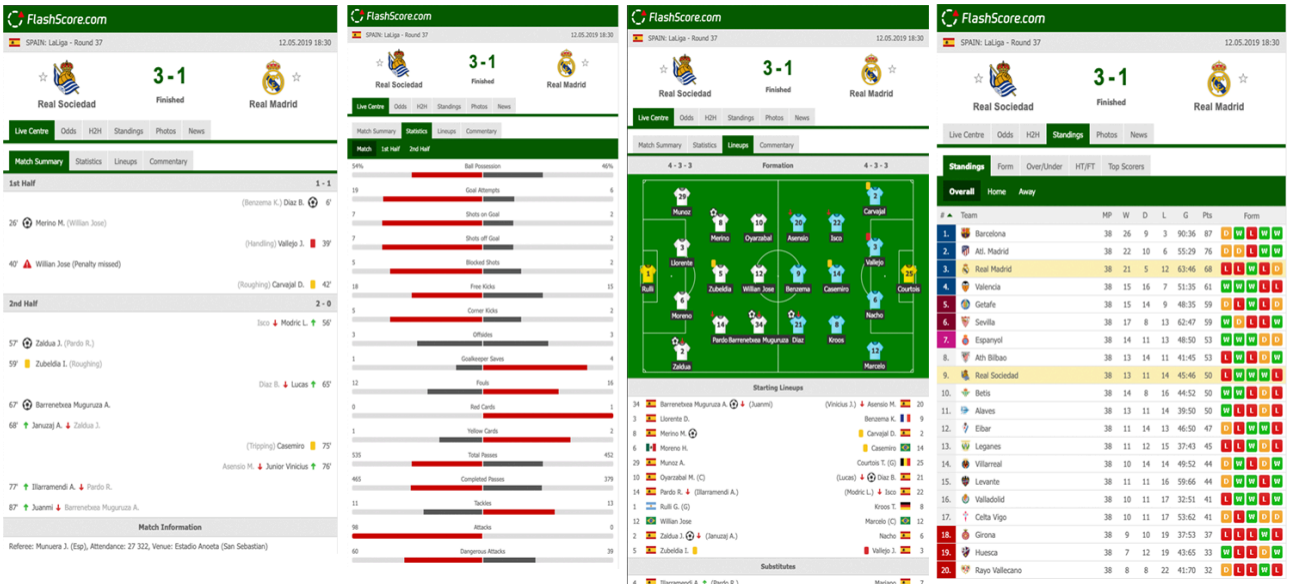


Figura 23.- FlashScore - Información del partido (Eventos principales - Estadísticas - Alineaciones - Clasificación)

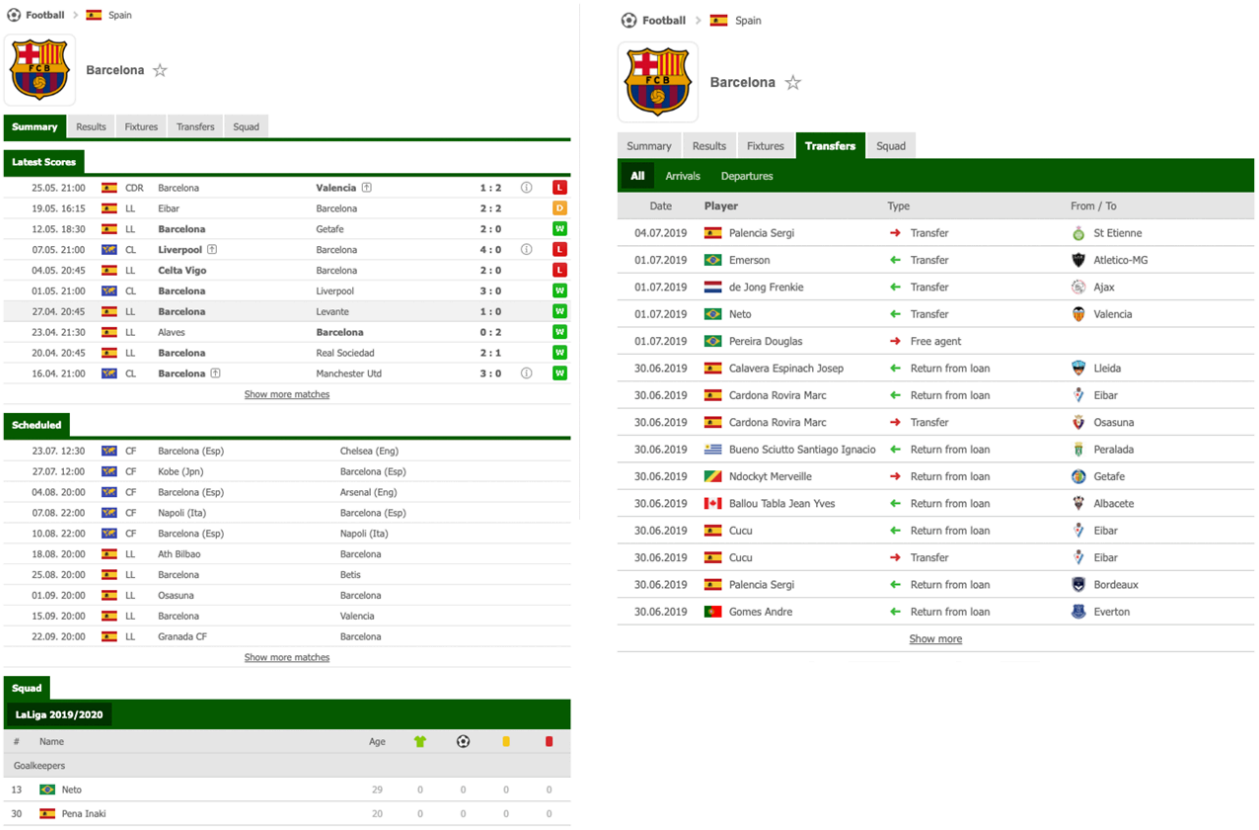


Figura 24.- FlashScore - Información de equipo (Información general / Traspasos)

- **La Liga (**www.laliga.es**)**

Otro sitio web a tener en cuenta es la página oficial de La Liga de Fútbol Profesional de España, en donde se ofrecen estadísticas individuales de cada equipo, partidos y clasificaciones actuales e históricas de la competición.

En este caso la estética de la página es totalmente diferente a las que hemos visto anteriormente ya que no se trata de una aplicación de resultados en directo, si no que se centra en la competición de futbol española. Este sitio web nos proporciona las siguientes informaciones: clasificación actual, información de partidos, estadísticas de jugadores e histórico de resultados/clasificaciones.

En lo referido a la clasificación actual, proporciona información sobre la clasificación de la temporada actual, así como estadísticas sobre partidos jugados, victorias y empates. Además, cuenta con un apartado en el que incluye información sobre los jugadores más influyentes en la temporada y la lista de resultados de la jornada actual.

La información de los partidos es bastante breve, incluyendo resultados, información arbitral y una serie de diferentes estadísticas sin incluir visualización alguna.

En lo que respecta a las estadísticas de jugadores, destacando el amplio número de filtros aplicables (liga, jornada, equipos, posición, espectro de la métrica, jugadores y búsqueda de texto libre), se muestra nombre de jugador, minutos disputados, partidos jugados, porcentaje de partidos jugados, sustituciones, tarjetas, goles… Esta información se muestra en formato tabla, careciendo de visualización.

Por último, La Liga dispone de un histórico de datos desde el que podemos consultar información tanto de clasificaciones como de partidos. La información de los partidos es bastante escasa ya que únicamente se muestra resultado, goleadores y alineaciones.

Al igual que con las herramientas anteriores, a continuación, en las siguientes figuras mostramos las capturas de pantalla en la que podemos observar las funcionalidades expuestas anteriormente:

- *Figura 25*: clasificación de equipos en la competición.
- *Figura 26*: información asociada a un partido concreto.
- *Figura 27*: tabla de estadísticas de jugadores de la competición.
- *Figura 28*: histórico de clasificaciones y resultados.

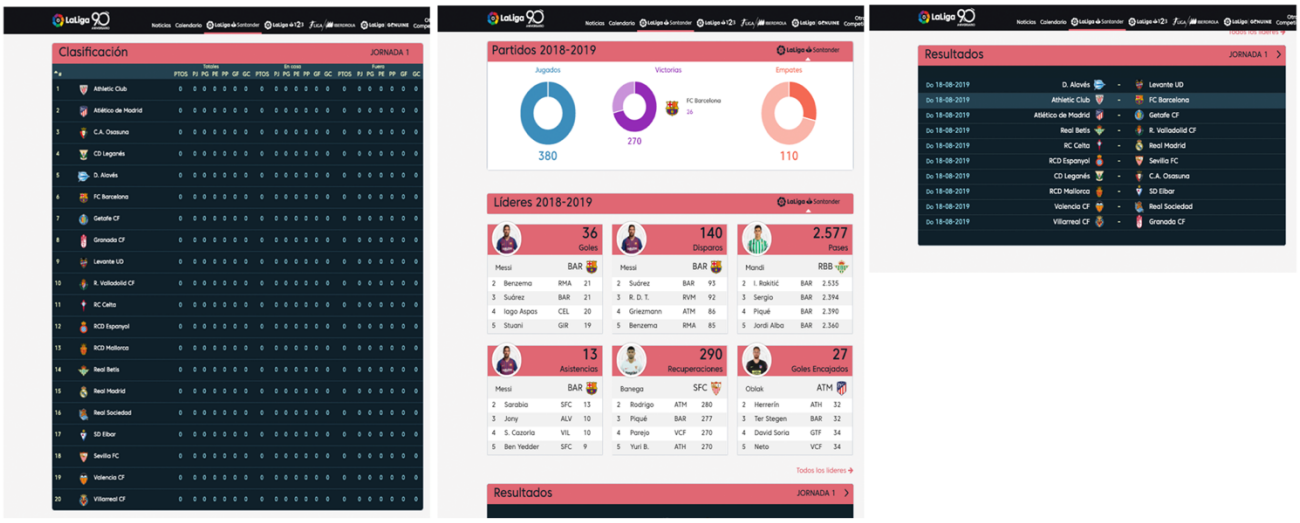


Figura 25.- La Liga - Información de clasificaciones (Clasificación - Estadísticas - Jugadores destacados - Jornada actual)

Figura 26.- La Liga – Información del partido

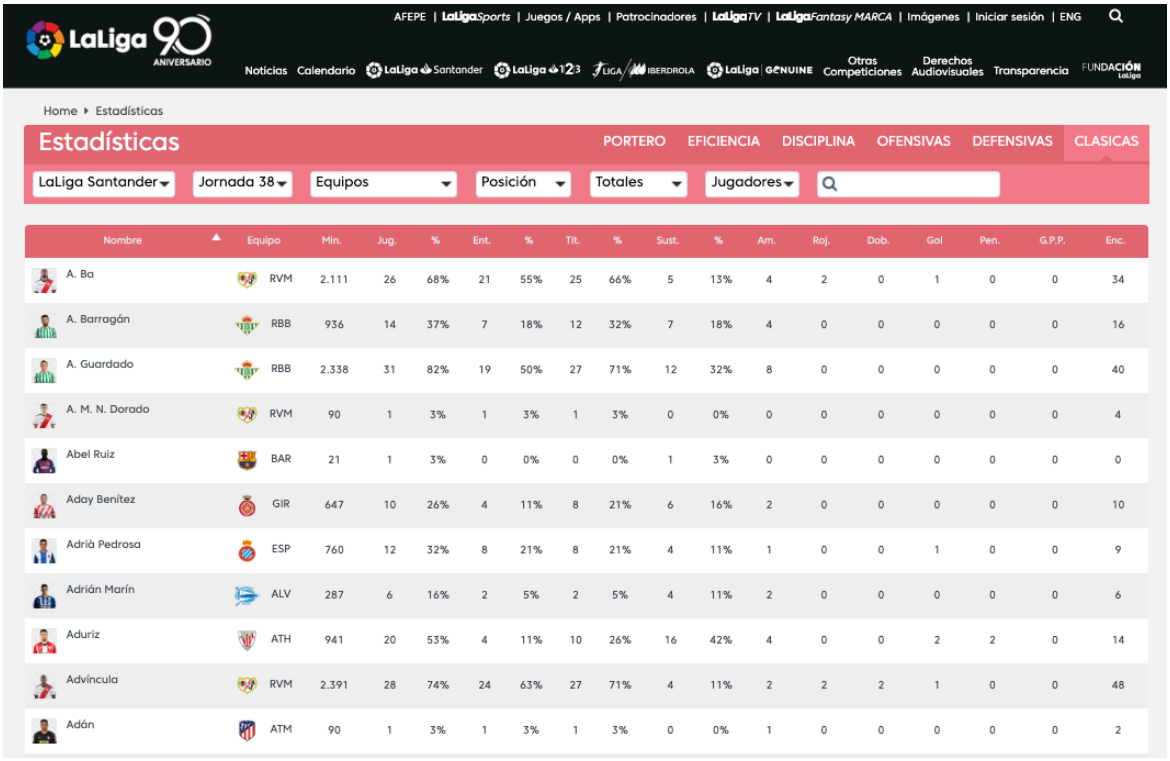


Figura 27.- La Liga - Estadísticas de jugadores

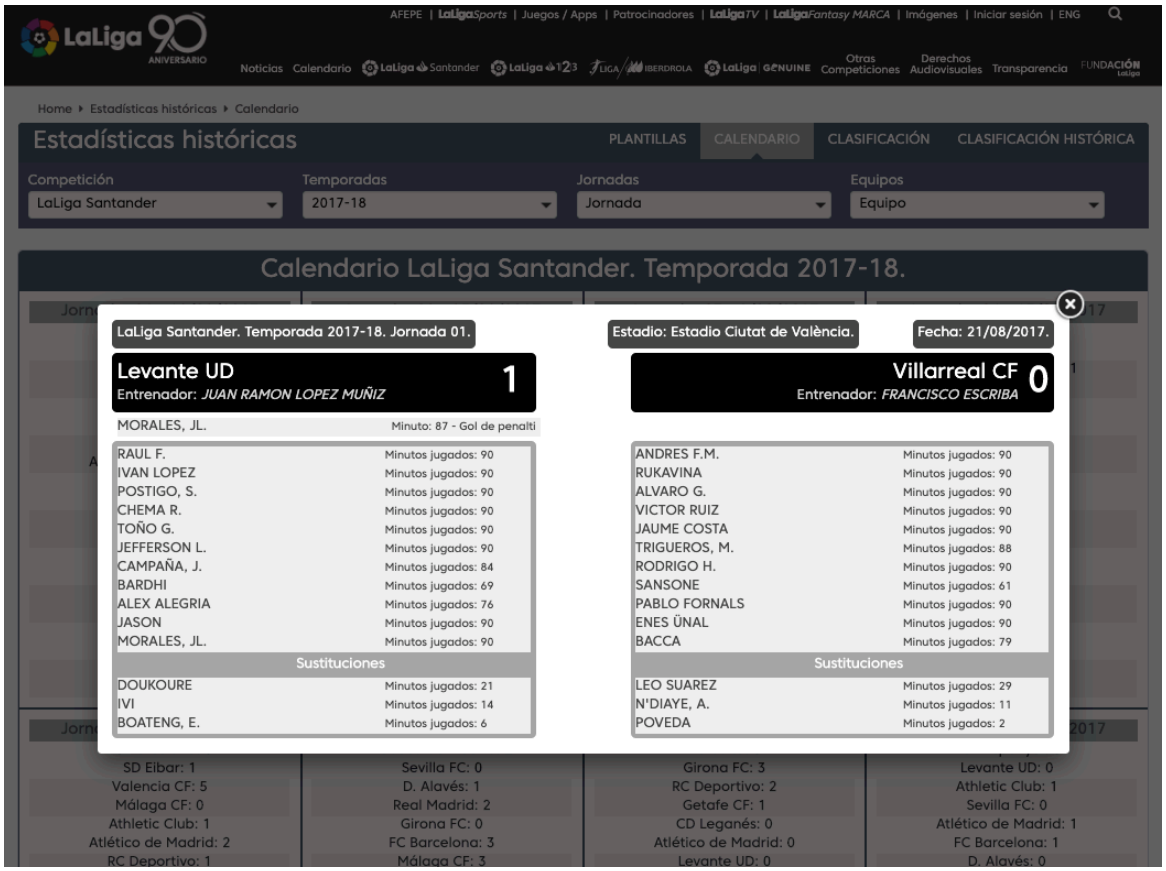


Figura 28.- La Liga - Histórico resultados/clasificaciones

- **Resultados Futbol (**www.resultados-futbol.com**)**

Este sitio web se asemeja a las dos primeras aplicaciones que han sido descritas, ya que su actividad principal es la de proporcionar resultados en directo para diferentes competiciones de ámbito futbolístico. En este caso, el sitio web cuenta con un estilo agradable en el que se muestran funcionalidades muy similares a las otras herramientas pero añadiendo más posibilidades de búsqueda e interacción con el usuario. Así, también diferenciaremos entre clasificaciones, información de partidos e información y estadísticas de equipos.

En cuanto a las clasificaciones, Resultados Futbol muestra las clasificaciones para la competición seleccionada junto con la jornada actual. A diferencia de las primeras herramientas, nos permite filtrar tanto por temporada como por número de jornada. Está página también incluye información sobre los jugadores más relevantes en la competición.

En lo que a la información de partidos se refiere, el sitio web proporciona bastante información tanto de los equipos como de las estadísticas del propio partido (goles, tarjetas, alineaciones, eventos, goles anulados, intervenciones del videoarbitraje, asistencias…). Además de la información de los eventos más relevantes del partido, también ofrece una tabla con diferentes estadísticas como tiros a puerta, paradas, saques de esquina, tiros al palo, faltas … En definitiva, proporciona una información bastante completa en comparación con otras herramientas.

Por último, en la información de un equipo concreto podemos encontrar diferentes pestañas en las que se muestra información general sobre el equipo seleccionado, histórico de partidos, histórico de temporadas, plantilla del equipo y estadísticas del equipo.

A continuación, se muestra una serie de imágenes que recogen las funcionalidades descritas anteriormente:

- *Figura 29*: clasificación de equipos en la competición y jugadores destacados.
- *Figura 30*: información asociada a un partido concreto.
- *Figura 31*: información general del equipo seleccionado.

