Universidad de Valladolid

Escuela de Ingeniería Informática de Segovia

Grado en Ingeniería Informática de Servicios y Aplicaciones

Desarrollo de un data lake para la gestión de datos sobre navegación marítima

---

Alumno: Miguel de la Fuente Muñoz

Tutores: Anibal Bregón Bregón
Miguel Ángel Martinez Prieto

"If you're going to try, go all the way."

Charles Bukowski

No hay colisión, ni ley ni gravedad
Que te pueda hacer caer aunque tiren a dar
Vetusta Morla

Agradecimientos

Quiero dar las gracias a mis tutores Aníbal y Miguel Ángel por la gran ayuda prestada a lo largo del desarrollo de este proyecto, además de por darme la posibilidad de trabajar en este proyecto con tecnologías totalmente novedosas. Además quería agradecer a hAItta Sàrl, en especial a Domingo Senise de Gracia y Mariano Ferrero, por su colaboración y consejos a lo largo del desarrollo de este proyecto. Me gustaría dar las gracias también a todas las personas que han formado parte de mi vida universitaria, tanto en la escuela como en otras carreras y a mi familia, ya que sin ellos no habría podido llegar hasta aquí.

Resumen

En la actualidad la posibilidad de acceder a grandes conjuntos de datos está cambiando nuestra forma de ver el mundo, afectando a un gran número de aspectos de la vida cotidiana como la economía y ayudando al crecimiento de nuevas tecnologías que nos permitan gestionarlos y utilizarlos. En nuestro proyecto perseguimos el almacenamiento, tratamiento y gestión de datos sobre la navegación marítima mundial.

En el ámbito de nuestro proyecto se gestionarán datos AIS (Automatic Identification System). El sistema AIS es un estándar aprobado por la IMO (International Maritime Organitation) mediante el cual los navíos pueden emitir información sobre su posición, rumbo, velocidad o destino, entre otros datos, de forma que otros navíos, estaciones y puertos los conozcan. El tratamiento de estos datos puede ayudarnos entre otras cosas a evitar colisiones, mejorar rutas de navegación con el fin de ahorrar combustible e incluso a predecir tendencias en el mercado de los productos transportados por el mar.

Para gestionar esta información se ha decidido que a lo largo de este proyecto se diseñará e implementará un Data Lake. Los Data Lakes surgen debido al aumento de la necesidad de almacenar y gestionar grandes volúmenes de datos por parte de las organizaciones. El uso de esos datos podría ayudar a mejorar la distribución de los navíos existentes, con el fin de minimizar gastos de combustible y horas de viaje.

Palabras claves: AIS, LRIT, Data Lake.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En la actualidad la posibilidad de acceder a grandes conjuntos de datos está cambiando la forma de ver el mundo, afectando a un gran número de áreas, como la economía. Sin embargo todos los avances derivados de la posibilidad de contar con tanta información no serían posibles sin los enormes avances existentes en satélites, microprocesadores y tecnologías de análisis de datos. Gracias a estos avances varias empresas, entre las que destacan Orbcomm, exactEarth o Spire Global, iniciaron una carrera entre sí con el fin de desplegar una red de satélites encargados de obtener datos.

Otra de las fuentes principales de obtención de datos es Internet, principalmente a través de las redes sociales. Según un estudio realizado por IBM Marketing Cloud [15] cada día se generan 2'5 quintillones de bytes de datos. Un ejemplo de generación rápida de datos sería Facebook, red social en la cual por cada minuto se crean 510.000 comentarios, 293.000 actualizaciones de estado o se suben 190.000 fotos.

La posibilidad de obtener tal cantidad de información en tan poco tiempo ha puesto en jaque a las tecnologías desarrolladas hasta hace unos años, ya que el volumen de información sobrepasaba la capacidad de almacenamiento disponible. En esta tesitura surgen nuevas tecnologías y arquitecturas cuyo objetivo es dar soporte para el almacenamiento y el tratamiento de grandes volúmenes de información, naciendo así el término Big Data, el cual se encuentra actualmente en auge. Este término suele ser utilizado para referirse a las diferentes arquitecturas desarrolladas con el objetivo de almacenar y dar valor a las ingentes cantidades de datos que se generan en la actualidad.

Estas tecnologías han tenido un gran crecimiento en los últimos años, convirtiéndose los datos en el principal motor del mundo, emergiendo como una de las principales áreas de investigación, junto con la IA. El hecho de ser tecnologías tan novedosas se encuentra ligado a la falta de profesionales cualificados en la materia, convirtiéndose en uno de los puestos más demandados. Según diversos estudios se estima se crearan unos 900.000 puestos de trabajo especializado en Big Data en los próximos 6 años, debido principalmente a las grandes mejoras en producción que pueden ocasionar, para las empresas, la implantación de estas tecnologías. Dentro de los puestos de trabajo de mayor crecimiento asociados a estas tecnologías se encuentran el puesto de Data Scientist o el de analista, de los cuales se espera un aumento del 28 % en su demanda para 2020, según estudios realizados por IBM [6] .

El hecho de poder manejar grandes volúmenes de datos, gracias a las diferentes soluciones Big Data existentes, aunado con el auge de la Inteligencia Artificial, ha influido en el gran desarrollo de áreas de trabajo muy diversas. Entre estas áreas se puede destacar su uso para

la mejora de la seguridad, en la sanidad o a la hora de entender y optimizar los procesos de negocio. En el último área se suele estudiar, analizar y optimizar las rutas de entregas y cadenas de suministros, utilizando para ello sensores GPS y sensores de identificación mediante radiofrecuencia. Estos datos suelen ser muy utilizados, junto a los producidos mediante redes sociales y tendencias de Internet, para realizar análisis basados en predicciones que permitan estudiar tendencias en el mercado, uniendo así esta área con el ámbito de financial trades. Esta unión ha sido muy utilizada en diversos proyectos, como por ejemplo en el caso del transporte aéreo, con proyectos como AIRPORTS [19], proyecto que persigue mejorar la gestión del tráfico aéreo o incluso en el área marítima, con proyectos como BigOceanData [1]. [1]

1.1. Motivación

En la actualidad la principal vía de transporte de mercancías, en especial de commodities, es el mar. Según diversos estudios se ha podido determinar que aproximadamente el 90 % del comercio mundial es transportado a través del mar, para lo cual se utilizan unos 120.000 navíos, los cuales son rastreados gracias a los mensajes que emiten.

Los navíos están obligados, por ley, a emitir mensajes cada cierto tiempo con diversos fines, permitiendo identificarlos y controlar la ruta que siguen. Estos mensajes pueden seguir diversos estándares, entre los que destacan AIS (Automatic Identification System) y LRIT (Long-Range Identification and Tracking), con diferentes campos. La diferencia principal entre estos estándares radica en el uso al que se enfocan, ya que mientras que los datos AIS están más dirigidos a fines comerciales los datos LRIT se enfocan hacia un ámbito gubernamental, ya que este estándar es gestionado por los gobiernos, los cuales se encargan de almacenar los datos relativos a los navíos que llevan su bandera.

Los datos de navegación marítima pueden ser obtenidos de 2 formas distintas, mediante satélites o a través de transmisiones de radio VHF. Estos sistemas se diferencian en el alcance de los datos que pueden obtener, ya que mientras que las transmisiones de radio VHF tienen un alcance limitado, transmiten mensajes entre los navíos y estaciones receptoras situadas en la costa, los datos vía satélite permiten cubrir prácticamente por completo todos los navíos que hay en el mar sin una distancia límite.

Estos estándares surgen con diversos fines, siendo el principal motivo de su nacimiento la mejora en la seguridad marítima, siendo muy utilizado en VTS (Vessel Traffic Service). Mediante el uso estos datos es posible monitorizar la posición y ruta de los navíos y gracias a ellos evitar colisiones entre navíos, siendo también útiles en casos con barcos extraviados, ya que es posible conocer el punto en el que se ha dejado de emitir mensajes,facilitando así la localización de los mismos.