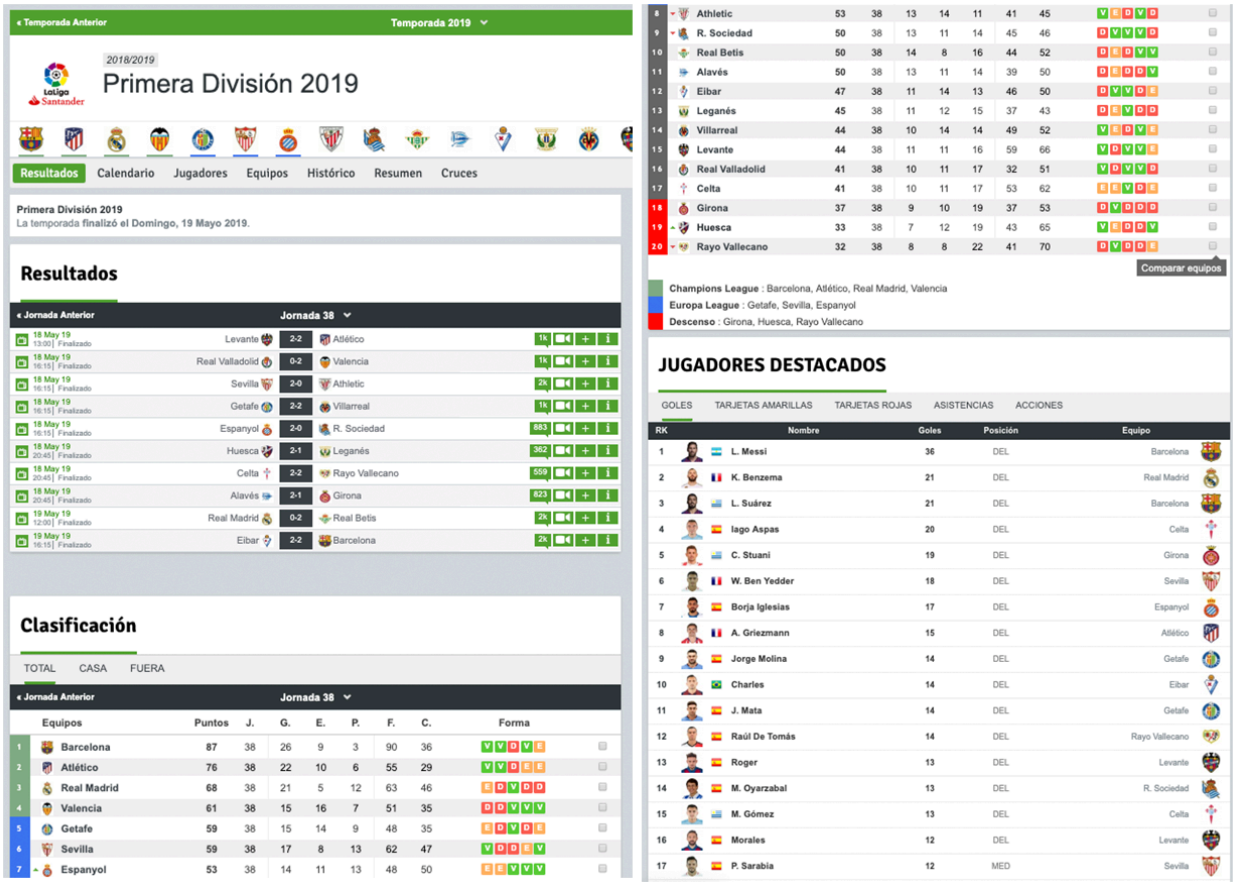


Figura 29.- Resultados Futbol – Información de clasificaciones

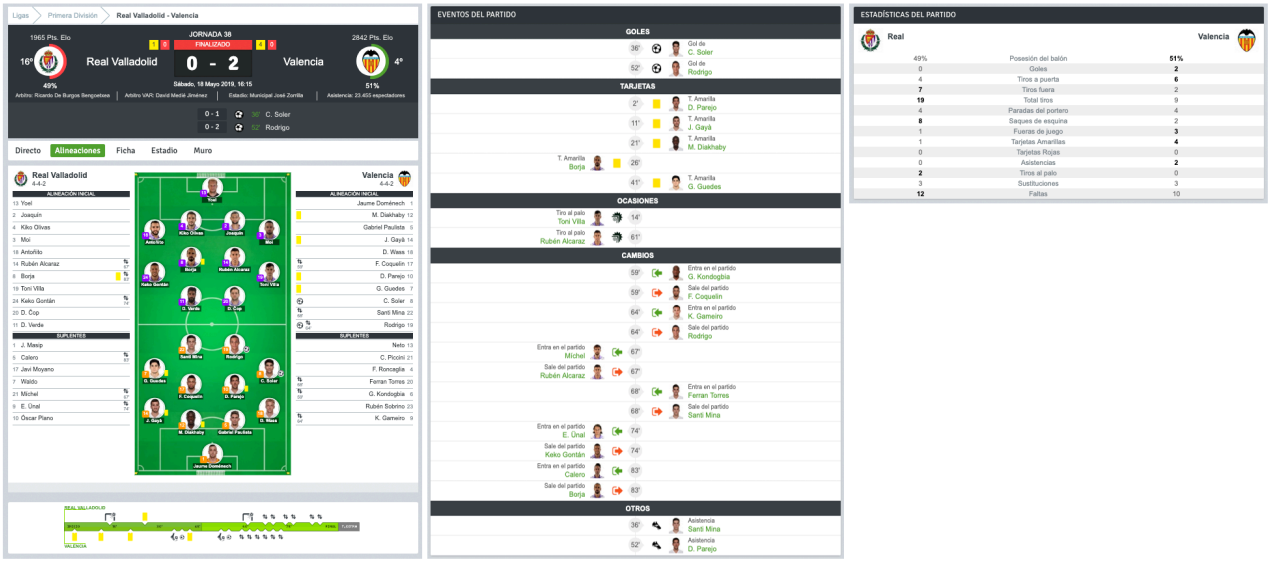


Figura 30.- Resultados Futbol – Información del partido (Datos y alineaciones – Eventos – Estadísticas)

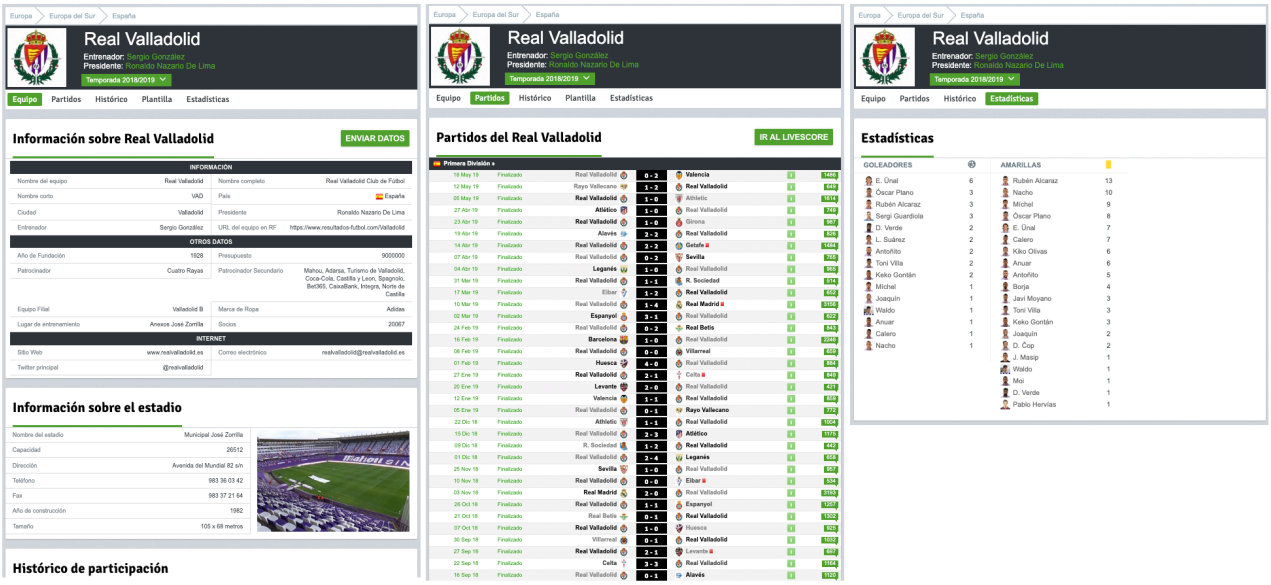


Figura 31.- Resultados Futbol – Información de equipo (Información – Histórico de partidos – Estadísticas)

2.4 Conclusiones y motivación

Una vez expuestas las principales aplicaciones de resultados y estadísticas deportivas que han sido seleccionadas, es muy importante destacar que en ninguna de ellas podemos apreciar un buen trabajo de visualización, dejando multitud de posibilidades y de información oculta para el usuario. Este es precisamente el objetivo principal de este proyecto, destacar la importancia de la visualización para comprender e investigar de una mejor forma la información que se almacena.

Así, el sitio web sobre el que se va a realizar la recopilación de datos mediante web scraping es la página de Resultados Fútbol, ya que cuenta con multitud de información correctamente estructurada y que cuenta con una navegación fluida, sencilla y personalizable para el usuario.

3 Plan de proyecto

En esta sección se indicará la metodología a seguir durante la realización del proyecto, así como la estimación temporal del mismo.

3.1 Metodología

En función de los requerimientos del sistema, se ha optado por seguir una metodología con un modelo incremental. Las principales ventajas que nos ofrece esta metodología son que proporciona unos tiempos de entrega menores y además se reduce la repetición del trabajo durante el proceso de desarrollo.

El modelo incremental de gestión de proyectos software tiene como objetivo el desarrollo progresivo de la funcionalidad del proyecto, en donde el producto a desarrollar va evolucionando con cada una de las entregas hasta que se consiguen satisfacer los requerimientos. De esta forma, se establecen una serie de entregas parciales en las que el producto debe mostrar una evolución con respecto a la entrega anterior.

Esta metodología se adapta perfectamente a las necesidades del proyecto ya que se programa una entrega por cada fase del proyecto descrita anteriormente: extracción, transformación, carga, indexación y visualización. Además, cada uno de los incrementos establecidos estará compuesto de una serie de etapas que serán replicadas hasta obtener el producto final con toda la funcionalidad. A continuación, se describe el flujo de etapas a seguir antes de realizar cada entrega:

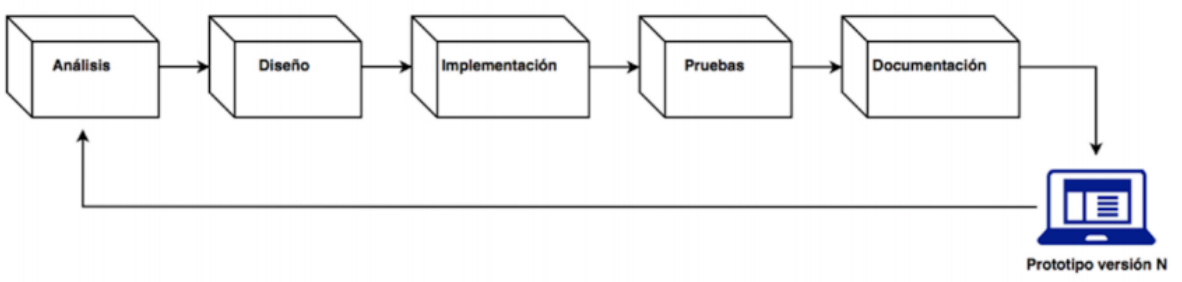


Figura 32.- Etapas de la metodología incremental

1. **Análisis**

   Es el primer paso a realizar en el modelo incremental que ha sido definido. En esta fase se extraerán las principales funcionalidades a tener en cuenta así como las nuevas funcionalidades que fuesen apareciendo durante la evolución del sistema. Durante esta fase

será de gran importancia el feedback continuo con los tutores del proyecto, evitando así posibles desviaciones en el plan.

2. Diseño

Una vez han sido extraídas las funcionalidades, la siguiente etapa del proceso es el diseño de los componentes del sistema de forma que satisfaga las necesidades establecidas durante la fase de análisis.

3. Implementación

La etapa de implementación es una de las fases de mayor importancia del proceso, ya que su función es la de transformar el diseño realizado a código de programación. Así, se obtendrá un prototipo “tangible" de la herramienta que vaya evolucionando en cada iteración.

4. Pruebas

Tras haber implementado las funcionalidades seleccionadas durante las etapas de análisis y diseño, se realizarán diferentes pruebas sobre el prototipo para evaluar el cumplimiento o no del resultado esperado para cada iteración.

5. Documentación

Como última etapa del proceso, una vez se han obtenido los resultados esperados durante la etapa de pruebas, los nuevos avances del proyecto serán recogidos en la documentación del mismo. Esto nos permite realizar también la memoria de forma incremental.

3.2 Fases de trabajo y estimación temporal

Para realizar la identificación de las diferentes fases de trabajo y de estimación temporal, se tomará como punto de partida el modelo incremental descrito anteriormente. A partir de este modelo, dispondremos de una estimación temporal de ocho meses correspondiente al periodo comprendido entre el mes de diciembre del año 2018 hasta el mes de julio del año siguiente, cuando se realizará la entrega del proyecto.

Las diferentes fases de trabajo se establecerán en función de los incrementos descritos en la metodología a seguir. De esta forma, se entregará un prototipo de la herramienta cada mes y medio de trabajo aproximadamente, dejando un pequeño margen de tiempo final para la solución de posibles problemas o inconvenientes a la hora de completar tareas pendientes y documentación.

Estas fases de trabajo equivalentes a los incrementos del proyecto serán muy similares entre ellas, pudiendo alargar o acortar una determinada fase en función del estado del proyecto.

Como es de esperar, durante los primeros meses de desarrollo del proyecto (diciembre de 2018 y enero de 2019) será necesario dedicar una mayor cantidad de tiempo a las fases de *análisis* y *diseño*, ya que durante estas primeras fases se invierte una mayor cantidad de tiempo debido a la investigación de herramientas competidoras en el mercado, tormenta de ideas de las futuras funcionalidades de la herramienta, entrevistas con los tutores del proyecto para definir correctamente el alcance del mismo… De igual forma, las fases de implementación, pruebas y documentación contarán con una duración más reducida debido a la falta de información y conocimiento del sistema.

A su vez, durante los meses de febrero, marzo, abril y mayo de 2019 ya se contará con información suficiente sobre el propósito y las funcionalidades del proyecto, lo que permitirá aplicar una breve reducción en la duración de las etapas de *análisis* y *diseño*, otorgando de un mayor abanico temporal a las etapas de *implementación* y *pruebas*. Durante este periodo, las entregas previstas serán más consistentes y con mayor funcionalidad.

Por último, durante los meses de junio y julio de 2019, se otorgará una mayor importancia a las etapas de *pruebas* y *documentación* ya que dispondremos de una visión muy completa del producto y será necesario disponer de toda la documentación a punto antes de la presentación del proyecto. Durante este periodo las fases de *análisis* y *diseño* se verán muy reducidas debido al avance en el desarrollo de la herramienta.

A continuación, podemos ver la planificación temporal descrita durante este apartado en forma de diagrama de Gantt, estableciendo de forma visual la distribución de las diferentes etapas del proyecto que han sido establecidas. Como hemos expuesto anteriormente, aproximadamente cada mes y medio se dispondrá de un prototipo actualizado de la herramienta a desarrollar.

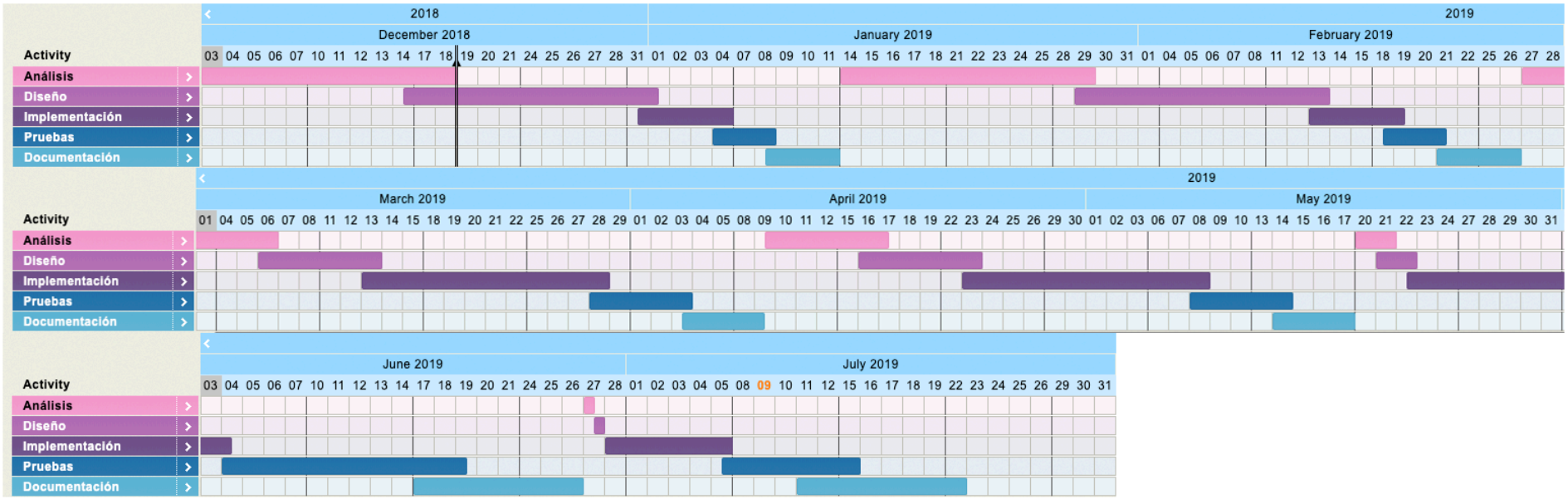


Figura 33.- Fases de trabajo y estimación temporal – Diagrama de Gantt

3.3 Presupuesto

Para la realización del presupuesto asociado al desarrollo de este Trabajo Fin de Máster, vamos a tener en cuenta la estimación temporal descrita en el apartado anterior y el conjunto de tecnologías descritas en apartados anteriores. A continuación, se expone el desglose del presupuesto final de este Trabajo Fin de Máster.

- **Presupuesto hardware:**

El presupuesto asociado al hardware del proyecto será prorrateado en función de la duración temporal del mismo y de la vida útil de los componentes. Para este desarrollo, los componentes hardware a tener en cuenta únicamente son el ordenador personal utilizado para su desarrollo y la conexión de que ha sido necesaria.

Componente	Coste total (€)	Vida útil (años)	% de uso	Coste Real (€)
MacBook Pro (Retina, 13-inch, Early 2015)	1.800 €	6	30 %	540 €
Conexión a internet	402 €	1	50 %	201 €
iPad 32 GB 2019	271 €	3	10 %	27,1 €
TOTAL				768,1 €

Tabla 1.- Presupuesto - Componentes Hardware

- **Presupuesto software:**

De la misma forma que con el presupuesto hardware, para el presupuesto software será prorrateado en función de la duración temporal del proyecto. Es importante destacar que, al haber utilizado componentes de software libre, el presupuesto software total será muy reducido.

Componente	Coste total (€)	Vida útil (años)	% de uso	Coste Real (€)
Python 3.6	0 €	-	-	0 €
Pythonista (Entorno de desarrollo para iPad)	10,99 €	1	100 %	10,99 €
PyCharm (Entorno de desarrollo)	0 €	-	-	0 €
MongoDB Community	0 €	-	-	0 €
Apache Nifi	0 €	-	-	0 €
ElasticSearch	0 €	-	-	0 €
Kibana	0 €	-	-	0 €
Kaizen	0 €	-	-	0 €
Robo3T	0 €	-	-	0 €
Google Chrome	0 €	-	-	0 €
TOTAL				10,99 €

Tabla 2.- Presupuesto - Componentes Software

- **Presupuesto de desarrollo:**

  Por último, una vez calculados los presupuestos relacionados con los elementos hardware y software del proyecto, el siguiente paso es el cálculo del presupuesto de personal para el desarrollo del proyecto. Para esta estimación se ha contado con el salario correspondiente a un único ingeniero informático en la ciudad de Valladolid, 22.000 €/año (≈11,3 €/hora). El desarrollo del Trabajo Fin de Máster ha sido compaginado con la actividad laboral, por lo que se establecerá una dedicación media de 4 horas diarias desde su inicio el día 3 de Diciembre del año 2018 hasta el día 22 de Julio del año 2019, computando un total de 171 días laborables.

  - 11,3 €/hora * 171 días * 4 horas/día ≈ 7.730 €

  Tras realizar los cálculos, obtenemos que el salario final a percibir por la persona encargada del desarrollo del proyecto es de 7.730 euros.

Una vez realizado el desglose en los cálculos del presupuesto de este Trabajo Fin de Máster, obtenemos un presupuesto total de 768 € + 10.99 € + 7.730 € = 8.509 euros.

4 Análisis y diseño

A lo largo de esta sección se describirán en profundidad las funcionalidades y características del proyecto, así como su construcción y desarrollo.

4.1 Análisis

Para que el desarrollo de un proyecto software finalice con éxito, es de vital importancia que antes de comenzar con la implementación de los diferentes módulos de los que se componga el sistema, se disponga de una completa y plena comprensión de los requisitos software del mismo.

La fase de análisis se encarga precisamente de la obtención de esta serie de requisitos que describan y establezcan la funcionalidad del sistema. De una forma más formal, podemos definir el proceso de análisis de requisitos como un proceso de descubrimiento, refinamiento, modelado y especificación del conjunto de características y funcionalidades de las que el sistema se va a encontrar dotado. Esta fase de análisis permite al desarrollador especificar la función y el rendimiento del software, indica la relación del software con otros elementos del sistema y establece las restricciones que debe cumplir el sistema.

Podemos dividir esta fase de análisis en las siguientes etapas:

1. Reconocimiento del problema

   El objetivo de esta etapa es el reconocimiento de todos los elementos básicos con los que va a contar la problemática a afrontar.

2. Evaluación y síntesis

   Una vez definidos los elementos básicos de la herramienta, es necesario definir todos los objetos de datos observables externamente, evaluar el flujo y el contenido de la información y definir y elaborar todas las funciones del software.

3. Modelado

   Durante esta etapa, el foco de atención se encuentra en los datos, en la información; siendo de vital importancia el tratamiento, el comportamiento y el contenido de la información de la aplicación.

Una vez definidas las etapas de las que se compone la fase de análisis, al igual que se ha venido haciendo a lo largo del documento, las fases del análisis serán abordadas en función del flujo de procesos de la aplicación: extracción, transformación, ingesta, indexación y visualización.

1. Extracción

Como ya se ha indicado en secciones anteriores, la extracción de la información para la herramienta se realizará aplicando técnicas de web scraping sobre el sitio web seleccionado. Web scraping conforma un proceso que nos permite extraer contenido y datos de un sitio web a partir del código HTML de la web. Actualmente estas técnicas son utilizadas en multitud de empresas digitales enfocadas a la recopilación de bases de datos; ejemplos de estas aplicaciones son:

- Rastreo, análisis y clasificación del contenido de un sitio web por parte de los motores de búsqueda.
- Sistemas de comparación de precios en la web. Estos sistemas obtienen automáticamente precios y descripciones de productos para sitios web de vendedores relacionados.
- Compañías de investigación de mercado que extraen información de foros y redes sociales.

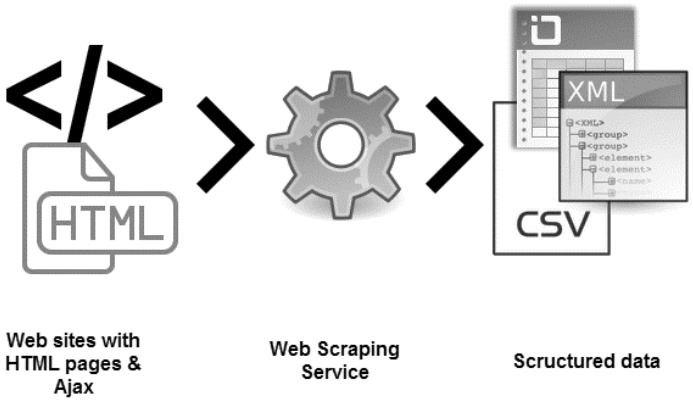


Figura 34.- Proceso web scraping

Para la implementación del web scraping de la herramienta se ha decidido utilizar la biblioteca Beautiful Soup de Python. Beautiful Soup es una biblioteca creada para la extracción de información a partir de documentos HTML y XML. Beautiful Soup incluye un parser o analizador, que nos proporciona diferentes formas para navegar, buscar y modificar el árbol de elementos HTML, lo que permite un gran ahorro de tiempo en la realización de estas tareas.

El sitio web sobre el que se va a realizar la extracción web es www.resultados-futbol.com, sitio web expuesto en la sección correspondiente al estado del arte dónde se indica por qué ha sido seleccionado entre las diferentes alternativas que han sido analizadas.

Ahora que ya han sido expuestos los elementos principales/básicos con los que va a contar la aplicación en su fase de extracción (lenguaje de programación, biblioteca de web scraping y

sitio web), pasamos a definir estos elementos en profundidad, describiendo el flujo y el contenido de la información a extraer.

A lo largo de las diferentes etapas del proceso, vamos a dividir la información en dos bloques: información de clasificaciones e información de partidos.

En cuanto a la extracción de la información de clasificaciones, se recogerán las siguientes estadísticas (en la *Figura 35* podemos ver como se estructura esta información en el sitio web):

- Temporada a la que corresponde la clasificación.
- Jornada a la que corresponde la clasificación.
- Posición en la clasificación.
- Nombre del equipo.
- Puntos en la clasificación.
- Número de partidos jugados.
- Número de partidos ganados.
- Número de partidos empatados.
- Número de partidos perdidos.
- Goles a favor.
- Goles en contra.

Las estadísticas previamente definidas serán recogidas para cada jornada de cada temporada desde la temporada 1987/1988 hasta la actualidad (2018/2019). En las *Figuras 36* y *37*, podemos ver la estructura del desplegable con el número total de temporadas y el desplegable con el número total de jornadas por temporada respectivamente.

	Equipos	Puntos	J.	G.	E.	P.	F.	C.	Forma
1	Barcelona	87	38	26	9	3	90	36	V V D V E
2	Atlético	76	38	22	10	6	55	29	V V D E E
3	Real Madrid	68	38	21	5	12	63	46	E D V D D
4	Valencia	61	38	15	16	7	51	35	D D V V V
5	Getafe	59	38	15	14	9	48	35	E D V D E
6	Sevilla	59	38	17	8	13	62	47	V D D E V
7	Espanyol	53	38	14	11	13	48	50	E E V V V
8	Athletic	53	38	13	14	11	41	45	V E D V D
9	R. Sociedad	50	38	13	11	14	45	46	D V V V D
10	Real Betis	50	38	14	8	16	44	52	D E D V V
11	Alavés	50	38	13	11	14	39	50	D E D D V
12	Eibar	47	38	11	14	13	46	50	D V V D E
13	Leganés	45	38	11	12	15	37	43	D E V D D
14	Villarreal	44	38	10	14	14	49	52	V E D V E
15	Levante	44	38	11	11	16	59	66	V D V V E
16	Real Valladolid	41	38	10	11	17	32	51	V D V V D
17	Celta	41	38	10	11	17	53	62	E E V D E
18	Girona	37	38	9	10	19	37	53	D V D D D
19	Huesca	33	38	7	12	19	43	65	V E D D V
20	Rayo Vallecano	32	38	8	8	22	41	70	D V D D E

Figura 35.- Estructura clasificación – www.resultados-futbol.com

Figura 36.- Listado de temporadas – www.resultados-futbol.com

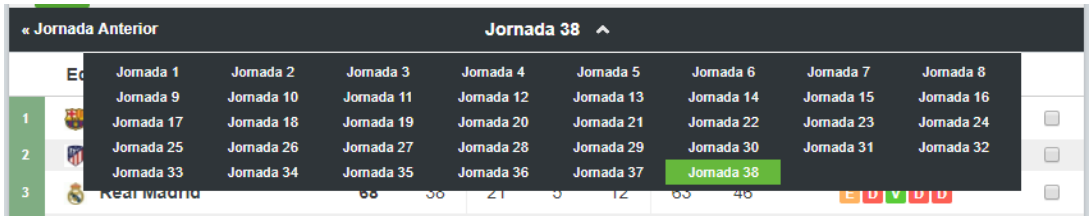


Figura 37.- Listado de jornadas – www.resultados-futbol.com

En lo que respecta a la información de partidos, podemos establecer tres grupos diferentes: información general del partido y estadísticas del partido (estadísticas del equipo local y estadísticas del equipo visitante). El conjunto de estadísticas que serán capturadas son las siguientes:

- Información general del partido:
    - Fecha de celebración partido.
    - Temporada correspondiente a la celebración del partido.
    - Jornada en la que se celebra dicho partido.
    - Árbitro principal del partido.
    - Estadio en el que se juega el partido.
    - Número de asistentes al partido.
    - Resultado del partido.

- Estadísticas del partido (equipo local):
    - Nombre del equipo local.
    - Entrenador del equipo local.
    - Formación del equipo local.
    - Alineación titular del equipo local.
    - Alineación suplente del equipo local.
    - Posesión del balón del equipo local.
    - Número de goles marcados por el equipo local.
    - Información de los goles marcados por el equipo local (jugador y minuto).
    - Número de asistencias completadas por el equipo local.
    - Información de las asistencias completadas por el equipo local (jugador y minuto).
    - Información de los penaltis marcados por el equipo local (jugador y minuto).
    - Información de los goles en propia puerta marcados por el equipo local (jugador y minuto).
    - Información de los goles en propia puerta del equipo visitante, que finalmente son goles para el equipo local (jugador y minuto).
    - Información de los goles anulados al equipo local (jugador y minuto).
    - Información de los goles marcados de falta por el equipo local (jugador y minuto).
    - Información de los penaltis fallados por el equipo local (jugador y minuto).
    - Información de los penaltis parados por el equipo local (jugador y minuto).
    - Número total de disparos efectuados por el equipo local.
    - Número total de disparos a puerta efectuados por el equipo local.
    - Número total de disparos fuera de puerta efectuados por el equipo local.
    - Número total de paradas del portero del equipo local a lo largo del partido.
    - Número total de saques de esquina efectuados por el equipo local.
    - Número total de fueras de juego señalados al equipo local.
    - Número total de disparos al palo efectuados por el equipo local.
    - Información de los disparos al palo efectuados por el equipo local (jugador y minuto)
    - Número total de faltas cometidas por el equipo local.
    - Número total de tarjetas amarillas recibidas por el equipo local.
    - Información de las tarjetas amarillas recibidas por el equipo local (jugador y minuto).
    - Número total de tarjetas rojas recibidas por el equipo local.
    - Información de las tarjetas rojas recibidas por el equipo local (jugador y minuto).
    - Información de las tarjetas rojas a raíz de segunda amarilla recibidas por el equipo local (jugador y minuto).
    - Número total de lesiones producidas en el equipo local.

- Información de las lesiones producidas en el equipo local (jugador y minuto).
- Número total de cambios realizados por el equipo local.
- Información de los jugadores que salen del campo en el equipo local (jugador y minuto).
- Información de los jugadores que entran al campo en el equipo local (jugador y minuto).

- Estadísticas del partido (equipo visitante):
  - Nombre del equipo visitante.
  - Entrenador del equipo visitante.
  - Formación del equipo visitante.
  - Alineación titular del equipo visitante.
  - Alineación suplente del equipo visitante.
  - Posesión del balón del equipo visitante.
  - Número de goles marcados por el equipo visitante.
  - Información de los goles marcados por el equipo visitante (jugador y minuto).
  - Número de asistencias completadas por el equipo visitante.
  - Información de las asistencias completadas por el equipo visitante (jugador y minuto).
  - Información de los penaltis marcados por el equipo visitante (jugador y minuto).
  - Información de los goles en propia puerta marcados por el equipo visitante (jugador y minuto).
  - Información de los goles en propia puerta del equipo local, que finalmente son goles para el equipo visitante (jugador y minuto).
  - Información de los goles anulados al equipo visitante (jugador y minuto).
  - Información de los goles marcados de falta por el equipo visitante (jugador y minuto).
  - Información de los penaltis fallados por el equipo visitante (jugador y minuto).
  - Información de los penaltis parados por el equipo visitante (jugador y minuto).
  - Número total de disparos efectuados por el equipo visitante.
  - Número total de disparos a puerta efectuados por el equipo visitante.
  - Número total de disparos fuera de puerta efectuados por el equipo visitante.
  - Número total de paradas del portero del equipo visitante a lo largo del partido.
  - Número total de saques de esquina efectuados por el equipo visitante.
  - Número total de fueras de juego señalados al equipo visitante.
  - Número total de disparos al palo efectuados por el equipo visitante.
  - Información de los disparos al palo efectuados por el equipo visitante (jugador y minuto)
  - Número total de faltas cometidas por el equipo visitante.
  - Número total de tarjetas amarillas recibidas por el equipo visitante.
  - Información de las tarjetas amarillas recibidas por el equipo visitante (jugador y minuto).
  - Número total de tarjetas rojas recibidas por el equipo visitante.
  - Información de las tarjetas rojas recibidas por el equipo visitante (jugador y minuto).
  - Información de las tarjetas rojas a raíz de segunda amarilla recibidas por el equipo visitante (jugador y minuto).
  - Número total de lesiones producidas en el equipo visitante.

- Información de las lesiones producidas en el equipo visitante (jugador y minuto).
- Número total de cambios realizados por el equipo visitante.
- Información de los jugadores que salen del campo en el equipo visitante (jugador y minuto).
- Información de los jugadores que entran al campo en el equipo visitante (jugador y minuto).

A continuación, se exponen las figuras con la estructura de la información descrita anteriormente:

1. *Figura 38*: muestra la estructura de la información general de un partido.
2. *Figura 39*: muestra la estructura de la información de las alineaciones de cada equipo para un partido concreto.
3. *Figura 40*: muestra los principales eventos ocurridos durante el partido en orden cronológico.
4. *Figura 41*: muestra la información de las estadísticas recogidas durante el partido seleccionado.

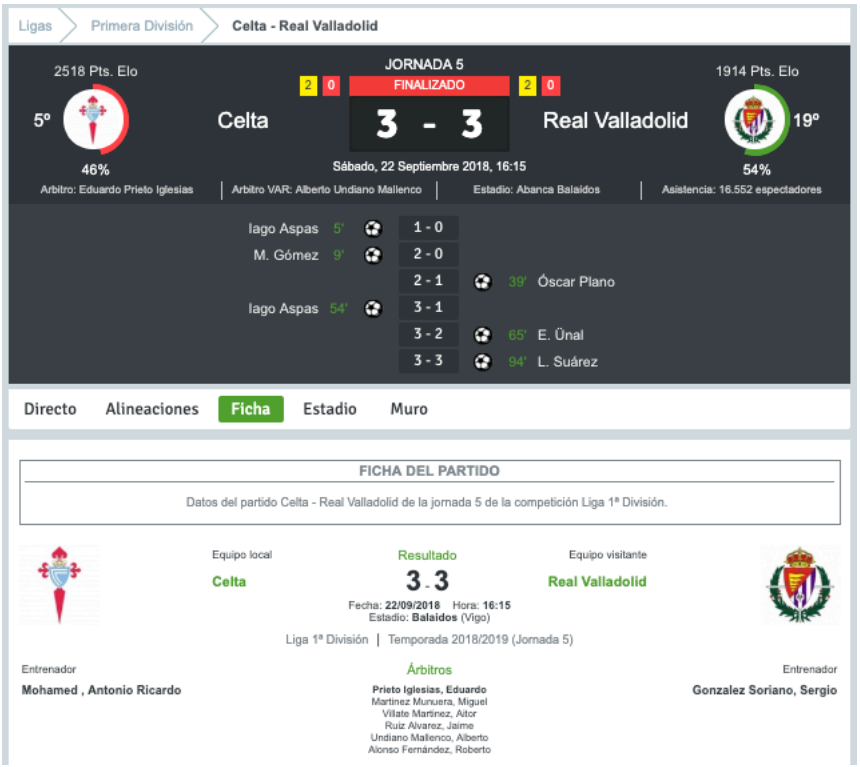


Figura 38.- Información general del partido – www.resultados-futbol.com

Figura 39.- Información de alineaciones de los equipos – www.resultados-futbol.com

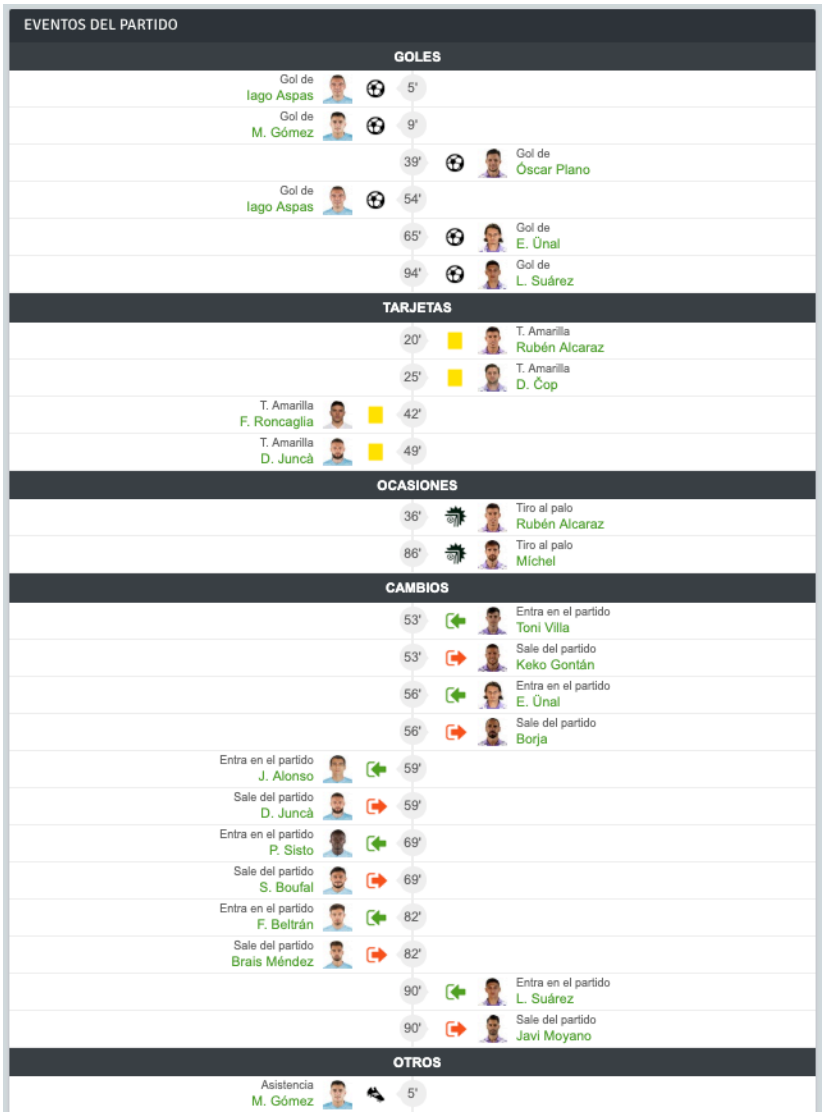


Figura 40.- Información de eventos del partido (goles, tarjetas, ocasiones, cambios...) – www.resultados-futbol.com

ESTADÍSTICAS DEL PARTIDO

Celta		Real
46%	Posesión del balón	**54%**
3	Goles	3
5	Tiros a puerta	4
3	Tiros fuera	**4**
12	Total tiros	**14**
1	Paradas del portero	**2**
7	Saques de esquina	**8**
2	Fueras de juego	1
2	Tarjetas Amarillas	2
0	Tarjetas Rojas	0
2	Asistencias	**3**
0	Tiros al palo	**2**
3	Sustituciones	3
9	Faltas	**11**

Figura 41.- Información de estadísticas del partido (tiros a puerta, saques de esquina, fueras de juego, faltas...) – www.resultados-futbol.com

De forma similar al proceso de extracción de la información de las clasificaciones, la información del partido se obtendrá para cada jornada de cada temporada para todos los partidos desde la temporada 1999/2000 hasta la temporada actual 2018/2019. Esto se debe a que desde la temporada 1999/2000 hacia atrás la información de los partidos no se encuentra tan detallada y no aportaría información de interés para el análisis.

El proceso de extracción de datos no requiere de la etapa de modelado de los mismos, ya que esta tarea se realizará una vez han sido transformados los datos en el siguiente proceso.

2. Transformación

De forma posterior a la extracción de la información para cada clasificación y cada partido recogido, desde el mismo módulo Python se transformará la información para finalmente disponer de un diccionario con los datos transformados que serán almacenados en base de datos en forma de documentos.

Una vez obtenida la información deseada es muy importante identificar los patrones en los que se encuentra estructurada dentro del código HTML de la página para seleccionar únicamente la información relevante para nuestro estudio y, una vez seleccionada, prestar especial atención a los posibles problemas que pueden ocurrir y acotarlos correctamente para eliminar un posible impacto a la hora de almacenar, indexar y visualizar la información. Ejemplos de estos problemas puede ser la ausencia de información en el lugar en el que debería encontrarse, el cambio de estructura de la información en un momento determinado o la inclusión de etiquetas HTML en la información extraída. Todos estos controles serán llevados a cabo durante este proceso de transformación de datos.

La etapa clave durante este proceso es el correcto modelado de la información descrita en el proceso de extración. Así, al igual que en el apartado anterior, distinguiremos entre el proceso de transformación de la información de las clasificaciones y de la información de los partidos. En cuanto a la complejidad del código, debido a la estructuración y la cantidad de información, el proceso de transformación de la información de los partidos será considerablemente más complejo que el proceso de transformación de la información de las clasificaciones.