Estos datos se pueden dedicar a una gran cantidad de propósitos, destacando su uso para mejorar la gestión del tráfico marítimo, el cual es un problema en aumento. La facilidad actual para realizar compras en cualquier país del mundo sin moverte de tu casa, a través de empresas como Amazon, cada vez está más interesadas en abarcar todo el proceso de comercio, probablemente provoque un aumento en el número de navíos existente. En el caso de Amazon, según un artículo publicado por The Verge [13], en el último año ha obtenido una licencia para poder operar a través del mar, lo que ha permitido por ejemplo enviar 150 contenedores a China en

[1] Commodity: Término procedente del idioma inglés utilizado para denominar a productos, mercancías o materias primas

los últimos 4 meses, aunque aún sin ser navíos propios de Amazon. El posible aumento en el número de navíos puede ocasionar problemas para la gestión del tráfico marítimo. Sin embargo, gracias a los mensajes emitidos por los navíos, es posible mejorar la gestión del transporte marítimo y las flotas desplegadas a lo largo del mundo, paliando estos problemas. Esta mejora podría permitir ahorros significativos para las empresas, gracias a la optimización de las rutas seguidas por los navíos. La optimización de dichas rutas permitiría entre otras cosas ahorrar combustible, colaborando así a reducir el impacto medioambiental causado por los navíos, así como una gran fuente de gastos para las empresas.

Otro de los principales usos de estos datos radica en la posibilidad de realizar predicciones de tendencias en los mercados dependientes de los productos transportados vía marítima. Este uso está ganando gran importancia ya que es un hecho que el transporte de mercancías a través del mar, especialmente si estas son commodities, es un mercado voluble en el que los precios de las mismas tienden a sufrir grandes fluctuaciones todos los días a lo largo de todo el mundo. Por ello, el análisis de las rutas que siguen los navíos, en tiempo real, puede permitir crear modelos predictivos que permitan a dichas empresas predecir posibles fluctuaciones antes de que tengan lugar. Un ejemplo de la importancia que está adquiriendo una buena gestión de los datos de navegación marítima se puede ver en que una gran compañía de buques petroleros, como es Maersk Tankers, ha realizado una fuerte inversión en Cargometrics, empresa especializada en el análisis y gestión de datos marítimos, según ha publicado Financial Times [28]. El objetivo de esta inversión es conseguir mejorar el despliegue de la flota de Maersk Tankers, algo en lo que puede ayudar en gran medida una empresa puntera en el análisis y gestión de datos marítimos como es Cargometrics.

La cantidad de datos que se generan, tanto por los mensajes emitidos por los navíos como por los generados a través de redes sociales o Internet, especialmente en buques dedicados al transporte de pasajeros, aumenta cada día. El gran aumento en la cantidad de datos obtenidos ha puesto en jaque a las arquitecturas tradicionales de almacenamiento de información, ocasionando graves problemas derivados principalmente de la escalabilidad. Gracias a los grandes avances que se han producido en el área de almacenamiento, asociados a la creación de nuevas arquitecturas Big Data, es posible encontrar solución a estos problemas de escalabilidad. En la actualidad una de las arquitecturas desarrolladas más novedosa y que más se está utilizando ante problemas de escalabilidad es el Data Lake. Esta arquitectura surge como un repositorio capaz de almacenar cualquier tipo de dato, estructurado, semi estructurado y no estructurado, sin ningún tipo de pre procesamiento ni esquema para su posterior análisis. La posibilidad de almacenar en bruto los datos, cargándose directamente desde las fuentes originales, permite gestionar información cuya utilidad puede ser desconocida en la actualidad, pero que en un futuro puede ser de gran interés, siendo esta una de las mayores ventajas de este tipo de arquitecturas.

1.2. Objetivos

El objetivo principal de este proyecto es el diseño e implementación de un Data Lake con el fin de almacenar y explotar los datos de navegación obtenidos, así como la creación de una aplicación de visualización de los mismos. La realización de este objetivo nos permitirá:

1. Entender los datos de navegación marítima y el ecosistema de obtención de estos.

2. Conocer el estado del mercado en aplicaciones que provean datos marítimos.

3. Modelar conceptualmente datos geográficos y datos relacionados en el ámbito de las commodities.

1.3. Organización del Documento

En esta sección se describe la estructura del documento realizado con el fin de servir como apoyo al lector. Se dividirá en los siguientes capítulos:

- **Capítulo 1. Introducción.** En este capítulo se realizará una presentación del proyecto realizado, así como de la motivación para llevarlo a cabo y los objetivos que se persiguen con su realización.
- **Capítulo 2. Entorno Marítimo.** En este capítulo se persigue describir el ecosistema marítimo, así como los diferentes protocolos de comunicación existentes en dicho ecosistema.
- **Capítulo 3. Estado del Arte.** En esta sección se detallarán diferentes aplicaciones que haces uso de datos sobre el tráfico marítimo, centrándonos en gran parte en las diferentes licencias que ofertan para obtener dichos datos. Este capítulo es de vital importancia ya que en él se decide la fuente de información a utilizar a lo largo de todo el proyecto.
- **Capítulo 4. Big Data.** En este capítulo se presentará el concepto Big Data y se explicará en detalle el concepto Data Lake y su arquitectura.
- **Capítulo 5. Modelo de Datos.** En esta sección se describirán las diferentes fuentes de información utilizadas en este proyecto, así como los atributos de estas que se utilizarán. Además de esto se desarrollará el Modelo Conceptual del proyecto a realizar y se detallarán las diferentes transformaciones realizadas sobre los datos desde el *Raw Data* hasta su uso por la aplicación web desarrollada.
- **Capítulo 6. Análisis.** En este proyecto se trata de realizar un breve análisis, siguiendo una metodología ágil, para el desarrollo de la aplicación web. Además de esto se realizará una pequeña sección de gestión de los riesgos que pueden afectar al desarrollo de este proyecto.
- **Capítulo 7. Gestión del Proyecto.** En este proyecto se describe la metodología utilizada en este proyecto, así como la planificación y el presupuesto del mismo.
- **Capítulo 8. Diseño.** En este capítulo se mostrarán las arquitecturas física y lógica del sistema desarrollado, así como los diseños de interfaz de la aplicación web a desarrollar.
- **Capítulo 9. Implementación.** En este capítulo se describirá en detalle cómo se ha desarrollado cada una de las herramientas que componen el proyecto, así como las tecnologías y lenguajes utilizados para el desarrollo del mismo.
- **Capítulo 10. Métricas.** En este capítulo se persigue describir una serie de métricas sobre el conjunto de datos, así como analizar los resultados de estas.
- **Capítulo 11. Pruebas.** En este capítulo se describen las pruebas que se han realizado para comprobar el correcto funcionamiento de la aplicación web desarrollada.

- **Capítulo 12. Manuales.** En este capítulo se podrá ver el manual de usuario de la aplicación web desarrollada.
- **Capítulo 13. Conclusiones y Trabajo Futuro.** En este capítulo se hace una pequeña valoración del proyecto realizado, así como describir las posibles mejoras a realizar en un futuro.

Capítulo 2

Entorno Marítimo

A principios de siglo se comenzaron a desarrollar diversos estándares, aplicados a la navegación marítima, que permiten identificar y monitorizar el tráfico marítimo a lo largo del mundo con el objetivo de mejorar su seguridad, tratando de evitar colisiones o incluso facilitando la búsqueda de navíos extraviados. Estos estándares permiten obtener gran cantidad de información tanto del viaje a realizar como los datos propios de cada navío, aunque estos datos varían en función del estándar que sigan. La obtención de estos datos se ha convertido en un gran negocio, participando grandes empresas para la obtención de datos, como ExactEarth en los satélites, empresas de gestión de la información obtenida, como Cargometrics.