A continuación, se indica el diccionario de datos correspondiente a la información de clasificaciones y el diccionario de datos correspondiente a la información de partidos:

- Información de clasificaciones

Diccionario	Atributo	Tipo de dato	Descripción
Clasificaciones	season	String	Temporada a la que pertenece la información capturada.
	matchday	Int32	Jornada a la que pertenece la información capturada.
	team_name	String	Nombre del equipo al que pertenece la información capturada.
	position	Int32	Posición en la clasificación asociada al equipo.
	points	Int32	Número total de puntos del equipo.
	played_matches	Int32	Número total de partidos jugados por el equipo.
	win	Int32	Número total de partidos ganados asociados al equipo.
	draw	Int32	Número total de partidos empatados asociados al equipo.
	lost	Int32	Número total de partidos perdidos asociados al equipo.
	goals_favor	Int32	Número total de goles marcados por el equipo.
	goals_against	Int32	Número total de goles encajados por el equipo.

Tabla 3.- Diccionario de datos para clasificaciones

- Información de partidos

* Debido a la extensión de la tabla, ésta ha sido incluida en el *Anexo 1* de este documento como *Tabla 4.- Diccionario de datos para partidos*.

Tras situar el contexto de este proceso de transformación de la información, pasamos a definir el modelado de los datos en forma de diccionarios Python para la fase de extracción de clasificaciones y de partidos.

Tanto para la fase de extracción de clasificaciones como para la fase de extracción de partidos, la estructura de diccionarios que fue definida inicialmente tuvo que ser modificada debido a diferentes problemas durante la etapa de visualización, debido a estas modificaciones, distinguiremos entre la primera fase del modelado, con la definición inicial, y la segunda fase del modelado, con la estructura finalmente desarrollada:

- Modelado de la información (primera aproximación):
    - Clasificaciones

La primera aproximación llevada a cabo para el modelado de la información proveniente de las clasificaciones consiste en la creación de un diccionario de datos por

jornada. De esta manera, el diccionario estaba compuesto por la información de la temporada asociada a los datos, la información de la jornada asociada a los datos y un array por cada posición de la clasificación compuesto por la posición, nombre del equipo, número de puntos, número de partidos disputados, partidos ganados, partidos empatados, partidos perdidos, goles a favor y goles en contra. A continuación, se muestra un ejemplo con el modelo de diccionario desarrollado para la primera jornada de la temporada 2017/2018:

```
{
    "season" : "2017/2018",
    "matchday" : 1,
    "pos_1" : {
        "position" : 1,
        "team_name" : "Real-Madrid",
        "points" : 3,
        "played_matches" : 1,
        "win" : 1,
        "draw" : 0,
        "lost" : 0,
        "favor" : 3,
        "against" : 0
    },
    "pos_2" : {
        "position" : 2,
        "team_name" : "Barcelona",
        "points" : 3,
        "played_matches" : 1,
        "win" : 1,
        "draw" : 0,
        "lost" : 0,
        "favor" : 2,
        "against" : 0
    },
    "pos_3" : {…},
     "pos_4" : {…},
…
}
```

Mediante esta primera aproximación, aparecieron diversos problemas a la hora de visualizar la información ya que no permitía realizar un análisis de la forma deseada, limitando considerablemente la cantidad y el tipo de gráficos y filtrados de la herramienta.

- Partidos

En lo referido al modelado de la información de los partidos, se desarrolló un diccionario compuesto por 82 campos que estructuraban la información extraída de la web definida durante el proceso de extración. Este diccionario se encontraba compuesto por los siguientes campos:

```
{
```

```
"match_id" : "2019_Girona_Valladolid",
"match_date" : "2018/08/17 20:15:00",
"match_season" : "2018/2019",
"match_matchday" : 1,
"match_referee" : "Cuadra Fernandez, Guillermo",
"match_stadium" : "Municipal Montilivi",
"match_attendance" : 10368,
"match_score" : "0-0",
"match_team1" : "Girona",
"match_team1_coach" : "Sacristan Mena, Eusebio",
"match_team1_formation" : "4-5-1",
"match_team1_align_main" : […],
"match_team1_align_sup" : […],
"match_team1_possession" : 67.0,
"match_team1_goals" : 0,
"match_team1_assists" : 0,
"match_team1_assist_info" : […],
"match_team1_goals_info" : […],
"match_team1_goal_penalty_info" : […],
"match_team1_goal_own_info" : […],
"match_team1_goal_own2_info" : […],
"match_team1_goal_cancelled_info" : […],
"match_team1_goal_freekick_info" : […],
"match_team1_penalty_missed_info" : […],
"match_team1_penalty_saved_info" : […],
"match_team1_penalty_saved2_info" : […],
"match_team1_totalshoots" : 13,
"match_team1_shootsontarget" : 1,
"match_team1_shootsofftarget" : 10,
"match_team1_goalkeepersaves" : 1,
"match_team1_cornerkicks" : 3,
"match_team1_offsides" : 1,
"match_team1_posts" : 0,
"match_team1_post_info" : […],
"match_team1_fouls" : 21,
"match_team1_yellowcards" : 1,
"match_team1_yellowcard_info" : […],
"match_team1_redcards" : 0,
"match_team1_card_red_info" : […],
"match_team1_card_red2yellow_info" : […],
"match_team1_injuries" : 0,
"match_team1_injury_info" : […],
"match_team1_subs" : 3,
"match_team1_subs_in_info" : […],
"match_team1_subs_out_info" : […],
"match_team2" : "Valladolid",
"match_team2_coach" : "",

"match_team2_formation" : "4-5-1",
"match_team2_align_main" : […],
"match_team2_align_sup" : […],
"match_team2_possession" : 33.0,
"match_team2_goals" : 0,
"match_team2_assists" : 0,
"match_team2_assist_info" : […],
"match_team2_goals_info" : […],
```

```
    "match_team2_goal_penalty_info" : […],
    "match_team2_goal_own_info" : […],
    "match_team2_goal_own2_info" : […],
    "match_team2_goal_cancelled_info" : […],
    "match_team2_goal_freekick_info" : […],
    "match_team2_penalty_missed_info" : […],
    "match_team2_penalty_saved_info" : […],
    "match_team2_penalty_saved2_info" : […],
    "match_team2_totalshoots" : 2,
    "match_team2_shootsontarget" : 1,
    "match_team2_shootsofftarget" : 1,
    "match_team2_goalkeepersaves" : 2,
    "match_team2_cornerkicks" : 2,
    "match_team2_offsides" : 1,
    "match_team2_posts" : 0,
    "match_team2_post_info" : […],
    "match_team2_fouls" : 20,
    "match_team2_yellowcards" : 1,
    "match_team2_yellowcard_info" : […],
    "match_team2_redcards" : 0,
    "match_team2_card_red_info" : […],
    "match_team2_card_red2yellow_info" : […],
    "match_team2_injuries" : 0,
    "match_team2_injury_info" : […],
    "match_team2_subs" : 3,
    "match_team2_subs_in_info" : […],
    "match_team2_subs_out_info" : […]
}
```

Al igual que en el caso de la estructuración de la información de las clasificaciones, con el modelo de datos anterior aparecían inconsistencias a la hora de visualizar la información. Estos errores se producían en los campos de información (match_team1_assist_info, match_team1_goals_info, match_team1_goal_penalty_info, match_team1_goal_cancelled_info, match_team1_penalty_saved_info, …). El problema con estos campos era su estructura, ya que en todos los casos se almacenaba un array con el tipo de información, el jugador de la información y el minuto de la información. Esto producía fallos en la visualización ya que la herramienta englobaba en un mismo campo los tres componentes del array.

A continuación, mostramos varios casos prácticos con la estructura de algunos de estos campos correspondientes a la información de tarjetas amarillas, sustituciones (jugadores entrantes) y goles en propia puerta:

```
    "match_team1_yellowcard_info" :
[[
            "yellow card",
            "Á. Granell",
            "87"
       ]]


    "match_team1_subs_in_info" :
```

```
[
        [
            "in",
            "A. Lozano",
            "62"
        ],
        [
            "in",
            "Pere Pons",
            "74"
        ],
        [
            "in",
            "P. Roberts",
            "81"
        ]
       ]

    "match_team1_goal_own_info" :
[
        [
            "goal own",
            "David López",
            "51"
        ]
       ]
```

La problemática con estos campos se resolverá con el nuevo modelo establecido en la aproximación final del modelado.

- Modelado de la información (aproximación final):
  - Clasificaciones

La solución al problema expuesto en el modelado de la información de clasificaciones fue sustituir el proceso de crear un único diccionario por jornada, por la creación de un diccionario por posición en la clasificación. De esta forma aumentará el número de registros creados pero estos contaran con una estructura menos compleja.

A continuación, se muestra la estructura final de los diccionarios generados con la información de clasificaciones. Para comprender mejor esta nueva estructura se muestran los diccionarios correspondientes a las posiciones primera, segunda y tercera correspondientes a la primera jornada de la temporada 2017/2018:

- Primera posición

```
{
    "season" : "2017/2018",
    "matchday" : 1,
    "team_name" : "Real-Madrid",
```

```
    "position" : 1,
    "points" : 3,
    "played_matches" : 1,
    "win" : 1,
    "draw" : 0,
    "lost" : 0,
    "goals_favor" : 3,
    "goals_against" : 0
}
```


- Segunda posición


```
{
    "season" : "2017/2018",
    "matchday" : 1,
    "team_name" : "Barcelona",
    "position" : 2,
    "points" : 3,
    "played_matches" : 1,
    "win" : 1,
    "draw" : 0,
    "lost" : 0,
    "goals_favor" : 2,
    "goals_against" : 0
}
```


- Tercera posición


```
{
    "season" : "2017/2018",
    "matchday" : 1,
    "team_name" : "Real-Sociedad",
    "position" : 3,
    "points" : 3,
    "played_matches" : 1,
    "win" : 1,
    "draw" : 0,
    "lost" : 0,
    "goals_favor" : 3,
    "goals_against" : 2
}
```


Con esta nueva estructura de la información se solucionaron los problemas en la etapa de visualización, permitiendo la creación de métricas, gráficos y filtros más complejos.

- Partidos

Para solucionar la problemática con los campos de información del diccionario expuestos anteriormente, se optó por eliminar dichos campos y sustituyendo cada campo de información por dos nuevas entradas en el diccionario, una para la información de jugadores y otra para la información del minuto de la acción. Para comprender mejor

este cambio en la estructura, a continuación, se muestra la sustitución del campo match_team1_subs_in_info utilizado como ejemplo en la sección anterior:

- Primera aproximación:

```
"match_team1_subs_in_info" :
[
        [
            "in",
            "A. Lozano",
            "62"
        ],
        [
            "in",
            "Pere Pons",
            "74"
        ],
        [
            "in",
            "P. Roberts",
            "81"
        ]
       ]
```

- Aproximación final:

```
   "match_team1_subs_in_player"  :  [  "A.  Lozano",  "Pere  Pons",  "P.
Roberts"],

   "match_team1_subs_in_minute" : [62, 74, 81],
```

Una vez actualizada la estructura del diccionario, la herramienta de visualización permitía la realización del análisis correctamente.

3. Ingesta

Una vez los datos han sido extraídos y transformados, el siguiente proceso del proyecto es el almacenamiento de estos datos. Como ya se ha indicado en secciones anteriores, la tecnología seleccionada como almacén de datos ha sido MongoDB, en dónde destaca el rendimiento en el procesamiento e inserción de los datos y la estructuración de los mismos en un formato equivalente a los diccionarios generados en Python una vez concluida la etapa de transformación.

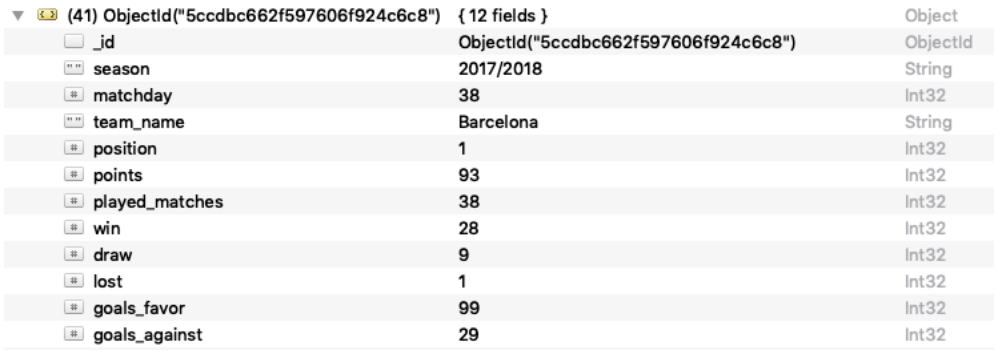
Así, para la organización de la información se establece la creación de una única base de datos compuesta por dos colecciones de documentos independientes, una para almacenar los datos correspondientes a la información de las clasificaciones y otra para el almacenamiento de los datos correspondientes a la información de los partidos.

Es importante destacar que el proceso de conexión con la base de datos y de inserción de documentos es realizado desde el propio programa Python una vez se han transformado los datos.

A continuación, mostramos un ejemplo de la estructura de los registros almacenados en cada colección de la base de datos MongoDB. Para realizar la visualización de los documentos, como se ha expuesto en secciones anteriores, se utiliza la herramienta Robo 3T, que dota de interfaz gráfica a las conexiones con MongoDB.

- Colección con la información correspondiente a las clasificaciones.

A modo de ejemplo se muestra el documento correspondiente a la primera posición para la última jornada de La Liga de Fútbol Profesional de la temporada 2017/2018:



(41) ObjectId("5ccdbc662f597606f924c6c8")	{ 12 fields }	Object
_id	ObjectId("5ccdbc662f597606f924c6c8")	ObjectId
season	2017/2018	String
matchday	38	Int32
team_name	Barcelona	String
position	1	Int32
points	93	Int32
played_matches	38	Int32
win	28	Int32
draw	9	Int32
lost	1	Int32
goals_favor	99	Int32
goals_against	29	Int32

Figura 42.- Estructura documento MongoDB para el almacenamiento de las clasificaciones

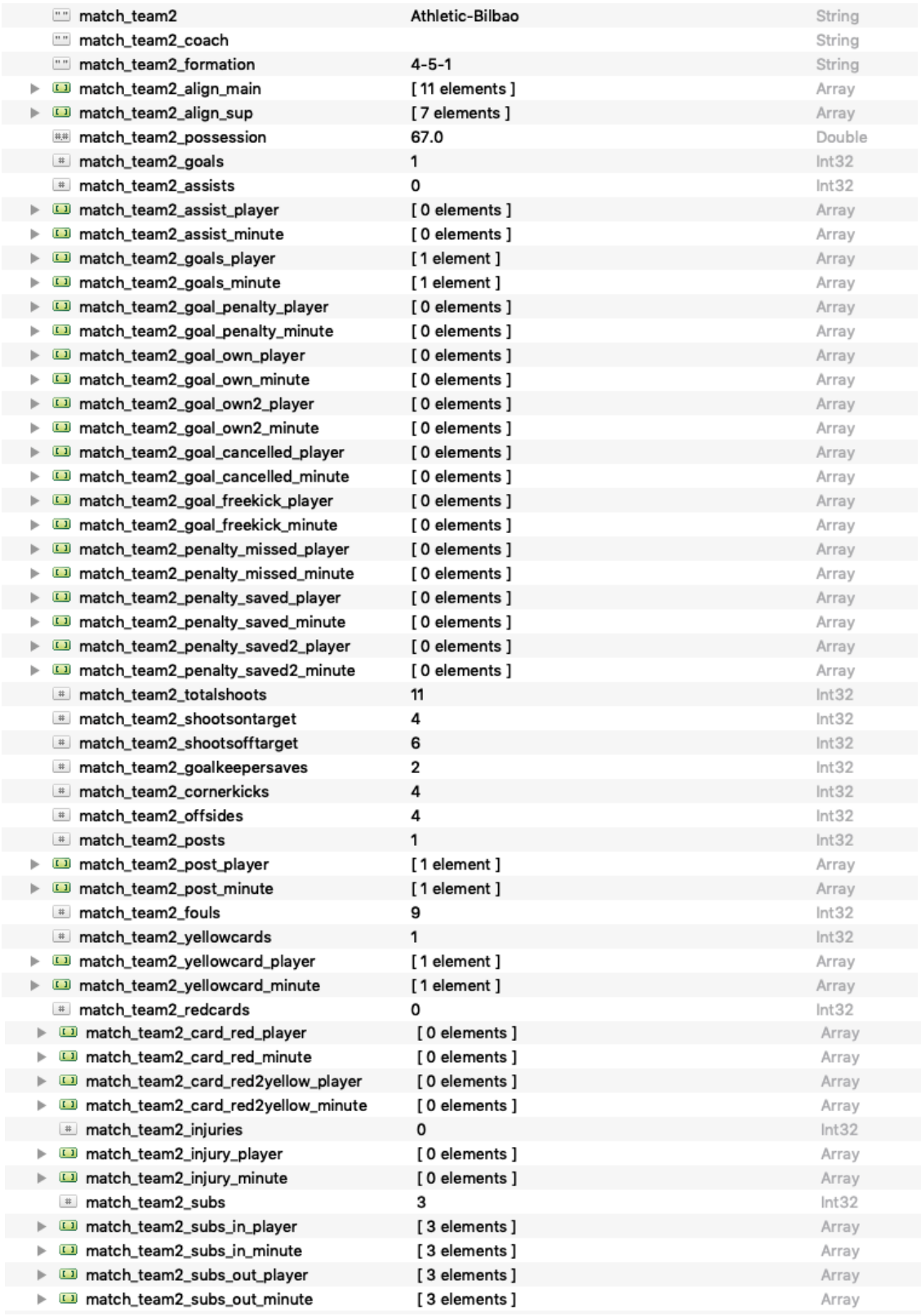
- Colección con la información correspondiente a los partidos.

A modo de ejemplo se muestra el documento correspondiente al partido disputado entre los equipos Rayo Vallecano y Athletic de Bilbao para la tercera jornada de la temporada 2018/2019 de La Liga de Fútbol Profesional:

(30) ObjectId("5d24ba8b2f597609906cdf10")	{ 117 fields }	Object
_id	ObjectId("5d24ba8b2f597609906cdf10")	ObjectId
match_id	2019_Rayo-Vallecano_Athletic-Bilbao	String
match_date	2018/10/24 19:00:00	String
match_season	2018/2019	String
match_matchday	3	Int32
match_referee	Cuadra Fernandez, Guillermo	String
match_stadium	Vallecas	String
match_attendance	0	Int32
match_score	1-1	String
match_team1	Rayo-Vallecano	String
match_team1_coach	Sanchez Muñoz, Miguel Angel	String
match_team1_formation	4-5-1	String
match_team1_align_main	[11 elements]	Array
[0]	Alberto García	String
[1]	J. Amat	String
[2]	Álex Moreno	String
[3]	L. Advíncula	String
[4]	A. Gálvez	String
[5]	G. Imbula	String
[6]	S. Comesaña	String
[7]	G. Kakuta	String
[8]	J. Pozo	String
[9]	A. Embarba	String
[10]	Raúl De Tomás	String
match_team1_align_sup	[7 elements]	Array
[0]	S. Dimitrievski	String
[1]	S. Akieme	String
[2]	Abdoulaye Ba	String
[3]	E. Velázquez	String
[4]	Álvaro García	String
[5]	Bebé	String
[6]	Alex Alegría	String
match_team1_possession	33.0	Double
match_team1_goals	1	Int32
match_team1_assists	0	Int32
match_team1_assist_player	[0 elements]	Array
match_team1_assist_minute	[0 elements]	Array
match_team1_goals_player	[1 element]	Array
[0]	J. Pozo	String
match_team1_goals_minute	[1 element]	Array
[0]	23	Int32

Figura 43.- Estructura documento MongoDB para el almacenamiento de los partidos – parte 1

Figura 44.- Estructura documento MongoDB para el almacenamiento de los partidos – parte 2



match_team2	Athletic-Bilbao	String
match_team2_coach		String
match_team2_formation	4-5-1	String
match_team2_align_main	[11 elements]	Array
match_team2_align_sup	[7 elements]	Array
match_team2_possession	67.0	Double
match_team2_goals	1	Int32
match_team2_assists	0	Int32
match_team2_assist_player	[0 elements]	Array
match_team2_assist_minute	[0 elements]	Array
match_team2_goals_player	[1 element]	Array
match_team2_goals_minute	[1 element]	Array
match_team2_goal_penalty_player	[0 elements]	Array
match_team2_goal_penalty_minute	[0 elements]	Array
match_team2_goal_own_player	[0 elements]	Array
match_team2_goal_own_minute	[0 elements]	Array
match_team2_goal_own2_player	[0 elements]	Array
match_team2_goal_own2_minute	[0 elements]	Array
match_team2_goal_cancelled_player	[0 elements]	Array
match_team2_goal_cancelled_minute	[0 elements]	Array
match_team2_goal_freekick_player	[0 elements]	Array
match_team2_goal_freekick_minute	[0 elements]	Array
match_team2_penalty_missed_player	[0 elements]	Array
match_team2_penalty_missed_minute	[0 elements]	Array
match_team2_penalty_saved_player	[0 elements]	Array
match_team2_penalty_saved_minute	[0 elements]	Array
match_team2_penalty_saved2_player	[0 elements]	Array
match_team2_penalty_saved2_minute	[0 elements]	Array
match_team2_totalshoots	11	Int32
match_team2_shootsontarget	4	Int32
match_team2_shootsofftarget	6	Int32
match_team2_goalkeepersaves	2	Int32
match_team2_cornerkicks	4	Int32
match_team2_offsides	4	Int32
match_team2_posts	1	Int32
match_team2_post_player	[1 element]	Array
match_team2_post_minute	[1 element]	Array
match_team2_fouls	9	Int32
match_team2_yellowcards	1	Int32
match_team2_yellowcard_player	[1 element]	Array
match_team2_yellowcard_minute	[1 element]	Array
match_team2_redcards	0	Int32
match_team2_card_red_player	[0 elements]	Array
match_team2_card_red_minute	[0 elements]	Array
match_team2_card_red2yellow_player	[0 elements]	Array
match_team2_card_red2yellow_minute	[0 elements]	Array
match_team2_injuries	0	Int32
match_team2_injury_player	[0 elements]	Array
match_team2_injury_minute	[0 elements]	Array
match_team2_subs	3	Int32
match_team2_subs_in_player	[3 elements]	Array
match_team2_subs_in_minute	[3 elements]	Array
match_team2_subs_out_player	[3 elements]	Array
match_team2_subs_out_minute	[3 elements]	Array

Figura 45.- Estructura documento MongoDB para el almacenamiento de los partidos – parte 3

4. Indexación

El siguiente proceso del proyecto se compone de la indexación de los datos almacenados de forma que disponga de una visualización con un rendimiento excelente, prácticamente sin demora en el filtrado y en la búsqueda de información.