El desarrollo de estos estándares se apoya en la creación de una arquitectura que permita a los navíos emitir mensajes con información sobre sí mismos así como sistemas de recepción y almacenamiento de estos. En la figura 2.1 se puede ver la arquitectura creada para la obtención y el almacenamiento de este tipo de datos.

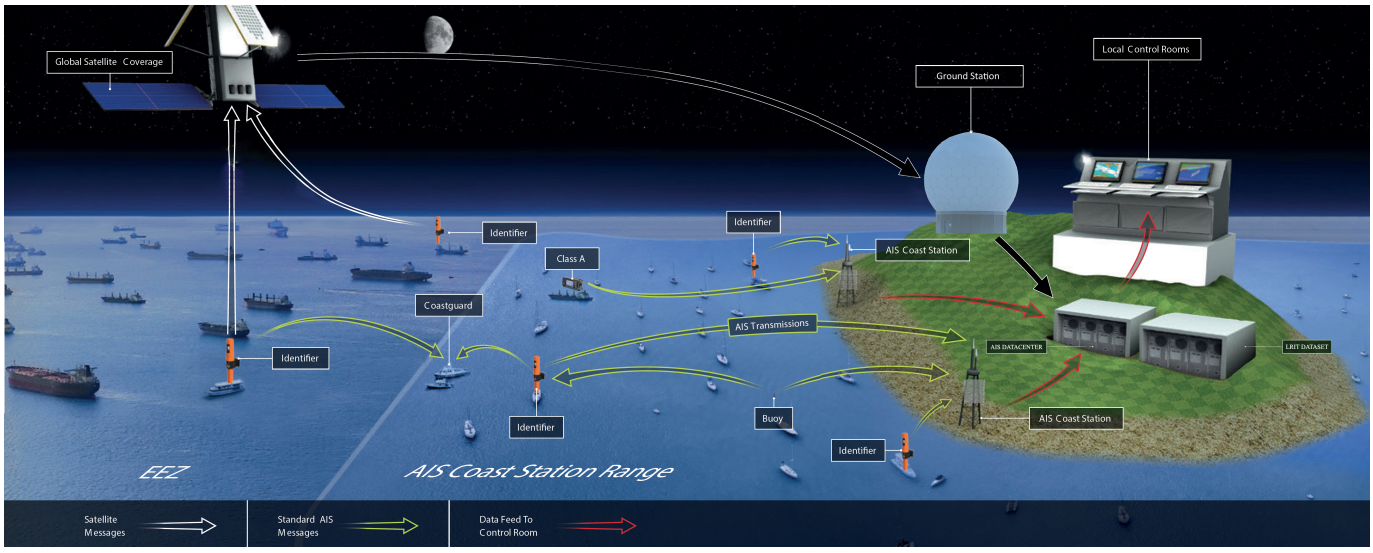


Figura 2.1: Arquitectura de los estándares de obtención de datos

2.1. Ecosistema de datos

La arquitectura vista anteriormente permite crear un ecosistema de datos a partir del cual poder obtener y almacenar la información de navegación marítima deseada. Este sistema consta de 3 partes:

Emisión. Los mensajes pueden ser enviados por:

- *Navíos*: Los navíos son los encargados de emitir diversos tipos de mensajes compuestos por información sobre la posición del navío, información relativa al viaje o datos estáticos relativos al navío, como por ejemplo su identificador. La forma de emitir mensajes varía en función del estándar, por ejemplo con el estándar AIS los mensajes se emiten automáticamente, mientras que con el estándar LRIT la emisión de mensajes requiere de participación directa de la tripulación.
- *AtoN Systems*: Sistemas electrónicos integrados en elementos externos a los navíos, como por ejemplo en balizas, que surgen como ayuda a la navegación. Este tipo de sistemas permiten identificar balizas, faros y navíos, así como emitir alarmas de balizas o datos meteorológicos y oceanográficos.
- *SAR (Search and Rescue) Transmitter*: Tecnología utilizada principalmente en dispositivos de localización de supervivientes a accidentes marítimos, suelen acoplarse a botes y balsas salvavidas.

Recepción. Los mensajes pueden ser recibidos por:

- *Navíos*: Los navíos pueden recibir mensajes emitidos por otros con el fin de evitar colisiones.
- *Estaciones costeras*: Actúan como receptores de señales de corto alcance, normalmente mensajes emitidos a través de radio VHF. Las estaciones se sincronizan entre sí de manera automática, evitando así que se den problemas de superposición a la hora de recibir los mensajes. Para poder sincronizarse, cuando cada estación cambia de slot temporal notifica tanto la nueva ubicación, frecuencia, como el slot en el que operará. Esto facilita la comunicación entre las estaciones y los navíos.

  Algunas de estas estaciones actúan como repetidores, con el fin de mejorar la cobertura, de los mensajes emitidos vía VHF, ampliando el rango de envío de los navíos

  Las estaciones son las encargadas de enviar los datos a los diversos Data Center donde se almacenan y analizan. En caso de que el estándar de datos recogidos sea LRIT estas estaciones podrán enviar datos a las autoridades nacionales para que ellas se encarguen de su almacenamiento.
- *Satélites*: Existen distintos tipos de satélites encargados de captar información de las zonas donde los receptores costeros no tienen señal. La utilización de diferentes satélites para la obtención de información marítima se encuentra ligada a la obtención de los diferentes tipos de datos o estándares existentes. Dentro de estos satélites destacan los satélites AIS o LRIT, que ofrecen una cobertura casi total del mundo.

Una vez captados, los datos son enviados a estaciones en tierra que se encargan del envío de los datos al Data Center.

- *AtoN Systems*: Los sistemas electrónicos AtoN, al igual que emitir mensajes, son capaces de recibir información.

- *Synthetic Aperture Radar*: Sistema radar que gracias a los algoritmos de procesado utilizados, es capaz de combinar la información obtenida mediante varios barridos, recreando un único barrido. Es muy utilizado en sistemas de cartografía, así como en aplicaciones de detección, por ejemplo para recibir emisiones de boyas a través del SAR transmitter.

Almacenamiento. Los datos captados por los sistemas receptores, anteriormente mencionados, son almacenados en Data Center. La funcionalidad de cada Data Center varía en función del estándar de datos captado, por ejemplo, si el estándar de datos captados es LRIT los propios gobiernos son los encargados del almacenamiento de los datos de los navíos que lleven su bandera. Un ejemplo de esto es EU LRIT CDC, Data Center gestionado por EMSA (European Maritime Safety Agency), en el cual se almacenan los datos marítimos relativos a los navíos de los países pertenecientes a la Unión Europea. En la tabla 2.1 se pueden ver algunos de los Data Center más conocidos.

Nombre	Tipo de dato	Tipo de fuente
AISHub	AIS	Comercial
MarineTraffic	AIS	Comercial
FleetMon	AIS	Comercial
EU LRIT CDC	LRIT	Gubernamental
United States National LRIT Data Centre	LRIT	Gubernamental

Tabla 2.1: Ejemplos de Data Center

Cabe destacar que los Data Center no se encargan únicamente del almacenamiento de los datos. Es muy común que los propios Data Center desarrollen diversas aplicaciones, asociadas al tratamiento de los datos obtenidos, así como que actúen como intermediarios, compartiendo información con otros Data Center.

Un aspecto a destacar de estos sistemas es la codificación de las comunicaciones entre los diversos sistemas marítimos que componen el ecosistema. Los mensajes emitidos son codificados utilizando NMEA-0183 [22], estándar creado para la comunicación entre los diversos agentes marítimos. Estos mensajes son decodificados una vez se almacenan en el Data Center, existen diversas aplicaciones que permiten gestionar este tipo de mensajes, por ejemplo freenmea [11]. En las siguientes secciones se explicarán los estándares de mayor relevancia.