Para realizar esta tarea, como ya se ha indicado en secciones anteriores, se ha seleccionado la tecnología ElasticSearch, un motor de búsqueda distribuido basado en Apache Lucene que cuenta con una interfaz web RESTful y que almacena los documentos a indexar en formato JSON.

En su página web, ElasticSearch define los índices como "bases de datos" en una base de datos relacional, en dónde se establece un mapeo que define múltiples tipos. De una forma más técnica podríamos definir los índices como espacios de nombres lógicos encargados de asignar la relación con una parte en concreto del clúster y que puede tener ninguna o múltiples replicas.

Podemos establecer un símil entre la estructura de la información en ElasticSearch con una base de datos relacional, en dónde un índice sería la base de datos y el tipo de índice sería la tabla de la base de datos. La principal diferencia con una base de datos relacional es que, en ElasticSearch un tipo de índice se compone de diferentes documentos con una serie de propiedades, mientras que en una base de datos relacional una tabla se compone por filas y columnas. Finalmente, es importante destacar que un clúster de ElasticSearch puede contener múltiples índices (bases de datos), que pueden contener múltiples tipos de índice (tablas), que a su vez contienen multitud de documentos (filas), y cada documento tiene unas propiedades definidas (columnas).

De esta forma, al igual que en el almacén de datos la información se divide en dos colecciones diferentes, en ElasticSearch se implementarán dos índices de búsqueda correspondientes a la información de las clasificaciones y a la información de los partidos.

A continuación, se indica la estructura que seguirá cada uno de estos índices:

- Índice correspondiente a la información de clasificaciones:

```
{
  "properties": {
    "goals_favor": {
      "type": "integer"
    },
    "goals_against": {
      "type": "integer"
    },
    "matchday": {
      "type": "integer"
    },
    "lost": {
      "type": "integer"
    },
    "season": {
      "type": "text",
```

```
      "fields": {
        "keyword": {
          "ignore_above": 256.0,
          "type": "keyword"
        }
      }
    },
    "draw": {
      "type": "integer"
    },
    "id": {
      "properties": {
        "$oid": {
          "type": "text",
          "fields": {
            "keyword": {
              "ignore_above": 256.0,
              "type": "keyword"
            }
          }
        }
      }
    },
    "position": {
      "type": "integer"
    },
    "played_matches": {
      "type": "integer"
    },
    "win": {
      "type": "integer"
    },
    "team_name": {
      "type": "text",
      "fields": {
        "keyword": {
          "ignore_above": 256.0,
          "type": "keyword"
        }
      }
    },
    "points": {
      "type": "integer"
    }
  }
}
```

- Índice correspondiente a la información de partidos:

* Debido a la extensión de la estructura del índice, ésta ha sido incluida en el *Anexo 1* de este documento como *Estructura del índice ElasticSearch para la búsqueda de partidos.*

Una vez han sido creados los índices para la indexación de documentos, la siguiente etapa de este proceso es el transporte de la información proveniente desde el almacén de datos (MongoDB) a los índices creados en el motor de búsqueda (ElasticSearch).

Para la realización de esta tarea, como ya se ha indicado en secciones anteriores, la tecnología utilizada es Apache Nifi, un software diseñado para la automatización del flujo de transporte de datos entre diferentes sistemas. De esta forma, se creará un proceso Nifi (*Figura 49*) del que podemos distinguir las siguientes etapas:

1. Conexión con MongoDB y obtención de la información.

En esta etapa se establece la conexión con la base de datos desde la que queremos obtener la información, indicando la ruta de conexión con el servidor MongoDB, la base de datos que contiene las colecciones con la información a extraer y la colección desde la que se van a extraer los documentos. Además de establecer la configuración con las entidades mencionadas anteriormente, es necesario indicar una serie de parámetros de configuración como la estrategia de cola de documentos, el número de tareas concurrentes del que se va a disponer, el horario de ejecución (automatización)…

A continuación, en la *Figura 46* mostramos la configuración utilizada para la obtención de la información de los partidos almacenados, siendo reutilizable para la obtención de la información de clasificaciones sustituyendo la colección de acceso por la que almacena dicha información.

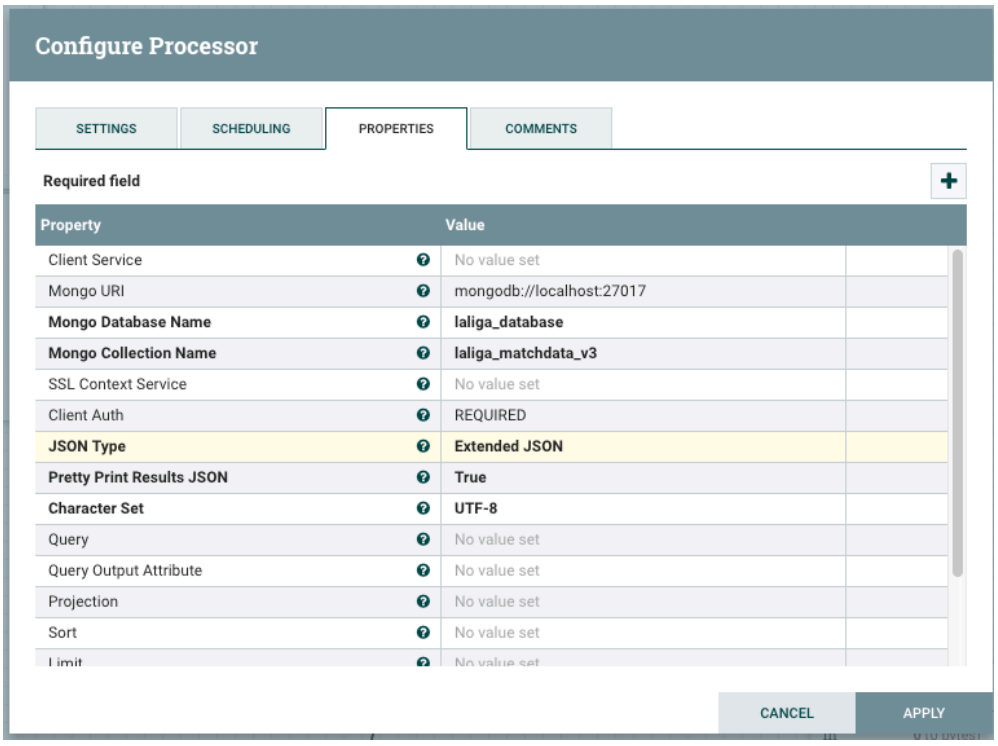


Figura 46.- Configuración Nifi - Proceso de obtención de datos

2. Transformación de la estructura del documento.

Tras la obtención de los documentos provenientes de la colección MongoDB es necesario realizar un pequeño proceso de transformación debido a que la tecnología ElasticSearch no permite el almacenamiento de documentos con el campo *_id* generado por MongoDB.

Debido a este motivo, se generará un nuevo nodo de transformación de texto en el proceso de transporte encargado de eliminar el carácter '_' del documento. En este nodo habrá que indicar el literal que deseamos sustituir y el nuevo literal resultante. Además de estos parámetros de sustitución, será necesario indicar la estrategia de sustitución y el modo de evaluación del texto como parámetros de configuración.

A continuación, en la *Figura 47* mostramos la configuración de parámetros que se ha utilizado en este proceso de transformación:

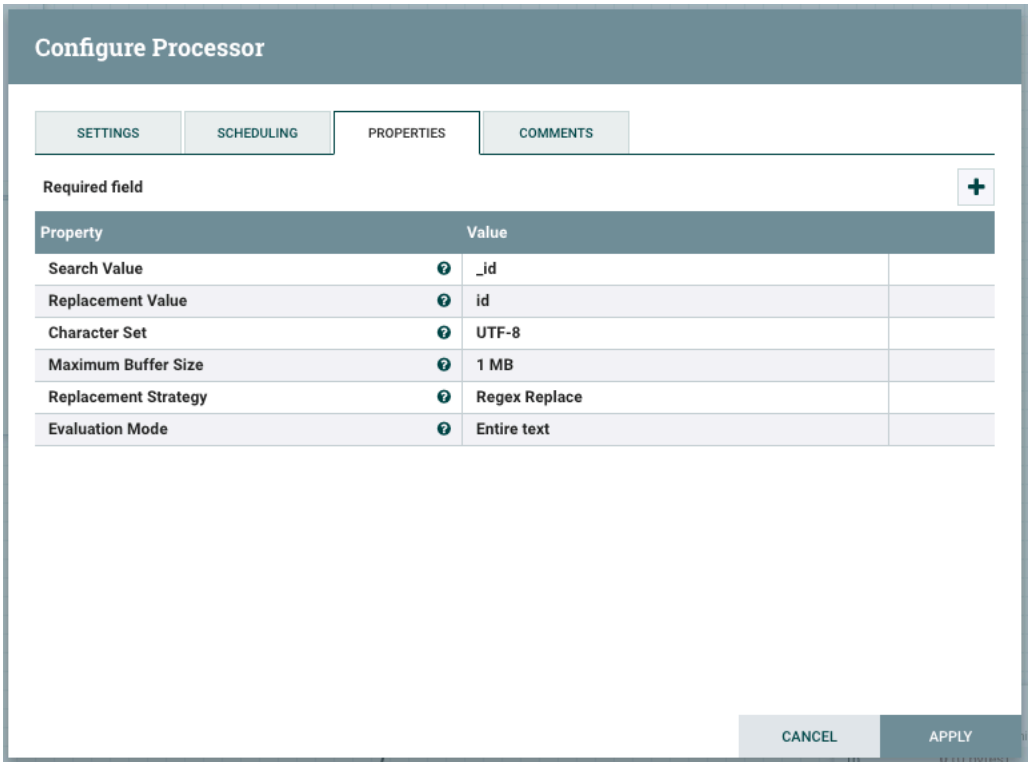


Figura 47.- Configuración Nifi - Proceso de transformación de datos

3. Conexión con ElasticSearch e inserción de la información.

Tras haber transformado el campo *_id* del documento, podremos realizar la inserción en los índices de búsqueda de ElasticSearch creados anteriormente sin problemas. Por lo tanto, deberemos crear un nuevo nodo en nuestro proceso de transformación encargado de la inserción e indexación de los documentos en el motor de búsqueda.

En este último nodo del proceso Nifi deberemos indicar la dirección del servidor ElasticSearch dónde se han creado los índices de búsqueda, el nombre del índice en concreto y el tipo de índice, en este caso de búsqueda. Al igual que en los nodos descritos anteriormente, existen diferente parámetros de configuración como los tiempos máximos de conexión y respuesta, la codificación de la información, el número máximo de documentos en la cola…

A continuación, en la *Figura 48* mostramos la configuración utilizada para la indexación de la información de los partidos, siendo reutilizable (al igual que el nodo de extración) para la indexación de la información de clasificaciones sustituyendo el índice de destino correspondiente.

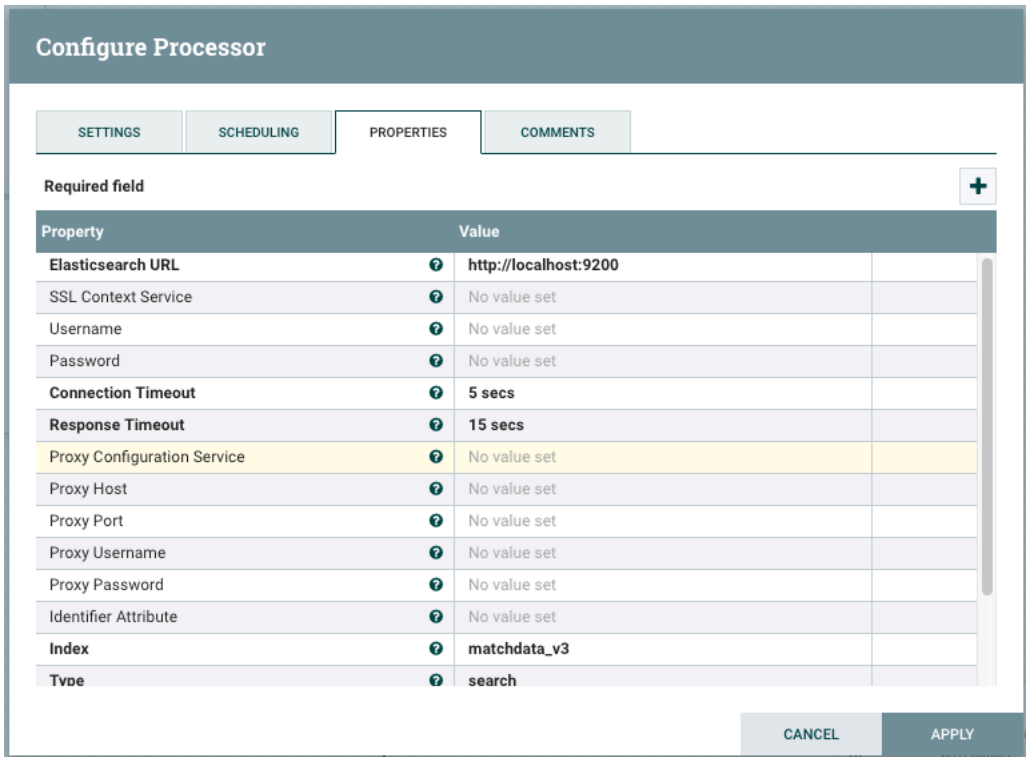


Figura 48.- Configuración Nifi - Proceso de indexación de datos

Una vez descritos los principales nodos del flujo de transformación Nifi, en la *Figura 49* se muestra la estructura completa de dicho proceso. Es importante destacar que para la etapa de extración de datos desde MongoDB es necesario incluir un nuevo nodo en caso de que aparezcan posibles errores, de esta forma se creará un nuevo nodo encargado de almacenar de forma local en el sistema los documentos que habrían fallado y el log correspondiente indicando la causa del fallo, esta operación se realiza mediante el nodo *PutFile*.

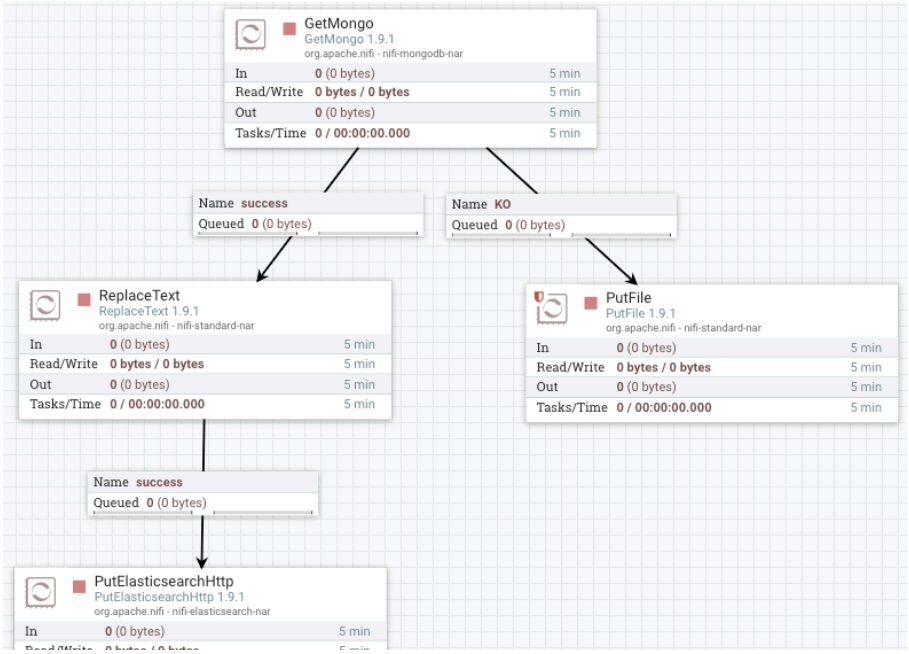


Figura 49.- Configuración Nifi – Flujo completo para la indexación de la información

5. Visualización

Por último, la etapa final en el desarrollo del proyecto comprende la visualización de la información extraída, transformada, almacenada e indexada. En secciones anteriores se ha expuesto la importancia de disponer de una fase de visualización en cualquier proceso de análisis de datos, permitiendo al usuario obtener una visión mucho más amplia del significado de esta información.

La herramienta utilizada para desarrollar la etapa de visualización es Kibana, ya descrita en secciones anteriores. La principal ventaja de esta herramienta es su rendimiento debido a que se encuentra directamente integrada en el entorno ElasticSearch, aprovechando las ventajas y la potencia del motor de búsqueda a la hora de representar y filtrar la información.

En esta etapa de visualización van a elaborarse un conjunto de diferentes cuadros de mando (dashboards), cada uno con un enfoque diferente:

- Información asociada a los árbitros del partido.
- Información asociada a la influencia de los jugadores en las estadísticas del equipo.
- Información asociada a las estadísticas de un partido concreto.
- Información asociada a la evolución en las clasificaciones para una temporada concreta.
- Información asociada a la evolución de un equipo en la clasificación a lo largo de diferentes temporadas.

Con este conjunto de cuadros de mandos espera cubrirse un amplio espectro en el análisis de la información capturada. Es importante que la creación de estadísticas y de diferentes visualizaciones es casi infinita en función de diferentes combinaciones, por lo que tratará de abordarse la mayor cantidad de funcionalidad posible.

Una vez indicados los diferentes cuadros de mandos de contenidos en la herramienta, pasamos al análisis de sus visualizaciones, elementos de filtrado y métricas de cada uno de ellos:

- **Información asociada a los árbitros del partido** (*LaLiga Matchdata: Referee Overview*).

  Este cuadro de mando proporcionará información asociada a un árbitro concreto para una temporada en concreto. Es importante destacar que, debido a la estructuración de la información, se realizará una distinción en las métricas en función de la información asociada al equipo local y la información asociada al equipo visitante. De esta forma, pasamos a enumerar el conjunto de objetos de los que se compone el cuadro de mando:

  - Panel de filtrado. Permitirá al usuario la selección de la temporada de la que se quiere obtener información, el número de jornadas incluidas en el análisis y el árbitro del que se desean obtener las estadísticas.

  - Número de partidos arbitrados por el árbitro seleccionado.