2.2. AIS (Automatic Identification System)

Estándar aprobado por la IMO (Organización Marítima Internacional) en el año 2002, creando un sistema cuyo objetivo principal es permitir que los navíos comuniquen su posición automáticamente, entre otros datos, de forma que otros navíos y estaciones puedan conocerla.

Este estándar es muy utilizado a lo largo del mundo para prevenir colisiones, mejorar la seguridad marítima e incluso la eficiencia de la navegación. Cabe destacar que este estándar se encuentra orientado a fines comerciales.

En sus inicios este sistema utilizaba únicamente señales de radio VHF, hasta que en el año 2006, gracias al despliegue de satélites realizado por diversas empresas, como Orbcomm, se comenzó a recibir señales vía satélite de los navíos, aumentando el alcance de emisión de señales y permitiendo complementar el uso de radar marítimo. Este aumento en el alcance permite el seguimiento de navíos en mar abierto, algo que no se podía conseguir únicamente con las señales de radio VHF.

El uso de este sistema de identificación es obligatorio para todos los navíos de más de 300 toneladas, los navíos comerciales y de recreo de más de 20 metros, así como todos los navíos CEMT clase 1 o superior.

Los navíos son capaces de enviar 27 tipos distintos de mensajes AIS utilizando el protocolo NMEA-0183, especificación a través de la cual pueden comunicarse instrumentos marítimos y receptores de GPS. La información contenida en estos mensajes puede dividirse en 3 tipos, estática, dinámica o sobre el viaje. Sin embargo, no es posible enviar toda la información relativa a un navío en un único mensaje, por ejemplo, la información relativa a la posición suele enviarse cada 2 segundos, llegando a enviarse cada 10 segundos en función del navío.

El envío de este tipo de datos suele realizarse a través de dos frecuencias de radio móviles, complementadas por otros dos canales, utilizados si surgen problemas de interferencias. Para conseguir que los navíos compartan el canal se utiliza TDMA (Time Division Multiple Access), técnica muy utilizada en la transmisión de señales digitales y cuya función es la de dividir la señal en slots de tiempo en los que los navíos envían su información. El tratamiento de los slots varía en función del sistema AIS que utilice cada navío, por ejemplo los sistemas AIS de clase A utilizan SOTDMA (Self-Organized Time Division Multiple Access) y en los de clase B se utiliza CSTDMA (Carrier Sense Time Division Multiple Access). Sin embargo es común a todas las clases de sistemas AIS el hecho de que por cada slot sólo puede enviarse un único mensaje AIS. Este hecho provoca que se deba realizar una reorganización en la emisión de datos en caso de que se envíen más mensajes de los disponibles en tiempo, la solución normal consiste en transmitir los mensajes de los navíos más cercanos a la estación base, mientras que los de los navíos más alejados se envían a otra estación base.

A pesar de los grandes avances en los que se ha visto envuelto este sistema sigue encontrándose con ciertas limitaciones. Algunas de las limitaciones más destacadas son:

- La exactitud en los datos se encuentra ligada a la calidad de la emisión realizada.
- No todas las naves se encuentran equipadas con este sistema.
- Cabe la posibilidad de que los sistemas AIS de un buque sean apagados, dando lugar a problemas a la hora de entender las rutas seguidas por los navíos.
- Este sistema puede enviar información errónea.

Pese a estas limitaciones el sistema AIS sigue siendo el más utilizado para la obtención de información marítima, siendo utilizado por gran cantidad de empresas, entre las que destacan MarineTraffic [17] y Fleetmon [10].

2.3. LRIT (Long-Range Identification and Tracking)

Sistema aprobado por la IMO en el año 2006, creando un estándar cuyo objetivo principal es permitir a los gobiernos que contraten estos servicios obtener la información asociada a los diferentes navíos que navegan por sus costas, con la antelación suficiente, permitiendo evaluar el riesgo asociado a estos y actuar en consecuencia.

Por normativa los navíos están obligados a emitir un mínimo de 4 veces diarias, pudiendo emitir como máximo cada 15 minutos. A diferencia de los sistemas AIS, en estos sistemas los mensajes no se emiten de manera automática, es decir, es el propio navío el que indica que desea enviar un mensaje. Este sistema es obligatorio para los siguientes tipos de navíos:

- Navíos de transporte de pasajeros
- Cargueros, incluidos los de alta velocidad, con un GRT (Gross Registered Tonnage) superior a 300
- Unidades móviles de perforación
- Naves de alta velocidad

Los datos obtenidos mediante este sistema son almacenados los diversos Data Center gubernamentales existentes. En estos se almacena únicamente la información de los navíos asociados al Data Center, por ejemplo en el caso de la Unión Europea se almacenan los datos de todos los navíos pertenecientes a esta en el EU LRIT CDC Data Center. Esto permite a los gobiernos realizar seguimientos sobre los navíos abanderados a lo largo de todo el mundo.

Este sistema surge como un complemento al sistema AIS, ya que permite obtener información en puntos que no alcanzan los sistemas AIS. La obtención de este tipo de datos es similar a la de los datos AIS, ya que se suelen obtener vía satélite. Los datos enviados por los navíos son recogidos por satélites proporcionados por CSP's (Communications Service Providers), empresas encargadas de la infraestructura y los servicios de comunicación de los elementos que componen el sistema LRIT, conectados a través de protocolos que garanticen la seguridad de esta información.

Una vez obtenida la información, ésta es enviada al ASP (Application Service Provider). Estos se encargan de completar la información asociada al navío, dotándolo de identidad al añadir los campos IMO (identificador asociado a la IMO) y MMSI (identificador único del navío), así como añadiendo la fecha y hora de recepción del mensaje. Gracias a la inserción de estos nuevos campos se genera un nuevo mensaje, el cual es enviado al Data Center.

Los Data Center son los encargados de almacenar y distribuir los datos obtenidos, de acuerdo con el Plan de Distribución de Datos. Existe la posibilidad de intercambiar información LRIT entre los diversos Data Center existentes gracias al sistema LRIT IDE [7] (International LRIT Data Exchange). Este sistema actúa como enlace entre los diversos Data Center gubernamentales, permitiendo el envío de información entre estos, siempre teniendo en cuenta para ello el Plan de Distribución de Datos. Este sistema únicamente almacena la información referente a las cabeceras de todas las transacciones realizadas, con fines estadísticos y de auditoría. La información relativa a los navíos siempre será por tanto almacenada en un Data Center. En la figura 2.2 se puede observar el esquema seguido para la obtención y gestión de este tipo de datos.

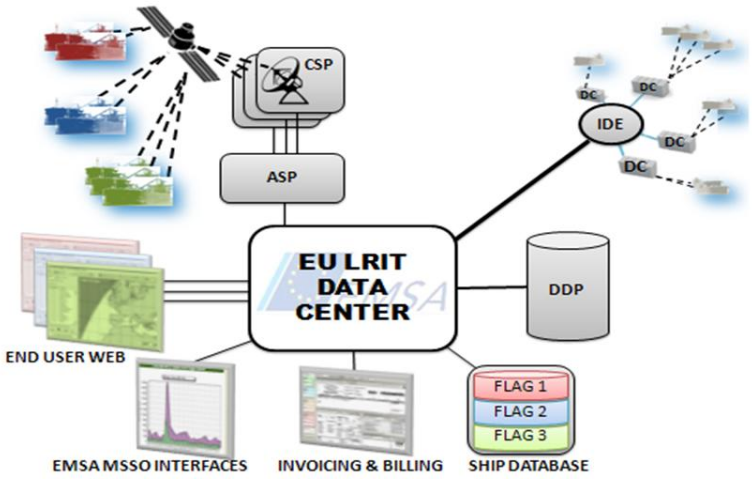


Figura 2.2: Ecosistema LRIT

La supervisión de este sistema, a nivel mundial, es realizada por la IMSO (International Mobile Satellite Organization), la cual fue nombrada como coordinadora LRIT por el MSC. Esta organización se encarga de auditar, al menos una vez al año, al IDE, con el fin de revisar el cumplimiento de las condiciones de seguridad y rendimiento del sistema.

Es posible acceder a diversos listados sobre los gobiernos que utilizan este tipo de información gracias a EMSA [8] (European Maritime Safety Agency), la cual nos permite ver desde su web los países asociados al LRIT, así como los que se encuentran conectados a través de IDE.

2.3.1. EMSA

Agencia europea de seguridad marítima encargada de la gestión y explotación del Data Center EU LRIT CDC. Esta agencia es la encargada de gestionar toda la información relevante a la navegación de los navíos con bandera europea. Sin embargo esta organización no se dedica únicamente a la gestión de este Data Center, sino que, gracias a la información almacenada en el mismo, ha desarrollado sistemas como SafeSeaNet, el cual persigue la monitorización de barcos en tiempo real.

SafeSeaNet aúna información obtenida a través de diversos protocolos, como AIS o LRIT, con el fin de monitorizar en tiempo real la información marítima mundial. La centralización de la información en un único sistema beneficia a diversas áreas de trabajo, como la gestión de emergencias, el control portuario o la preparación y respuesta ante la contaminación.

En la actualidad EMSA está desarrollando un framework llamado IMDatE [16] , con el que se persigue almacenar y combinar la información obtenida, tanto por los protocolos AIS y LRIT, como por otras aplicaciones como SafeSeaNet o CSN. La posibilidad de gestionar todos estos datos radica en la capacidad de este sistema para soportar transacciones entre las diversas aplicaciones existentes. El principal objetivo de este sistema es apoyar y mejorar las aplicaciones EMSA existentes, añadiendo nuevas funcionalidades que permitan a los usuarios

una mejor interacción con la misma, destacando el seguimiento automático de navíos o la posibilidad de el aumento en las opciones de visualización. Este sistema está compuesto por 4 características principales:

- **Integrated Ship Profile Service**: Servicio encargado de proporcionar una visión combinada de la información relacionada a un navío o a una flota. Para ello este servicio se basa en las diferentes fuentes de información utilizadas por IMDatE.
- **Area Centric Service**: Este servicio permite la visualización completa de un área seleccionada.
- **Maritime Surveillance Service**: Servicio cuya función consiste en facilitar el análisis de la información relativa al tráfico de buques, ayudando así a la vigilancia marítima.
- **EU Common Maritime Space Monitoring Service**: Servicio de apoyo a las aplicaciones CMS cuya función consiste en supervisar los navíos que participan en el comercio costero en la Unión Europea y los que llevan a cabo servicios regulares entre los puertos de la misma.

Como previamente se ha comentado, existen diversas organizaciones de carácter similar a EMSA, como PoleStar. Cada cual se encarga de gestionar algún Data Center. Gracias a LRIT IDE es posible la comunicación entre los diversos organismos gubernamentales. En la propia web de EMSA es posible acceder al listado de todos los países que han contratado el sistema LRIT, así como todos los Data Center conectados a través de LRIT IDE.

Capítulo 3

Estado del Arte

En la mayoría de las ocasiones en este capítulo se describen aplicaciones o sistemas similares a los que se pretenden desarrollar en el proyecto. Sin embargo en esta ocasión, dado que se trata de un proyecto de investigación, así como al entrar en un ámbito poco tratado, ha hecho imposible la creación de un capítulo de estado del arte al uso.

En el inicio del proyecto, en la fase de adaptación al dominio del problema que se pretende solucionar, se analizaron una gran cantidad de aplicaciones que permitían bajo una diversa variedad de licencias la obtención de datos sobre el tráfico marítimo mundial. Por ello se ha decidido que en esta sección se trate el proceso de comparativa y análisis realizado entre todas las aplicaciones significativas que utilicen o provean datos de información marítima.

En este capítulo se detallarán las diversas aplicaciones encontradas que ofrecen y utilizan datos de navegación marítima, tanto gratuitas como de pago, y las diferencias dentro de cada aplicación entre la versión gratuita o las distintas versiones de pago que ofertan.

Los datos que se desean obtener pueden ser de dos tipos, como se ha comentado en el capítulo anterior, datos AIS (Automatic Identification System) y datos LRIT (Long-Range Identification and Tracking), siendo de vital importancia la información disponible sobre los barcos y su mercancía, así como la información histórica que se ha almacenado sobre cada barco. En este capítulo únicamente se tratarán las aplicaciones que oferten datos obtenidos a través del protocolo AIS, al haber sido detalladas las aplicaciones más interesantes asociadas al protocolo LRIT en el capítulo anterior.

3.1. Obtención de Datos AIS

El sistema AIS fue aprobado por la IMO (Organización Marítima Internacional) en el año 2002, creando un estándar cuyo objetivo principal es permitir que los navíos comuniquen su posición, entre otros datos, de forma que otros navíos y estaciones puedan conocerla y así evitar colisiones.

A continuación se pueden ver una serie de aplicaciones que utilizan este tipo de datos, centrándonos en los datos que nos oferta cada plataforma, así como las diferentes licencias que se pueden adquirir.

myShipTracking. MyShipTracking es una aplicación web encargada de la visualización de datos tanto de barcos como de puertos, pudiendo acceder a un histórico de datos y a la posición actual de cada barco y su cargamento. Los datos disponibles en esta plataforma únicamente

pueden ser descargados y gestionarlos si se dispone de receptores AIS, muchas aplicaciones ofrecen licencias gratuitas y datos siempre que se les pueda ofrecer datos.

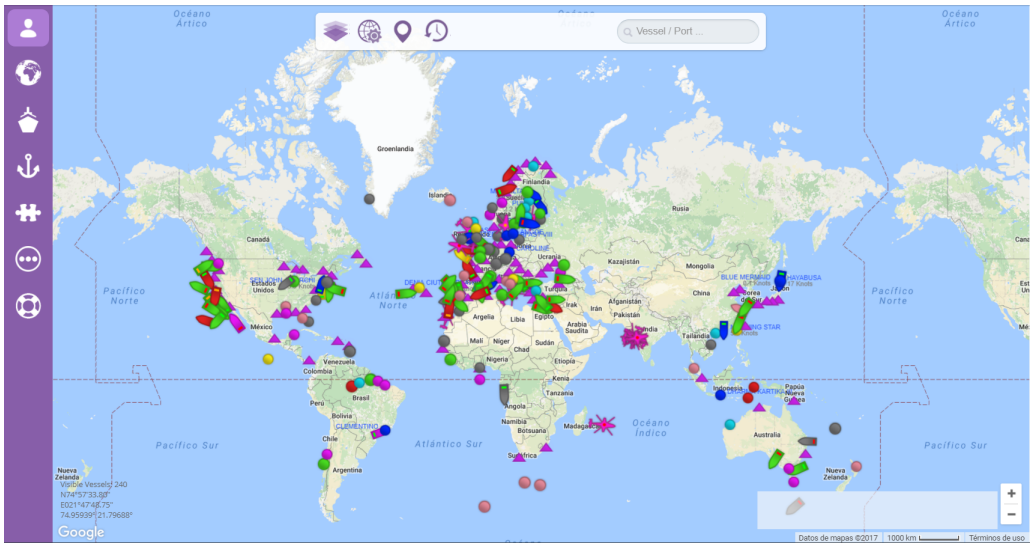


Figura 3.1: Inicio MyShipTracking

Esta aplicación destaca por la gran cantidad de datos que maneja y por su visualización, la cual nos permite la geolocalización de los diversos barcos y puertos en tiempo real. Una característica muy importante de la visualización es su capacidad de seleccionar un barco o un puerto y mediante un Pop-Up poder ver la información más destacada del elemento seleccionado. Como se muestra en la figura 3.2.