  - Métricas sobre el conjunto de *faltas señaladas, tarjetas amarillas, tarjetas rojas* y *penaltis* señalados a lo largo de los partidos arbitrados. Esta información se dividirá en equipos locales y equipos visitantes.

  - Métricas con la el promedio de faltas señaladas, tarjetas amarillas y tarjetas rojas tanto para equipos locales como para equipos visitantes.

  - Top 5 equipos a los que se les ha señalado el mayor número de faltas, tanto para equipos locales como para visitantes.

  - Top 5 equipos a los que se les ha señalado el menor número de faltas (juego limpio), tanto para equipos locales como para visitantes.

  - Top 5 equipos a los que se les ha mostrado el mayor número de tarjetas amarillas, tanto para equipos locales como para visitantes.

- Top 5 equipos a los que se les ha mostrado el menor número de tarjetas amarillas, tanto para equipos locales como para visitantes.

- Top 10 con los jugadores que han sido amonestados con tarjeta amarilla un mayor número de veces, tanto para equipos locales como para visitantes.

- Información en formato tabla con el conjunto de todos los equipos arbitrados incluyendo *faltas, tarjetas amarillas, tarjetas rojas* y *penaltis,* tanto para equipos locales como para visitantes.

- Top 1 al equipo local y al equipo visitante a los que les han sido señalados el mayor número de penaltis.

A continuación, en las *Figuras 50* y *51* mostramos el contenido del cuadro de mando descrito anteriormente:

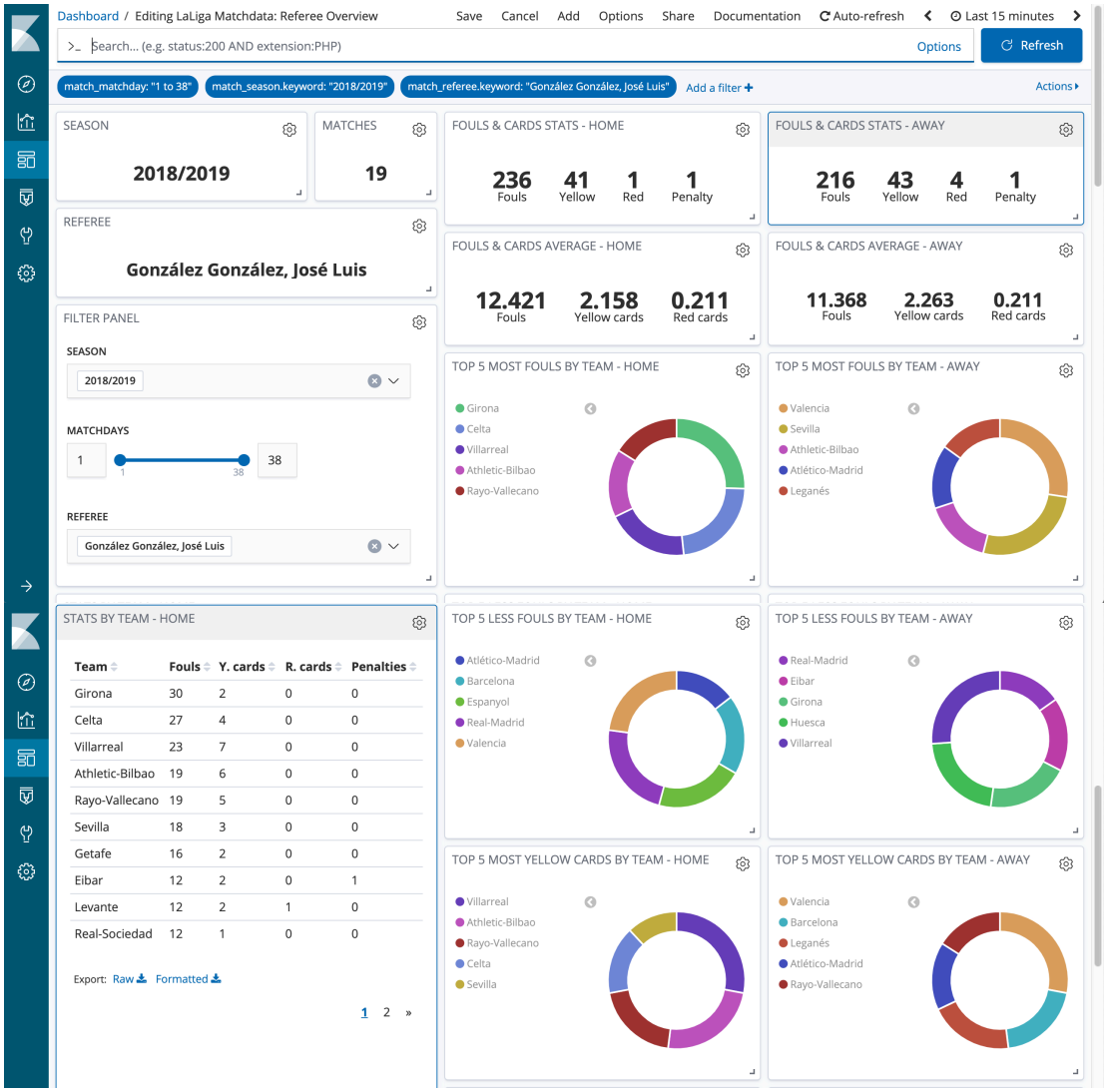


Figura 50.- Kibana – LaLiga Matchdata: Referee Overview – Parte 1

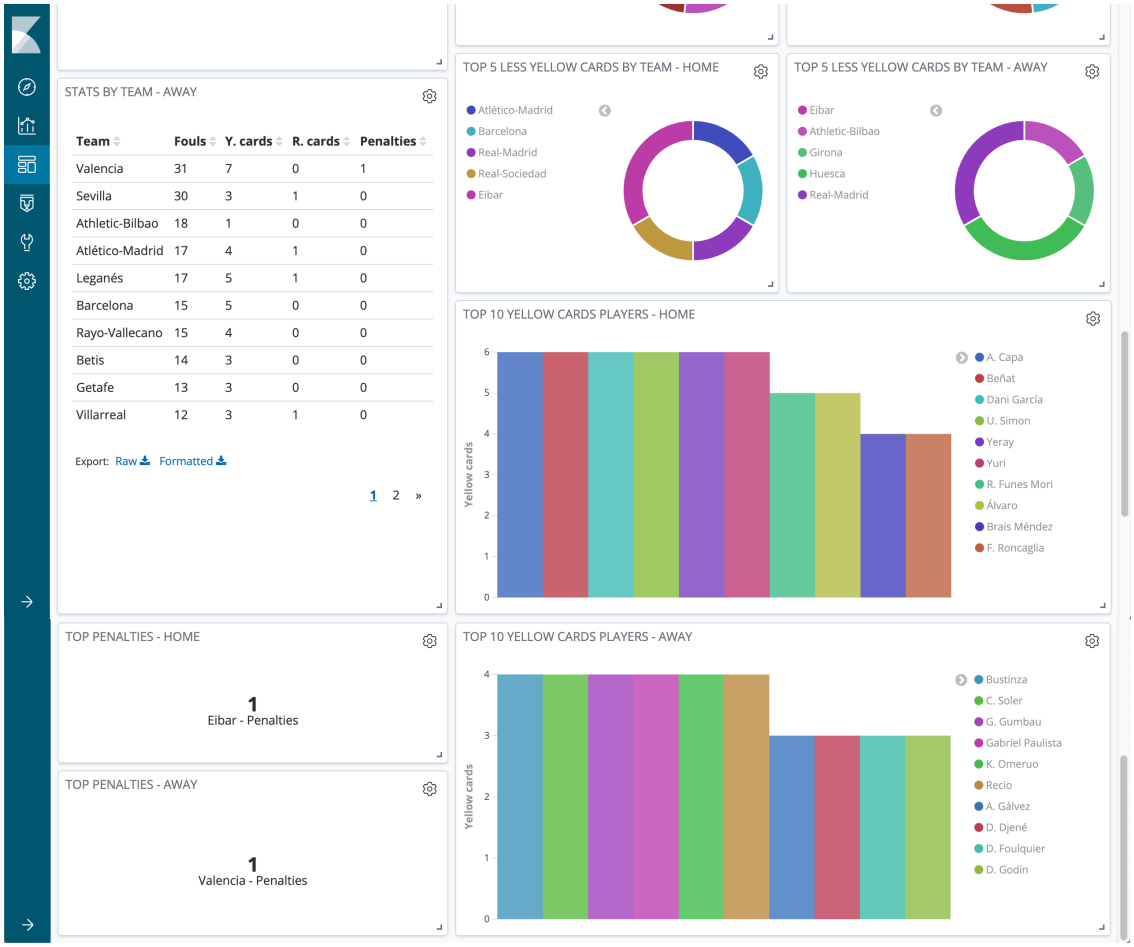


Figura 51.- Kibana – LaLiga Matchdata: Referee Overview – Parte 2

- **Información asociada a la influencia de los jugadores en las estadísticas del equipo** (*LaLiga Matchdata: Home Team Player Influence*).

Este cuadro de mando proporcionará información asociada a la influencia de los jugadores que han tomado parte en el once titular del partido en las estadísticas generales del equipo. Este cuadro de mando ha sido desarrollado únicamente para los partidos en los que el equipo jugaba en casa, siendo totalmente equivalente al visitante únicamente sustituyendo los campos objeto de estudio. A continuación, pasamos a enumerar el conjunto de objetos de los que se compone el cuadro de mando:

- Panel de filtrado. Permitirá al usuario seleccionar el conjunto de temporadas sobre el que se desea realizar el análisis de estadísticas, el equipo del que se desea obtener la información y los jugadores a incluir/no incluir en la generación de dichas estadísticas.

- Conjunto de estadísticas relacionadas al ataque del equipo: *goles marcados, asistencias, disparos a puerta, disparos fuera, tiros al palo, lanzamientos de esquina, faltas recibidas* y *fueras de juego señalados.*

- De forma similar al conjunto de estadísticas anterior, se muestra el conjunto de estadísticas relacionadas a la defensa del equipo: *goles encajados, paradas del portero, faltas cometidas, tarjetas amarillas* y *tarjetas rojas.*

- Nube de palabras con el nombre de los 10 jugadores que mayor cantidad de goles han marcado.

A continuación, en la *Figura 52* mostramos contenido del cuadro de mando descrito anteriormente:

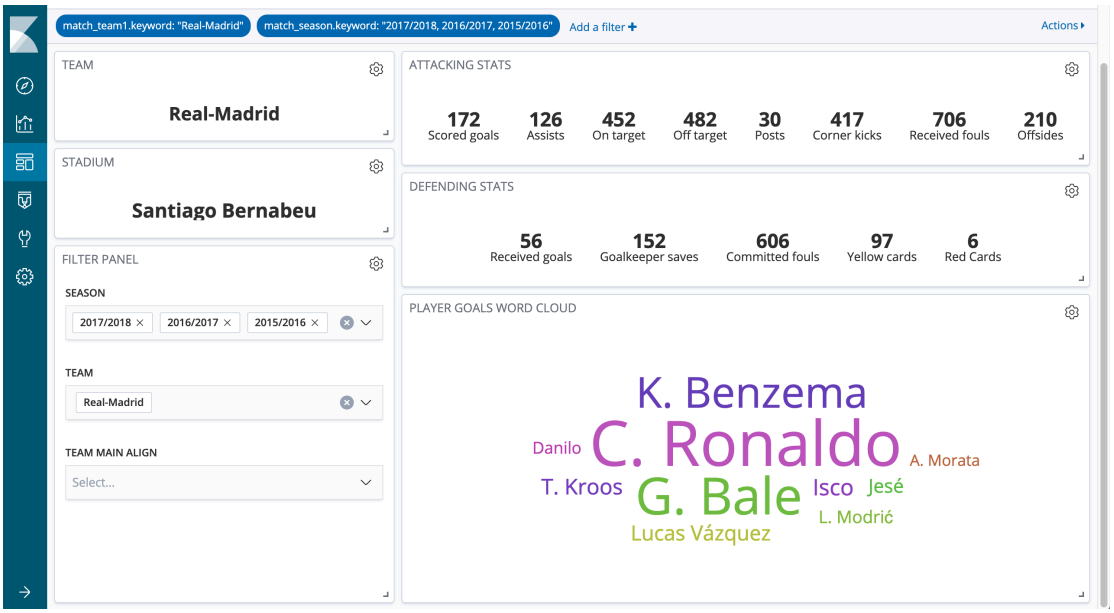


Figura 52.- Kibana – LaLiga Matchdata: Home Team Player Influence

- **Información asociada a las estadísticas de un partido concreto** (*LaLiga Matchdata: Match Overview*).

Este cuadro de mando proporcionará información asociada al conjunto de estadísticas recogidas durante un partido en concreto. A continuación, pasamos a enumerar el conjunto de objetos de los que se compone el cuadro de mando:

- Panel de filtrado. Permitirá al usuario seleccionar la temporada en la que se desea analizar el partido, el equipo local del partido y el equipo visitante.
- Estadio en el que se celebra el partido.
- Fecha de celebración del partido.
- Entrenadores de cada uno de los equipos del partido.
- Formaciones de cada uno de los equipos del partido.
- Alineaciones titulares y suplentes de cada uno de los equipos del partido.
- Goles anotados por cada uno de los equipos del partido.
- Conjunto de estadísticas asociadas a los goles por cada uno de los equipos del partido (goles estándar, goles de falta, goles de penalti, goles cancelados, goles en propia puerta y tiros al palo.
- Comparación entre las estadísticas del partido para cada equipo (*disparos totales, disparos a puerta, disparos fuera, posesión del balón, paradas del portero, saques de esquina, fueras de juego, faltas* y *tarjetas).*

A continuación, en la *Figura 53* mostramos el contenido del cuadro de mando descrito anteriormente:

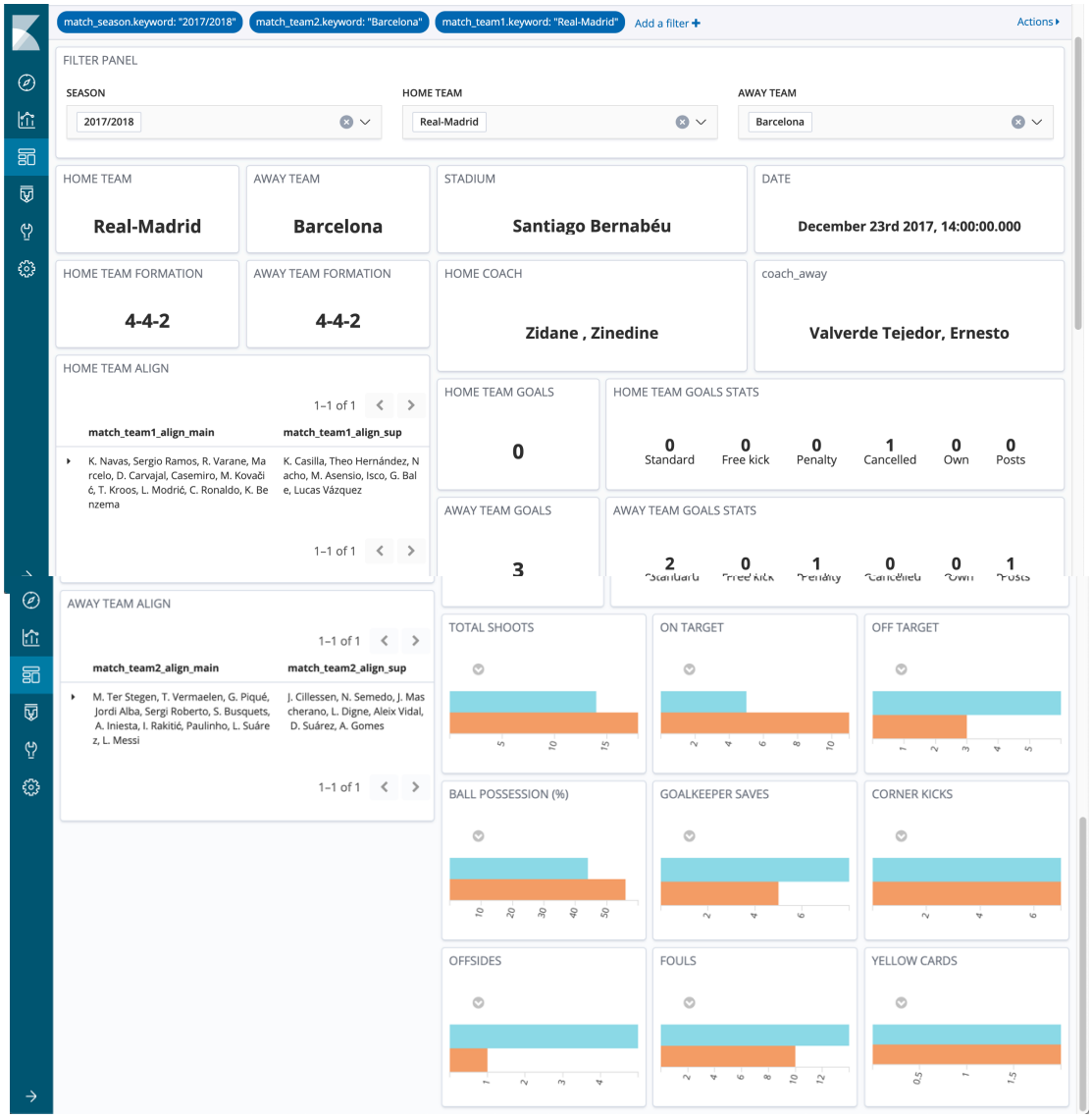


Figura 53.- Kibana – LaLiga Matchdata: Match Overview

- **Información asociada a la evolución en las clasificaciones para una temporada concreta** (*LaLiga Standings: Season Summary*).

Este cuadro de mando proporcionará la información relacionada a las clasificaciones a lo largo de las diferentes jornadas de una temporada concreta. A continuación, pasamos a enumerar el conjunto de objetos de los que se compone el cuadro de mando:

  - Panel de filtrado. Permite seleccionar la temporada de la que se desea obtener la información.
  - Equipo ganador de la temporada.
  - Estadísticas del equipo ganador de la temporada (partidos ganados, empatados y perdidos).
  - Puntos del equipo ganador.
  - Equipo con mayor número de victorias a lo largo de la temporada.
  - Equipo con mayor número de empates a lo largo de la temporada.
  - Equipo con mayor número de derrotas a lo largo de la temporada.
  - Evolución del primer clasificado a lo largo de cada jornada de la temporada.
  - Evolución de los tres equipos en descenso a lo largo de cada jornada de la temporada.
  - Información en formato tabla con los datos de la clasificación al final de la temporada
  - Equipo con mayor/menor número de goles anotados a lo largo de la temporada.
  - Equipo con mayor/menor número de goles encajados a lo largo de la temporada.
  - Top 10 equipos con mayor número de goles anotados a lo largo de la temporada.
  - Top 10 de los equipos con mayor número de goles encajados a lo largo de la temporada.