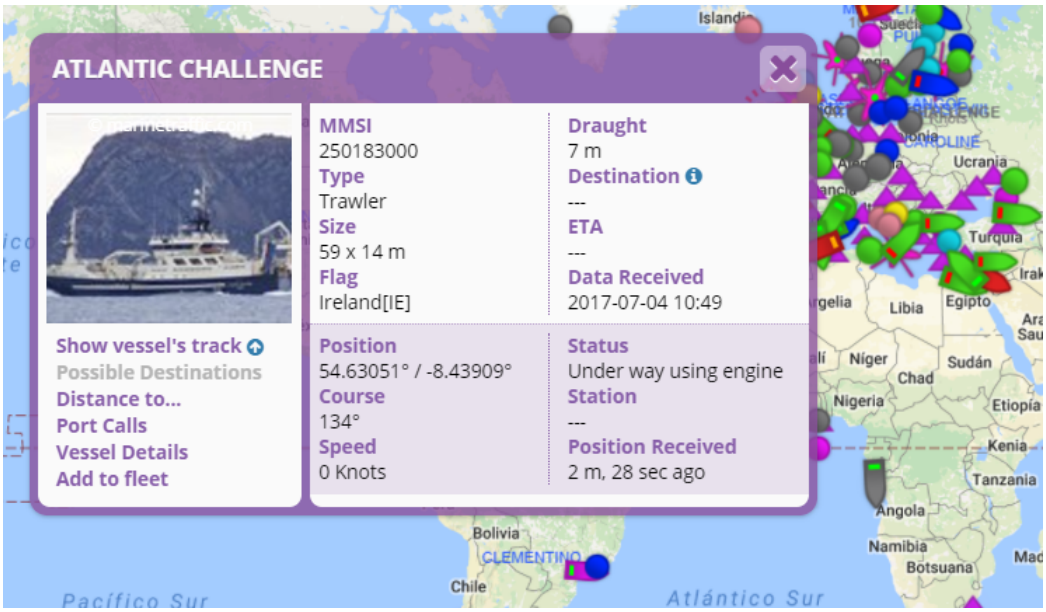


Figura 3.2: Ejemplo de Pop-UP

Esta herramienta también permite acceder a un listado de puertos y barcos así como realizar filtrados con el fin de acceder a la información detallada de los elementos seleccionados. Dentro de la información detallada de cada navío destaca el tipo de navío y tanto el GRT (Gross

Tonnage) como el DWT (Deadweight Tonnage), sin embargo no es posible acceder a ningún dato sobre la carga del mismo. Además también ofrece gran cantidad de datos tanto de localización actual como un histórico de posiciones del navío.

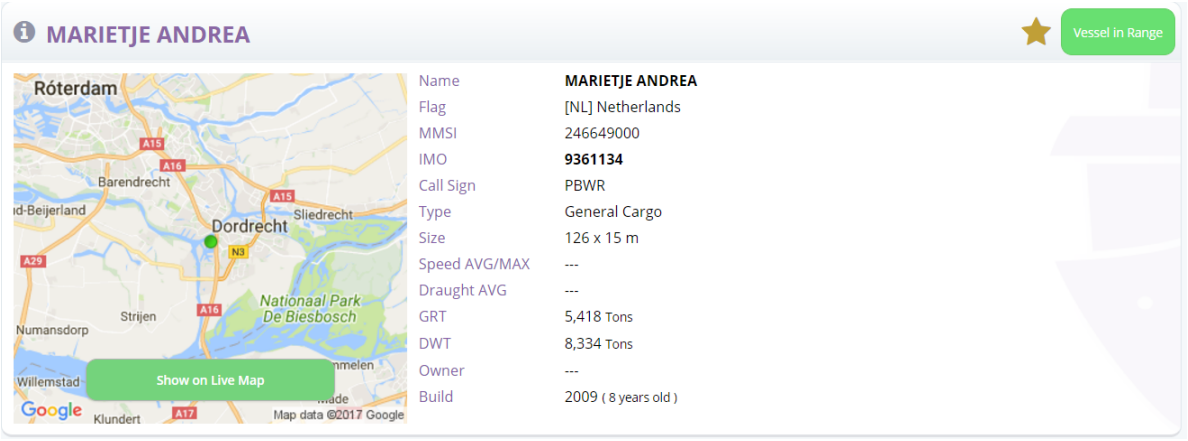


Figura 3.3: Información detallada de un navío

Esta aplicación es totalmente gratuita, existiendo la posibilidad de crear una cuenta gratuita en la plataforma, la cual habilita la opción de crear receptores AIS, lo cual es susceptible de facilitar la obtención de los datos deseados así como el envío de datos. Esta parte privada permite también la creación de nuevos puertos, barcos y flotas. Es posible acceder a esta a través del siguiente enlace: http://www.myshiptracking.com/

ShippingExplorer. Esta aplicación consta de dos partes, una pública que permite visualizar datos sobre naves y puertos, tanto en un mapa (aunque sin haberse registrado previamente la visualización es de peor calidad, al reducir la cantidad de información que se muestra) como por listados. En estos listados permite realizar filtrados con los que poder encontrar los datos deseados y ver la información detallada de las naves o puertos seleccionados. Dentro de la información ofertada, como en la anterior aplicación, destacan el GRT, el DWT y el tipo de barco, además de informar si lleva carga especial.

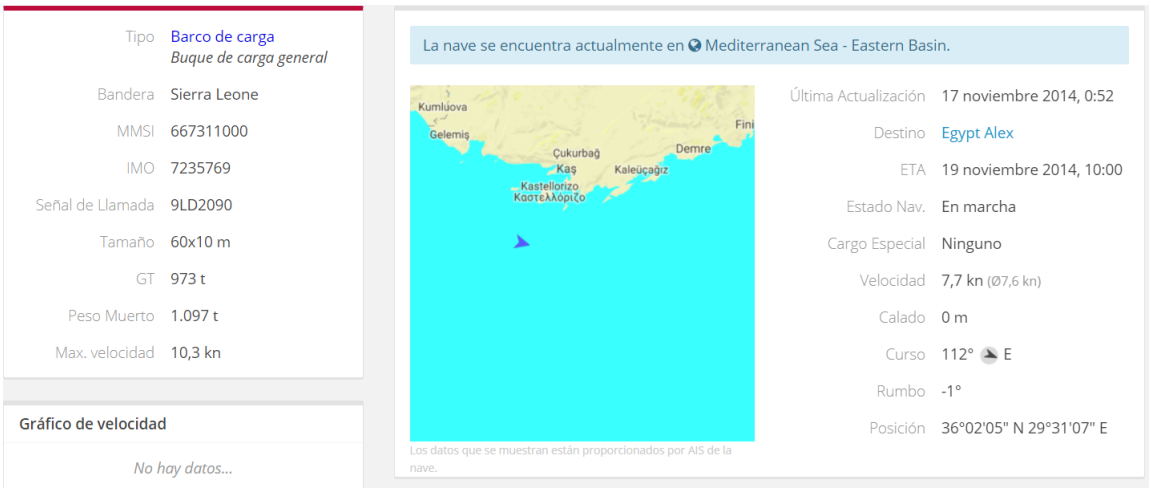


Figura 3.4: Datos generales en ShippingExplorer

En algunos casos es posible acceder a una mayor información, tanto de la carga como del propietario, el gerente, el constructor o las dimensiones.

Figura 3.5: Información detallada

Una vez llegados a este punto se podrán añadir fotos y comentarios sobre el elemento seleccionado, algo que solo es posible hacer si previamente se ha accedido al sistema como usuario registrado.

Un aspecto a destacar de la aplicación en su parte pública es la posibilidad de acceder a un foro en el que tratar temas de interés de los diversos usuarios de la plataforma. Sin haber accedido previamente a la plataforma como un usuario registrado sólo se podrá leer los mensajes existentes en el foro, por lo que en el caso de desear crear nuevos hilos se necesitará acceder como un usuario registrado.

La característica principal de esta aplicación es su navegador, el cual permite realizar un seguimiento en tiempo real de las muchas naves y puertos de los que se dispone de información. Esta aplicación permite la visualización de datos mediante gran cantidad de filtrados y la geolocalización de navíos y puertos. Destaca por su capacidad para conseguir un gran número de estadísticas sobre navíos y puertos y por la posibilidad de crear alertas, las cuales se pueden recibir vía e-mail o por SMS. La característica de la aplicación que más interesante puede resultar para proyecto a realizar, es la del histórico de Recorrido y Puerto, opción que nos permite descargar la información de dicho apartado.

Sin embargo este navegador no es gratuito y actualmente no se puede acceder a ninguna demo ya que no hay ninguna disponible. Esta plataforma ofrece varios tipos de licencia con ciertas diferencias entre sí, como se puede observar en la tabla 3.1.

Servicio	Cuenta Web Gratuita	Licencia Básica	Licencia Avanzada
Nº Usuarios	1	1	1-10
Mapa y posiciones	Sí	Sí	Sí
SW de Cliente (Navegador)	No	Sí	Sí
Alertas	No	Sí (7€100 SMS)	Sí (7€100 SMS)
Histórico de recorrido	No	Sí	Sí
Análisis de tráfico	No	Sí (15€/mes)	Sí (15€/mes)
Interacción en el foro	Sí	Sí	Sí
Soporte Básico	Sí	Sí	Sí
Soporte Premium	No	Sí	Sí
Precio	Gratis	42€/mes	Desde 80€/mes en función del número de usuarios

Tabla 3.1: Tipos de Licencias de ShippingExplorer

Para ver más información sobre el precio de la licencia avanzada, detallando el precio con distinto número de usuarios, se puede visitar la página web de la plataforma, más concretamente accediendo al apartado de precios mediante el siguiente enlace: http://www.shippingexplorer.net/es/pricing

Además de las anteriores opciones existe la posibilidad de conseguir acceso gratuito a la versión completa del navegador ShippingExplorer, para ello debemos enviar datos AIS, ya sea vía receptor AIS o mediante los programas Ship Plotter o AISMon.

ShipFinder. Aplicación web que permite la visualización y búsqueda de navíos y puertos. Esta aplicación consta de una parte pública y de otra parte privada, teniendo la parte pública una funcionalidad muy reducida. En la parte pública lo único que se podrá hacer es visualizar un mapa con los navíos y puertos de los que se disponen datos, así como la posibilidad de buscar navíos o puertos con el fin de acceder a la información detallada de estos.

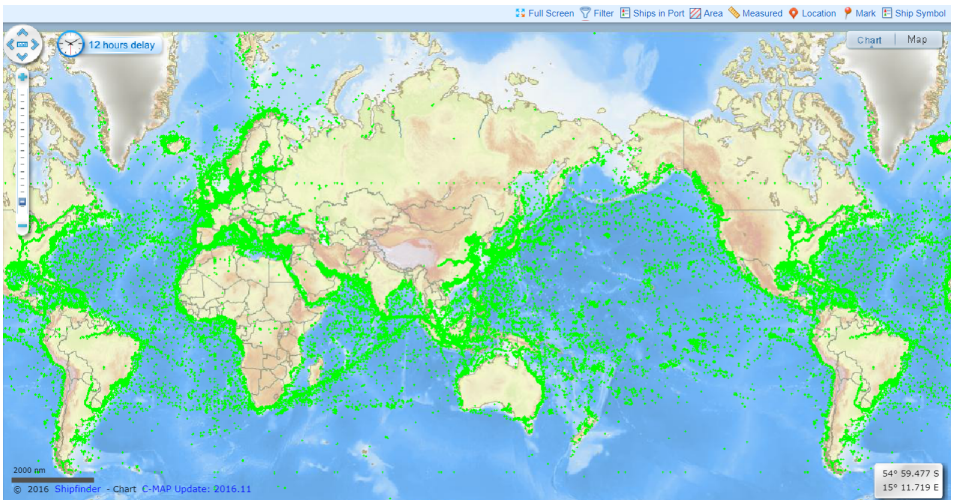


Figura 3.6: Mapa ShipFinder

La información disponible en esta aplicación es inferior a las anteriores, como demuestra la figura 3.7. Además la forma de localizar barcos es mucho más compleja al no permitir filtrados, sólo en la parte pública, ni tener acceso a un listado de navíos.

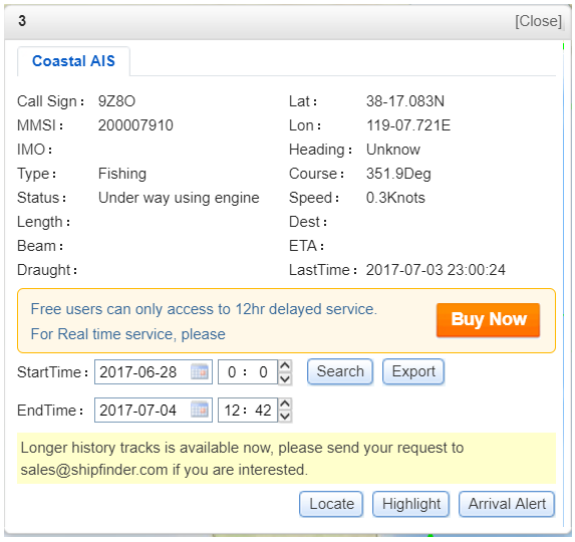


Figura 3.7: Información detallada de un navío

Esta aplicación, al igual que la anterior, permite la gestión de alertas que informen sobre el estado de las embarcaciones deseadas en un cierto momento o de las embarcaciones que acceden a un área previamente indicado. Sin embargo esta opción no está habilitada para los usuarios sin registrar ni para los usuarios con una cuenta gratuita.

Como las anteriores aplicaciones ShipFinder utiliza información AIS, sin embargo esta aplicación divide la información AIS recibida en función del tipo de licencia deseada, ya que se pueden obtener diversos tipos de licencias sobre datos AIS obtenidos mediante receptores situados a nivel de costa o sumar a estos los datos obtenidos vía satélite. Sin embargo nos centraremos en las licencias sobre los datos que se obtienen mediante receptores situados en las costas. Se pueden ver las diferentes licencias costeras en la tabla 3.2, ya que se desconoce el coste asociado a las licencias de obtención de datos vía satélite. Se puede ver lo que ofrecen estas licencias en la web de la plataforma, accediendo a través de la siguiente url: http://www.shipfinder.com/Home/Price

Servicio	Licencia Gratuita	Map Standard	Map Standard Plus	Chart Standard	Chart Standard Plus
Nº Usuarios	1	1	3	1	3
Open Street Map	Sí	Sí	Sí	Sí	Sí
C-MAP Chart	Sí	No	No	Sí	Sí
AIS Data	12 horas de retraso	Tiempo real	Tiempo real	Tiempo real	Tiempo real
Ship Highlighting (20 ships)	No	Sí	Sí	Sí	Sí
Ship tracks (30 days)	No	Sí	Sí	Sí	Sí
Alertas	No	Sí	Sí	Sí	Sí
Barcos en puerto	No	Sí	Sí	Sí	Sí
Precio	Gratis	42€/mes	60€/mes	52€/mes	69€/mes

Tabla 3.2: Tipos de Licencias de ShipFinder Coastal AIS

Un posible problema de esta herramienta se da a la hora de tratar de descargar datos, ya que los datos están almacenados en una base de datos perteneciente a IHS Fairplay, el cual es un servicio únicamente disponible en China, por lo que no es posible acceder a esos datos ni descargarlos.