A continuación, en la *Figura 54* mostramos el contenido del cuadro de mando descrito anteriormente:

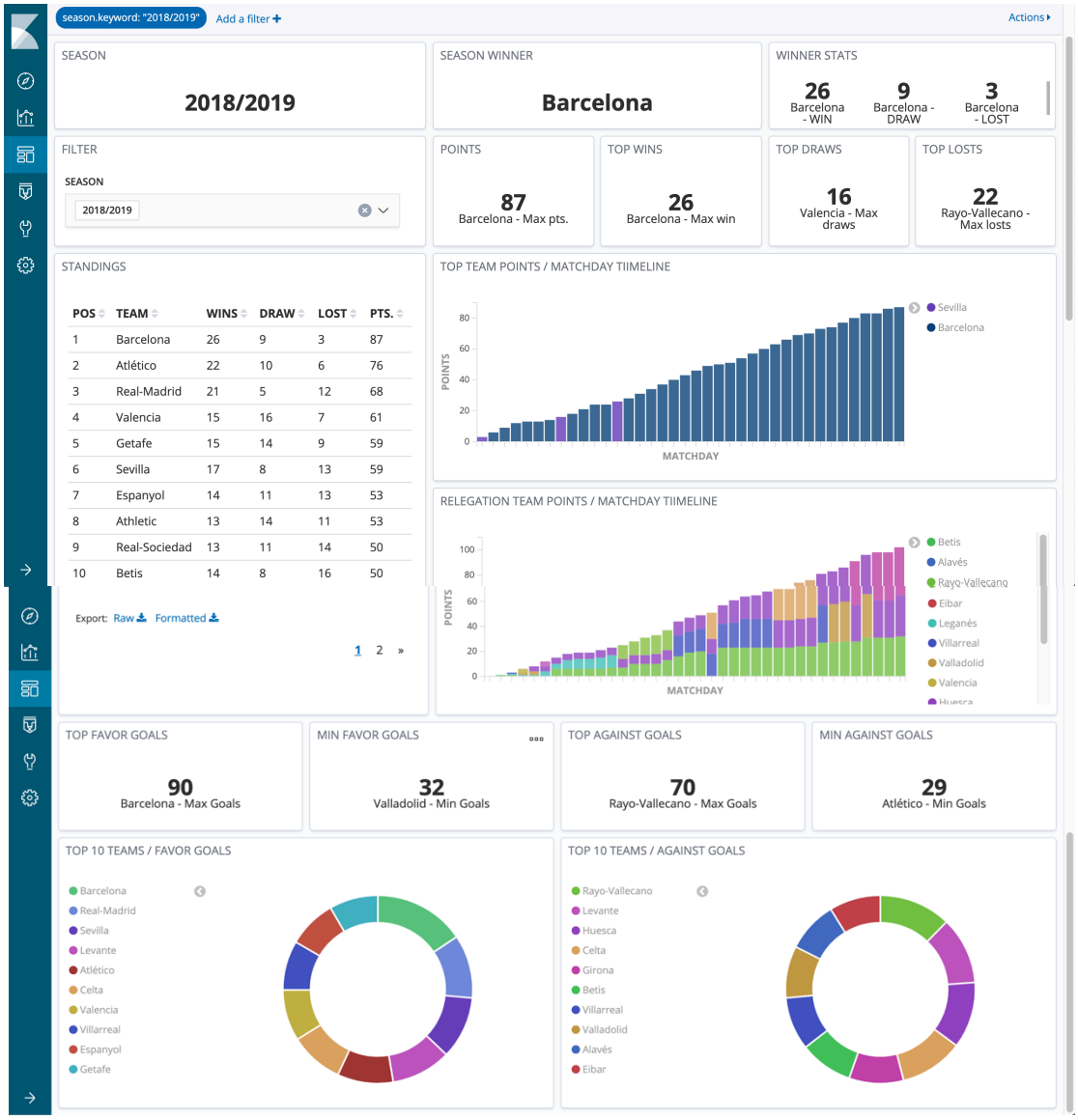


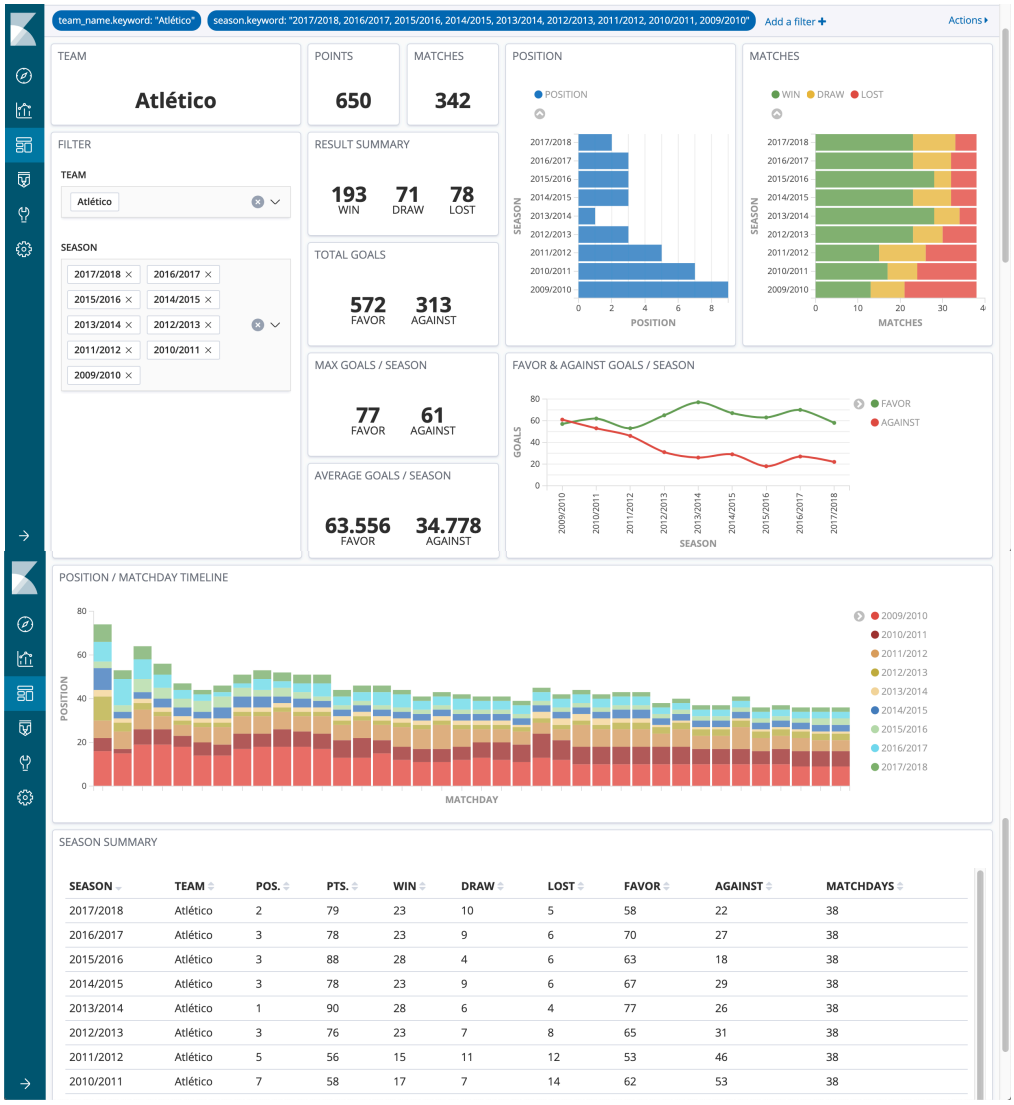
Figura 54.- Kibana – LaLiga Standings: Season Summary

- **Información asociada a la evolución de un equipo en la clasificación a lo largo de diferentes temporadas** (*LaLiga Standings: Team Overview*).

Este cuadro de mando proporcionará la información relacionada a la evolución general del equipo seleccionado a lo largo del conjunto de temporadas seleccionado. A continuación, pasamos a enumerar el conjunto de objetos de los que se compone el cuadro de mando:

  - Panel de filtrado. Permite seleccionar el equipo sobre el que se desea realizar el análisis y el conjunto de temporadas sobre las que realizar el análisis.
  - Puntos totales a lo largo de las temporadas seleccionadas.
  - Partidos totales disputados.
  - Número total de partidos ganados, empatados y perdidos.
  - Número total de goles (anotados/encajados).
  - Máximo número de goles (anotados/encajados).
  - Promedio de goles (anotados/encajados).
  - Evolución en la posición en la clasificación a final de temporada a lo largo del conjunto de temporadas seleccionado.
  - Evolución en el conjunto de partidos *ganados, perdidos* y *empatados* a lo largo del conjunto de temporadas seleccionadas.
  - Evolución de los goles anotados/encajados a lo largo de las temporadas seleccionadas.
  - Evolución en la posición en la clasificación a lo largo de cada jornada del conjunto de temporadas seleccionadas.
  - Información en formato tabla sobre el resultado a final de temporada para el conjunto de temporadas seleccionadas.

A continuación, en la *Figura 55* mostramos el contenido del cuadro de mando descrito anteriormente:



SEASON	TEAM	POS.	PTS.	WIN	DRAW	LOST	FAVOR	AGAINST	MATCHDAYS
2017/2018	Atlético	2	79	23	10	5	58	22	38
2016/2017	Atlético	3	78	23	9	6	70	27	38
2015/2016	Atlético	3	88	28	4	6	63	18	38
2014/2015	Atlético	3	78	23	9	6	67	29	38
2013/2014	Atlético	1	90	28	6	4	77	26	38
2012/2013	Atlético	3	76	23	7	8	65	31	38
2011/2012	Atlético	5	56	15	11	12	53	46	38
2010/2011	Atlético	7	58	17	7	14	62	53	38

Figura 55.- Kibana – LaLiga Standings: Team Overview

4.2 Diseño

En esta sección se especificará la arquitectura de la que se compone el proyecto desarrollado a lo largo del documento.

La arquitectura del proyecto comprende la definición de los componentes lógicos del sistema y como estos se relacionan entre ellos. Al igual que durante la exposición de las secciones previas, la arquitectura lógica del sistema se verá influenciada por las diferentes fases del proyecto: Extracción, Transformación, Ingesta, Indexación y Visualización:

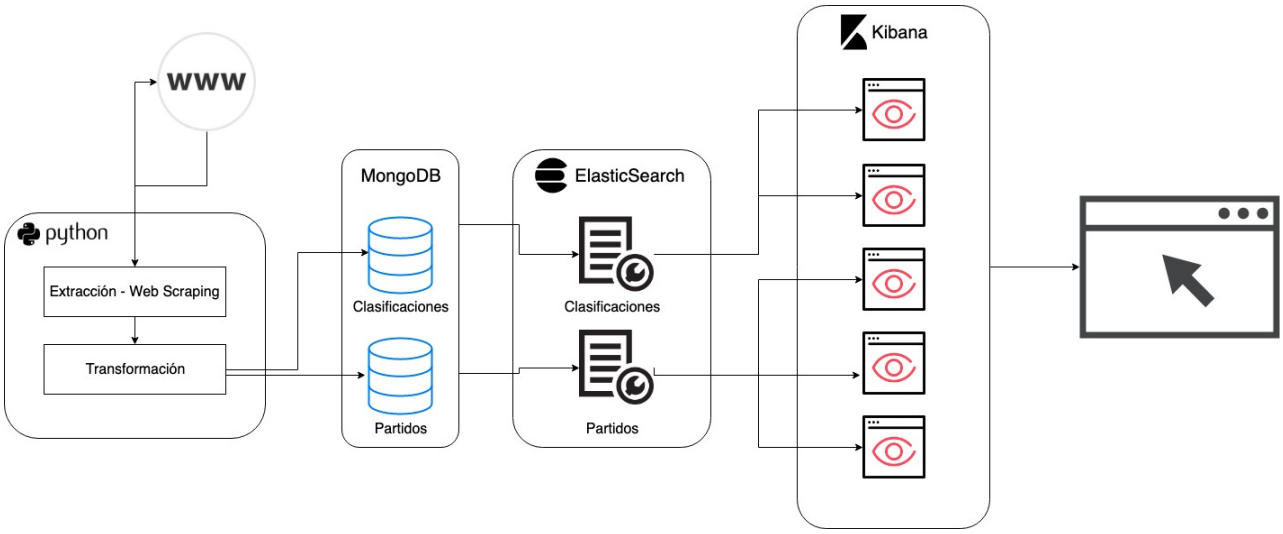


Figura 56.- Arquitectura del sistema

Se describirá brevemente la relación entre las fases de la figura anterior ya que estas relaciones han sido expuestas a lo largo de todo el documento.

1. Módulo Python de extracción y transformación.

   Desde este módulo se establece la conexión con el sitio web desde el que va a obtenerse la información. Una vez extraída esta información, desde el mismo módulo Python es transformada (Edge Computing) y estructurada en diccionarios Python listos para ser ingestados en el almacén de datos.

2. Módulo almacén de datos

   Una vez transformada la información, ésta es ingestada desde el módulo anterior en el servidor MongoDB en dos colecciones separadas, una para la información de clasificaciones y otra para la información de partidos.

3. Módulo de indexación

Mediante la herramienta Apache Nifi, se establece un flujo de datos encargado de recoger la información del almacén de datos (MongoDB) e insertar todos los documentos de las colecciones en los índices correspondientes en el motor de búsqueda (ElasticSearch).

4. Módulo de visualización

Una vez contamos con todos los documentos indexados en el motor de búsqueda, desde la herramienta de visualización Kibana se desarrollan los diferentes cuadros de mando de los que se va a componer la etapa de visualización del proyecto.

5. Visualización

Por último, un usuario accederá a los cuadros de mando de Kibana a través de la dirección del servidor desde un navegador web.

Una vez desarrollados los apartados anteriores, damos por finalizada la etapa de análisis y diseño de la herramienta implementada.

5 Conclusiones y líneas de trabajo futuro

En esta última sección del documento se expondrán las conclusiones derivadas de la realización del proyecto.

Como ya se desarrolló a lo largo del apartado correspondiente al estudio del estado del arte de este documento, existen multitud de sitios web con información de estadísticas deportivas. El punto débil de este conjunto de sitios web es la etapa de visualización, ya que aunque la cantidad de estadísticas e información es correcta, la mayoría de sitios representan esta información en tablas de datos sin gráficos ni comparación entre estadísticas. Debido a esta necesidad en la visualización de la información surgió la idea del desarrollo de este proyecto, siendo la principal ventaja de este desarrollo la capacidad de incluir tanto las métricas como las visualizaciones deseadas de forma que se obtenga un análisis personalizado y único.

Otro aspecto a destacar es la creación de un sistema escalable en cuanto a cantidad de datos, ya que aunque para este desarrollo se ha centrado en la Liga Profesional de Fútbol española, puede ser aplicable, con una serie de pequeñas modificaciones, a cualquier liga del mundo, lo que nos permitiría disponer de un análisis muy completo y complejo y aun así contar con un buen rendimiento en la visualización gracias al motor de búsqueda de ElasticSearch.

5.1 Dificultades

Han surgido una serie de dificultades a la hora de implementar el proyecto, casi todas debidas a la falta de experiencia en las diferentes tecnologías utilizadas. A lo largo de mi formación tanto académica como laboral nunca había tenido oportunidad de trabajar con la mayoría de las tecnologías utilizadas (MongoDB, Kibana, Apache Nifi…), por lo que ha sido necesario dedicar un importante periodo de tiempo al manejo y la administración de estas tecnologías.

Otra dificultad importante a tener en cuenta en el desarrollo del proyecto ha sido la complejidad en la estructuración de la información y en el número de etapas en las que se compone (Extracción, Transformación, Ingesta, Indexación y Visualización). Este conjunto de etapas implica que obtener un fallo a la hora de visualizar un dato en concreto supone la realización del flujo de trabajo completo, volviendo a pasar nuevamente por cada una de las etapas. Un ejemplo de esta problemática ha sido expuesto en la sección de análisis, con el cambio en las estructuras de almacenamiento de datos.

Otro problema a destacar ha sido el entorno de desarrollo para la herramienta. En un principio, se comenzó el desarrollo del proyecto en una máquina proporcionada por la universidad dotada de 16GB de memoria y 80GB de disco. Pero dado que el rendimiento de este entorno no era capaz de cubrir las necesidades básicas para el correcto funcionamiento del proyecto, puesto que ni siquiera permitía la correcta ejecución de las aplicaciones de interfaz gráfica para las aplicaciones de

MongoDB y ElasticSearch. De esta forma, se decidió desarrollar el proyecto en mi ordenador personal dotado de 8GB de memoria y 512GB de disco, obteniendo un rendimiento impecable en el conjunto de todos los procesos del sistema.

5.2 Conocimientos adquiridos

Si bien ha sido complejo el desarrollo del proyecto y la toma de contacto con las diferentes tecnologías utilizadas, la cantidad de conocimientos adquiridos ha sido excepcional, desarrollando mi conocimiento a nivel de programación, en el ámbito de las tecnologías Big Data y las tecnologías de visualización.

En este aspecto, me gustaría destacar el impacto de las utilización de técnicas de web scraping, ya que nos permiten extraer información de prácticamente cualquier sitio web siempre y cuando no se encuentre expresamente prohibido en los términos del sitio web.

5.3 Líneas de trabajo futuro

El proyecto expuesto en este documento cuenta con diversas ampliaciones futuras, sobre todo en el ámbito de la visualización debido a la cantidad de información recogida. Las principales líneas a seguir son:

- Creación de una base de datos auxiliar con toda la información del equipo arbitral para cada partido, de forma que puedan incluirse asistentes de línea, cuarto árbitro y árbitro asistente de video (VAR) en el análisis de la información.

- Agrupación de la información de equipos locales/visitantes en una estructura conjunta de forma que no sea necesario separar el análisis a la hora de visualizar.

- Ampliación del conjunto de ligas de futbol sobre las que realizar el análisis con las ligas de futbol más importantes del mundo (Premier League, Ligue 1, Calcio…).

- Ampliación del número de deportes objeto de análisis, incluyendo baloncesto y tenis.

- Creación de un clúster con diferentes nodos para almacenar la información cuando el tamaño de la misma crezca considerablemente.

- Automatización del proceso de forma que dispongamos prácticamente en streaming de la información actualizada a nivel de partido.

Anexo 1 - Figuras y estructuras adicionales

5. Tabla 4.- Diccionario de datos para partidos

Diccionario	Atributo	Tipo de dato	Descripción
Partidos	match_id	String	Identificador único de la información asociada al partido capturado. Se compone a partir de la concatenación de los equipos que disputan el partido y la temporada a la que pertenece.
	match_date	String	Fecha de celebración del partido capturado.
	match_season	String	Temporada a la que pertenece la información capturada.
	match_matchday	Int32	Jornada a la que pertenece la información capturada.
	match_referee	String	Árbitro principal del partido.
	match_stadium	String	Estadio en el que se celebra el partido.
	match_attendance	Int32	Número de asistentes al partido.
	match_score	String	Resultado final del partido.
	match_team1	String	Nombre del equipo local del partido.
	match_team1_coach	String	Entrenador del equipo local del partido.
	match_team1_formation	String	Formación del equipo local del partido.
	match_team1_align_main	Array[Strings]	Alineación titular del equipo local del partido.
	match_team1_align_sup	Array[Strings]	Alineación suplente del equipo local del partido.
	match_team1_possesion	Double	Porcentaje de posesión del balón por parte del equipo local del partido.
	match_team1_goals	Int32	Número total de goles anotados por el equipo local del partido.
	match_team1_assists	Int32	Número total de asistencias dadas por el equipo local del partido.

	match_team1_assists_player	Array[Strings]	Información de los jugadores del equipo local que han dado asistencias en el partido.
	match_team1_assists_minute	Array[Int32]	Información de los minutos del equipo local en los que se han dado las asistencias en el partido.
	match_team1_goals_player	Array[Strings]	Información de los jugadores del equipo local que han marcado gol en el partido.
	match_team1_goals_minute	Array[Int32]	Información de los minutos del equipo local en los que se han marcado goles en el partido.
	match_team1_goal_freekick_player	Array[Strings]	Información de los jugadores del equipo local que han marcado gol de falta en el partido.
	match_team1_goal_freekick_minute	Array[Int32]	Información de los minutos del equipo local en los que se han marcado goles de falta en el partido.
	match_team1_penalty_missed_player	Array[Strings]	Información de los jugadores del equipo local que han fallado un penalti en el partido.
	match_team1_penalty_missed_minute	Array[Int32]	Información de los minutos del equipo local en los que se han fallado penaltis en el partido.
	match_team1_penalty_saved_player	Array[Strings]	Información de los jugadores del equipo local que han detenido un penalti en el partido.
	match_team1_penalty_saved_minute	Array[Int32]	Información de los minutos del equipo local en los que se han detenido penaltis en el partido.
	match_team1_penalty_saved2_player	Array[Strings]	Información de los jugadores del equipo visitante que han detenido un penalti en el partido.
	match_team1_penalty_saved2_minute	Array[Int32]	Información de los minutos del equipo local en los que se les ha parado un penalti por parte del otro equipo en el partido.

	match_team1_totalshoots	Int32	Número total de disparos realizados por el equipo local del partido.
	match_team1_shootsontarget	Int32	Número total de disparos a puerta realizados por el equipo local del partido.
	match_team1_shootsofftarget	Int32	Número total de disparos fuera realizados por el equipo local del partido.
	match_team1_goalkeepersaves	Int32	Número total de paradas del portero por el equipo local del partido.
	match_team1_cornerkicks	Int32	Número total de saques de esquina por el equipo local del partido.
	match_team1_offsides	Int32	Número total de fueras de juego por el equipo local del partido.
	match_team1_posts	Int32	Número total de disparos al palo por el equipo local del partido.
	match_team1_post_player	Array[Strings]	Información de los jugadores del equipo local que han disparado al palo en el partido.
	match_team1_post_minute	Array[Int32]	Información de los minutos del equipo local en los que se ha disparado al palo en el partido.
	match_team1_fouls	Int32	Número total de faltas cometidas por el equipo local del partido.
	match_team1_yellowcards	Int32	Número total de tarjetas amarillas señaladas al equipo local del partido.
	match_team1_yellowcard_player	Array[Strings]	Información de los jugadores del equipo local que han recibido tarjeta amarilla en el partido.
	match_team1_yellowcard_minute	Array[Int32]	Información de los minutos del equipo local en los que se han señalado tarjeta amarilla en el partido.
	match_team1_redcards	Int32	Número total de tarjetas rojas señaladas al equipo local del partido.
	match_team1_card_red_player	Array[Strings]	Información de los jugadores del equipo local

			que han recibido tarjeta roja en el partido.
	match_team1_card_red_minute	Array[Int32]	Información de los minutos del equipo local en los que se han señalado tarjeta roja en el partido.
	match_team1_card_red2yellow_player	Array[Strings]	Información de los jugadores del equipo local que han recibido segunda tarjeta amarilla en el partido.
	match_team1_card_red2yellow_minute	Array[Int32]	Información de los minutos del equipo local en los que se han señalado tarjeta roja por segunda amarilla en el partido.
	match_team1_injuries	Int32	Número total de lesiones del equipo local del partido.
	match_team1_injury_player	Array[Strings]	Información de los jugadores del equipo local lesionados durante el partido.
	match_team1_injury_minute	Array[Int32]	Información de los minutos del equipo local en los que se han producido lesiones en el partido.
	match_team1_subs	Int32	Número total de sustituciones realizadas por el equipo local del partido.
	match_team1_subs_in_player	Array[Strings]	Información de los jugadores del equipo local que han sido sustituidos en el partido.
	match_team1_subs_in_minute	Array[Int32]	Información de los minutos del equipo local en los que se han producido sustituciones en el partido.
	match_team1_subs_out_player	Array[Strings]	Información de los jugadores suplentes del equipo local que han entrado en el partido.
	match_team1_subs_out_minute	Array[Int32]	Información de los minutos del equipo local en los que se han producido sustituciones en el partido.
	match_team2	String	Nombre del equipo visitante del partido.
	match_team2_coach	String	Entrenador del equipo visitante del partido.

	match_team2_formation	String	Formación del equipo visitante del partido.
	match_team2_align_main	Array[Strings]	Alineación titular del equipo visitante del partido.
	match_team2_align_sup	Array[Strings]	Alineación suplente del equipo visitante del partido.
	match_team2_possesion	Double	Porcentaje de posesión del balón por parte del equipo visitante del partido.
	match_team2_goals	Int32	Número total de goles anotados por el equipo visitante del partido.
	match_team2_assists	Int32	Número total de asistencias dadas por el equipo visitante del partido.
	match_team2_assists_player	Array[Strings]	Información de los jugadores del equipo visitante que han dado asistencias en el partido.
	match_team2_assists_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han dado las asistencias en el partido.
	match_team2_goals_player	Array[Strings]	Información de los jugadores del equipo visitante que han marcado gol en el partido.
	match_team2_goals_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han marcado goles en el partido.
	match_team2_goal_freekick_player	Array[Strings]	Información de los jugadores del equipo visitante que han marcado gol de falta en el partido.
	match_team2_goal_freekick_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han marcado goles de falta en el partido.
	match_team2_penalty_missed_player	Array[Strings]	Información de los jugadores del equipo visitante que han fallado un penalti en el partido.
	match_team2_penalty_missed_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han fallado penaltis en el partido.
	match_team2_penalty_saved_player	Array[Strings]	Información de los jugadores del equipo

			visitante que han detenido un penalti en el partido.
	match_team2_penalty_saved_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han detenido penaltis en el partido.
	match_team2_penalty_saved2_player	Array[Strings]	Información de los jugadores del equipo local que han detenido un penalti en el partido.
	match_team2_penalty_saved2_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se les ha parado un penalti por parte del otro equipo en el partido.
	match_team2_totalshoots	Int32	Número total de disparos realizados por el equipo visitante del partido.
	match_team2_shootsontarget	Int32	Número total de disparos a puerta realizados por el equipo visitante del partido.
	match_team2_shootsofftarget	Int32	Número total de disparos fuera realizados por el equipo visitante del partido.
	match_team2_goalkeepersaves	Int32	Número total de paradas del portero por el equipo visitante del partido.
	match_team2_cornerkicks	Int32	Número total de saques de esquina por el equipo visitante del partido.
	match_team2_offsides	Int32	Número total de fueras de juego por el equipo visitante del partido.
	match_team2_posts	Int32	Número total de disparos al palo por el equipo visitante del partido.
	match_team2_post_player	Array[Strings]	Información de los jugadores del equipo visitante que han disparado al palo en el partido.
	match_team2_post_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se ha disparado al palo en el partido.
	match_team2_fouls	Int32	Número total de faltas cometidas por el equipo visitante del partido.

	match_team2_yellowcards	Int32	Número total de tarjetas amarillas señaladas al equipo visitante del partido.
	match_team2_yellowcard_player	Array[Strings]	Información de los jugadores del equipo visitante que han recibido tarjeta amarilla en el partido.
	match_team2_yellowcard_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han señalado tarjeta amarilla en el partido.
	match_team2_redcards	Int32	Número total de tarjetas rojas señaladas al equipo visitante del partido.
	match_team2_card_red_player	Array[Strings]	Información de los jugadores del equipo visitante que han recibido tarjeta roja en el partido.
	match_team2_card_red_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han señalado tarjeta roja en el partido.
	match_team2_card_red2yellow_player	Array[Strings]	Información de los jugadores del equipo visitante que han recibido segunda tarjeta amarilla en el partido.
	match_team2_card_red2yellow_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han señalado tarjeta roja por segunda amarilla en el partido.
	match_team2_injuries	Int32	Número total de lesiones del equipo visitante del partido.
	match_team2_injury_player	Array[Strings]	Información de los jugadores del equipo visitante lesionados durante el partido.
	match_team2_injury_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han producido lesiones en el partido.
	match_team2_subs	Int32	Número total de sustituciones realizadas por el equipo visitante del partido.
	match_team2_subs_in_player	Array[Strings]	Información de los jugadores del equipo

			visitante que han sido sustituidos en el partido.
	match_team2_subs_in_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han producido sustituciones en el partido.
	match_team2_subs_out_player	Array[Strings]	Información de los jugadores suplentes del equipo visitante que han entrado en el partido.
	match_team2_subs_out_minute	Array[Int32]	Información de los minutos del equipo visitante en los que se han producido sustituciones en el partido.

Tabla 4.- Diccionario de datos para partidos

6. Estructura del índice ElasticSearch para la búsqueda de partidos:


```
{
  "properties": {
    "match_team2_card_red_player": {
      "type": "text",
      "fields": {
        "keyword": {
          "ignore_above": 256.0,
          "type": "keyword"
        }
      }
    },
    "match_team1_align_main": {
      "type": "text",
      "fields": {
        "keyword": {
          "ignore_above": 256.0,
          "type": "keyword"
        }
      }
    },
    "match_team1_subs_out_minute": {
      "type": "long"
    },
    "match_team2_assist_player": {
      "type": "text",
      "fields": {
        "keyword": {
          "ignore_above": 256.0,
          "type": "keyword"
        }
      }
    },
    "match_matchday": {
      "type": "long"
    },
```

```
"match_team2_subs_in_minute": {
  "type": "long"
},
"match_team1_totalshoots": {
  "type": "long"
},
"match_team2_totalshoots": {
  "type": "long"
},
"match_team1_goal_penalty_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_fouls": {
  "type": "long"
},
"match_team1_goal_own2_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"id": {
  "properties": {
    "$oid": {
      "type": "text",
      "fields": {
        "keyword": {
          "ignore_above": 256.0,
          "type": "keyword"
        }
      }
    }
  }
},
"match_team1_shootsofftarget": {
  "type": "long"
},
"match_team2_goal_freekick_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_offsides": {
```

```
  "type": "long"
},
"match_attendance": {
  "type": "long"
},
"match_team1_posts": {
  "type": "long"
},
"match_team2_penalty_saved_minute": {
  "type": "long"
},
"match_team1_yellowcard_minute": {
  "type": "long"
},
"match_team1_goal_cancelled_minute": {
  "type": "long"
},
"match_team1_penalty_saved2_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_redcards": {
  "type": "long"
},
"match_team2_goal_own2_minute": {
  "type": "long"
},
"match_team1_assist_minute": {
  "type": "long"
},
"match_team1_goal_own_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_penalty_missed_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_card_red2yellow_minute": {
  "type": "long"
},
```

```
"match_team2_penalty_saved2_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_penalty_missed_minute": {
  "type": "long"
},
"match_team2_injury_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_formation": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_align_sup": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_redcards": {
  "type": "long"
},
"match_team2_shootsofftarget": {
  "type": "long"
},
"match_team2_yellowcard_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_injury_player": {
  "type": "text",
```

```
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_yellowcard_minute": {
  "type": "long"
},
"match_team2_goals_minute": {
  "type": "long"
},
"match_team2_subs_out_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_card_red_minute": {
  "type": "long"
},
"match_team2_post_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_post_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_card_red_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_cornerkicks": {
  "type": "long"
},
"match_team1_injuries": {
```

```
  "type": "long"
},
"match_team1_injury_minute": {
  "type": "long"
},
"match_team2_goalkeepersaves": {
  "type": "long"
},
"match_team2_cornerkicks": {
  "type": "long"
},
"match_season": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_goal_penalty_minute": {
  "type": "long"
},
"match_team2_post_minute": {
  "type": "long"
},
"match_team2_subs_out_minute": {
  "type": "long"
},
"match_team1_post_minute": {
  "type": "long"
},
"match_team2_goal_penalty_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_goal_own2_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_goal_own_minute": {
  "type": "long"
},
"match_team1_goal_freekick_minute": {
  "type": "long"
},
```

```
"match_team1_subs_out_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_assist_minute": {
  "type": "long"
},
"match_referee": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_card_red2yellow_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_injury_minute": {
  "type": "long"
},
"match_team2_assists": {
  "type": "long"
},
"match_team2_card_red_minute": {
  "type": "long"
},
"match_team2_subs_in_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_goal_own2_minute": {
  "type": "long"
},
"match_team1_goal_penalty_minute": {
  "type": "long"
},
"match_team2_goals_player": {
  "type": "text",
```

```
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_subs": {
  "type": "long"
},
"match_team2_penalty_saved_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_assist_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_shootsontarget": {
  "type": "long"
},
"match_team2_shootsontarget": {
  "type": "long"
},
"match_team1_possession": {
  "type": "float"
},
"match_team2_card_red2yellow_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_penalty_saved2_minute": {
  "type": "long"
},
"match_team2_fouls": {
  "type": "long"
},
"match_team2_goals": {
  "type": "long"
},
"match_team2_offsides": {
```

```
  "type": "long"
},
"match_team2_yellowcards": {
  "type": "long"
},
"match_team2_possession": {
  "type": "float"
},
"match_team1_coach": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_goalkeepersaves": {
  "type": "long"
},
"match_team1_yellowcards": {
  "type": "long"
},
"match_team2_align_main": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_formation": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_yellowcard_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"matchid": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
```

```
    }
  }
},
"match_team1_goal_cancelled_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_penalty_saved2_minute": {
  "type": "long"
},
"match_team1_penalty_saved_minute": {
  "type": "long"
},
"match_team2_coach": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_posts": {
  "type": "long"
},
"match_team1_subs_in_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_injuries": {
  "type": "long"
},
"match_team1_subs_in_minute": {
  "type": "long"
},
"match_team1_assists": {
  "type": "long"
},
"match_team2_penalty_missed_minute": {
  "type": "long"
},
"match_team2_goal_cancelled_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
```

```
      "type": "keyword"
    }
  }
},
"match_date": {
  "format": "yyyy/MM/dd HH:mm:ss||yyyy/MM/dd||epoch_millis",
  "type": "date"
},
"match_team1_penalty_missed_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_goal_cancelled_minute": {
  "type": "long"
},
"match_team2_goal_own_minute": {
  "type": "long"
},
"match_team1_goals_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_card_red2yellow_minute": {
  "type": "long"
},
"match_team1_goals": {
  "type": "long"
},
"match_team1_penalty_saved_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_stadium": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
```

```
"match_team1_goals_minute": {
  "type": "long"
},
"match_team2_goal_own_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_goal_freekick_minute": {
  "type": "long"
},
"match_score": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_subs": {
  "type": "long"
},
"match_team2": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team1_goal_freekick_player": {
  "type": "text",
  "fields": {
    "keyword": {
      "ignore_above": 256.0,
      "type": "keyword"
    }
  }
},
"match_team2_align_sup": {
  "type": "text",
```

```
      "fields": {
        "keyword": {
          "ignore_above": 256.0,
          "type": "keyword"
        }
      }
    }
  }
}
```

Anexo 2 - Contenidos del CD-ROM

Se entregan dos CD-ROM con el mismo contenido, en el que se almacenan:

1. Memoria del Trabajo Fin de Máster en formato PDF, con el nombre "*TFM_AlvaroMonedero_2019.pdf*".

2. Backup de los índices ElasticSearch, dividido en el índice de clasificaciones y el índice de partidos con los nombres *"elasticsearch_standings_backup"* y *"elasticsearch_matchdata_backup"*. Estos ficheros contienen tanto los esquemas de los índices de ElasticSearch como la lista de documentos que los componen. Por cuestiones de limpieza, estos archivos se encuentran bajo el directorio "*ElasticSearch".*

3. Exportación de los cuadros de mando desarrollados en Kibana. Kibana no permite la opción de exportación de los cuadros de mando, por lo que se ha generado un fichero PDF para cada dashboard desarrollado. Así, bajo el directorio *"Kibana"* se encuentran los siguientes ficheros:
    - *"Kibana_LaLiga Matchdata - Home Team Player Influence.pdf"*
    - *"Kibana_LaLiga Matchdata - Match Overview.pdf"*
    - *"Kibana_LaLiga Matchdata - Referee Overview.pdf"*
    - *"Kibana_LaLiga Standings - Season Summary.pdf"*
    - *"Kibana_LaLiga Standings - Team Overview.pdf"*

4. Código Fuente del programa Python utilizado para la extracción, transformación y almacenamiento de la información. Los diferentes ficheros Python se encuentran bajo el directorio *"Código Fuente"* y se compone de los siguientes archivos:

    - *"_laliga_main.py"* – Fichero con el flujo principal de la aplicación y la inserción en base de datos.
    - *"_laliga_match.py"* – Fichero con la extracción y la transformación de partidos.
    - *"_laliga_standings.py"* – Fichero con la extracción y la transformación de clasificaciones.
    - *"_laliga_teams.py"* – Fichero que contiene la lista de todos los equipos que han sido extraídos.

Referencias Web

- What is Big Data and why it is important in today's world
  - Url: https://ipbsmim.com/big-data-important-todays-world/
  - Autor: International Partnership of Business Schools
  - Última visita: 15 de Julio de 2019

- ¿Por qué Big Data tiene tanta relevancia hoy en día?
  - Url: https://blogs.dxc.technology/2017/04/25/por-que-big-data-tiene-tanta-relevancia-hoy-en-dia/
  - Autor: Jerry Overton (DXC Technology)
  - Última visita: 15 de Julio de 2019

- El Big Data está de moda, ¿y a mí qué?
  - Url: https://blogthinkbig.com/el-big-data-esta-de-moda-y-a-mi-que
  - Autor: Elena Díaz
  - Última visita: 15 de Julio de 2019

- The impact of Big Data, past and future
  - Url: https://www.deloitteforward.nl/en/data-analytics/the-impact-of-big-data-past-and-future/
  - Autor: Can Yurtseven (Deloitte)
  - Última visita: 15 de Julio de 2019

- Google File System paper
  - Url: https://ai.google/research/pubs/pub51
  - Autor: Sanjay Ghemawat, Howard Gobioff & Howard Gobioff (Google)
  - Última visita: 15 de Julio de 2019

- Apache Hadoop
  - Url: https://hadoop.apache.org/
  - Autor: The Apache Software Foundation
  - Última visita: 15 de Julio de 2019

- Why Data Visualization Matters to Your Business
  - Url: https://www.grepsr.com/data-visualization-why-it-matters-to-your-business/?gclid=Cj0KCQjwjMfoBRDDARIsAMUjNZpMsTb7ooZR5dr_5tGg7S1LVvlRKVIJ-ZfH4DRII-FwN2rjjVeLw-MaAtNKEALw_wcB
  - Autor: Pradeep Poudel
  - Última visita: 15 de Julio de 2019

- Beautiful conclusions from complex datasets
  - Url: https://www.ted.com/speakers/david_mccandless
  - Autor: David McCandless
  - Última visita: 15 de Julio de 2019

- 10 Big Data Trends You Should Know
  - Url: https://www.kdnuggets.com/2018/09/10-big-data-trends.html
  - Autor: Dai Carillo (PureB2B)
  - Última visita: 15 de Julio de 2019

- Caminar con éxito hacia la Industria 4.0: Capítulo 14 – Dispositivos (I) Internet de las cosas (IoT)
  - Url: https://ticnegocios.camaravalencia.com/servicios/tendencias/caminar-con-exito-hacia-la-industria-4-0-capitulo-14-dispositivos-i-internet-de-las-cosas-iot/
  - Autor: Kevin Ashton
  - Última visita: 15 de Julio de 2019

- This Is Why You Need To Learn About Edge Computing
  - Url: https://www.forbes.com/sites/jonmarkman/2018/04/03/this-is-why-you-need-to-learn-about-edge-computing/#171512321a56
  - Autor: Jon Markman
  - Última visita: 15 de Julio de 2019

- ¿Qué es un chatbot?
  - Url: https://www.40defiebre.com/que-es/chatbot
  - Autor: 40deFiebre
  - Última visita: 15 de Julio de 2019

- Advantages and Disadvantages of Python Programming Language
  - Url: https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121
  - Autor: Mindfire Solutions
  - Última visita: 15 de Julio de 2019

- Python Frameworks and Libraries for Web Scraping
  - Url: https://www.scrapehero.com/python-web-scraping-frameworks/
  - Autor: ScrapeHero
  - Última visita: 15 de Julio de 2019

- The database for modern applications
  - Url: https://www.mongodb.com/
  - Autor: MongoDB
  - Última visita: 15 de Julio de 2019

- Robo 3T
  - Url: https://robomongo.org/
  - Autor: 3T Software Labs
  - Última visita: 15 de Julio de 2019

- Apache Nifi
  - Url: https://nifi.apache.org/
  - Autor: The Apache Software Foundation
  - Última visita: 15 de Julio de 2019

- The Elastic Stack
  - Url: https://www.elastic.co/
  - Autor: Elasticsearch
  - Última visita: 15 de Julio de 2019

- ¿Qué es y cómo funciona Elasticsearch?
  - Url: https://www.ochobitshacenunbyte.com/2018/08/28/que-es-y-como-funciona-elasticsearch/
  - Autor: DAVIDOCHOBITS
  - Última visita: 15 de Julio de 2019

- Características y fases del modelo incremental
  - Url: https://www.obs-edu.com/es/blog-project-management/metodologias-agiles/caracteristicas-y-fases-del-modelo-incremental
  - Autor: OBS Business School
  - Última visita: 15 de Julio de 2019

- Qué es el web scraping y para qué sirve
  - Url: https://www.antevenio.com/blog/2019/03/que-es-el-web-scraping-y-para-que-sirve/
  - Autor: Antevenio
  - Última visita: 15 de Julio de 2019

- Beautiful Soup
  - Url: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
  - Autor: Leonard Richardson
  - Última visita: 15 de Julio de 2019

- El crecimiento de las apuestas deportivas en internet
  - Url: https://columnacero.com/deportes/9401/el-crecimiento-de-las-apuestas-deportivas-en-internet/
  - Autor: Columna Cero
  - Última visita: 15 de Julio de 2019