VesselFinder. Aplicación web encargada de la visualización de datos AIS marítimos mediante un mapa sobre el que se podrán buscar navíos.

Esta aplicación además permite acceder a listados tanto de navíos como de puertos y a la información detallada de cualquier elemento seleccionado, destacando el tipo de navío, el GRT, el DWT o los últimos puertos por los que ha pasado y su localización actual. Sin embargo parte de la información de estos elementos sólo se encuentra disponible en caso de tener acceso a una cuenta Premium, como se puede observar en la figura 3.8.

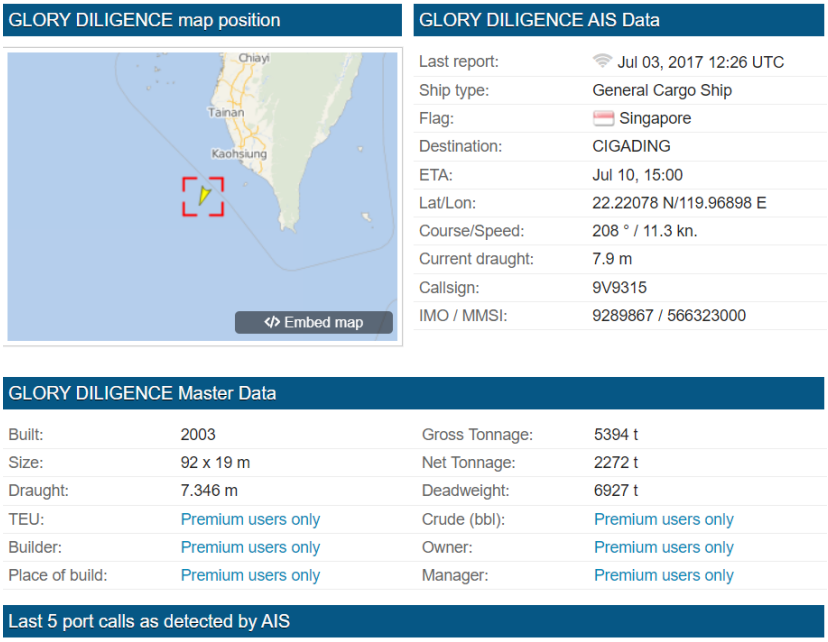


Figura 3.8: Información detallada

Es posible crear una cuenta en dicha plataforma, la única diferencia entre un usuario sin registrar y un usuario registrado sin cuenta Premium reside en que los usuarios registrados pueden crear flotas (Max 50) y añadir lugares propios sobre los que recibir alertas. Existe también la posibilidad de conseguir una cuenta Premium en la que podremos crear flotas de 500 navíos y poder acceder a la información completa de los navíos y puertos.

Las opciones más interesantes de esta aplicación son “Historical AIS Data” y “Real-Time AIS Data”, los cuales permiten solicitar información sobre navíos. En ambas opciones los datos solicitados son de pago, el coste de los datos es enviado al correo una vez que los trabajadores de VesselFinder evalúan los datos a enviar. Además la información que se puede solicitar es limitada ya que para solicitar información es necesario indicar el navío o conjunto de navíos de los que se desea obtener información.

Los datos obtenidos mediante la opción “Historical AIS Data” pueden ser usados para realizar diversos tipos de análisis, sin embargo el que más nos puede interesa es “Vessel Movements Reports”, mediante el que podemos estudiar las trayectorias seguidas por los navíos y el estado de estos en cada momento. Sin embargo como previamente se ha comentado no es posible conocer el coste de los datos hasta que no les hayamos solicitado, ya que será enviado al correo que previamente hayamos indicado en la solicitud. Podemos acceder a esta plataforma a través del siguiente enlace: https://www.vesselfinder.com/

FleetMon. Fleetmon es una de las aplicaciones web más utilizada para el seguimiento de navíos. Permite acceder a una base de datos tanto de navíos como de puertos, así como acceder a la información detallada de los navíos/puertos seleccionados. Como en otras aplicaciones se puede ver el tipo de navío, así como el GRT y el DWT, siendo posible acceder a una mayor cantidad de información en función de la licencia, se detallará más adelante.

Esta aplicación permite la descarga de datos sobre puertos, de manera directa en un formato .xls. Sin embargo, existe un problema y es que no es posible descargar más que un archivo con 84 registros sin estar registrado. Una vez registrado, si se desea descargar información se debe disponer de los créditos suficientes como para efectuar el pago y descargar la información. En la tabla 3.3 se pueden ver los créditos que se consiguen al mes en función de cada tipo de licencia. La aplicación asociada a esta plataforma permite gran cantidad de filtrados para encontrar y clasificar barcos, por ejemplo en función del tipo de barco o de los atributos. En la tabla 3.3 se explicarán los diferentes campos que forman el fichero que podemos descargar de manera gratuita.

Campo	Descripción
Name	Nombre asociado al navío
Country	País que representa el navío
IMO	Organización Marítima Internacional (ver glosario)
MMSI	Número de identificación del servicio móvil marítimo (ver glosario)
Callsign	Señales de identificación que envían los navíos
Category	Tipo de navío
Vessel_type	Subcategoría dentro del tipo de navío
Length	Longitud del navío o eslora
Width	Anchura del navío o manga
DWT	Deadweight Tonnage (ver glosario)
gross_tonnage	Ver glosario
Location	Localización del navío en el momento que descargamos el fichero
Last Update	Fecha que indica la última modificación de la posición del navío
Public link	Enlace a los detalles del navío en la página de FleetMon

Tabla 3.3: Campos que conforman el fichero descargado de FleetMon

Esta aplicación nos permite estar al corriente de las últimas noticias marítimas existentes gracias al componente "Maritime News". Esta aplicación trabaja con datos AIS obtenidos tanto vía terrestre como vía satélite, esta última es una nueva opción que permite el seguimiento de navíos en mar abierto, consiguiendo así un mayor volumen de datos sobre los navíos y sus movimientos. De esta forma se consigue mejorar tanto la seguridad de los navíos, ya que ante cualquier problema localizarlos es muy rápido, como facilitar a los gerentes de las compañías un mejor seguimiento de los mismos.

Una de las principales características de esta plataforma es la posibilidad de rastrear navíos, utilizando el explorador FleetMon Explorer, gracias a este explorador es posible conocer la densidad del tráfico o las condiciones temporales a las que se encuentra expuesto el navío entre otras opciones. Sin embargo este componente sólo está disponible para usuarios registrados. Esta aplicación permite obtener diversas herramientas, dentro de las cuales podremos destacar

tres: Vessel Risk Rating, aplicación encargada de comprobar el riesgo potencial al que se ve expuesto un navío; OPIS Tanker Tracker, mediante la que es posible realizar seguimientos a navíos que transportan productos refinados, pudiendo conocer el cargamento de cada navío; y por último, tenemos Agricultural Commodities Trading Database, esta aplicación es muy interesante ya que nos ofrece una base de datos de comercio de productos agrícolas en la que es posible acceder a los registros detallados de buques que contienen commodities. Esta aplicación cumple con varios de los requisitos destacados del proyecto, como el nombre de la mercancía del navío o la cantidad de esta. Se puede ver la información obtenida utilizando esta licencia en la tabla 3.4. Esto podría resultar de gran interés para el proyecto a desarrollar. Sin embargo en la versión gratuita existe cierta información que no podemos visualizar, como se puede ver en la figura 3.9.

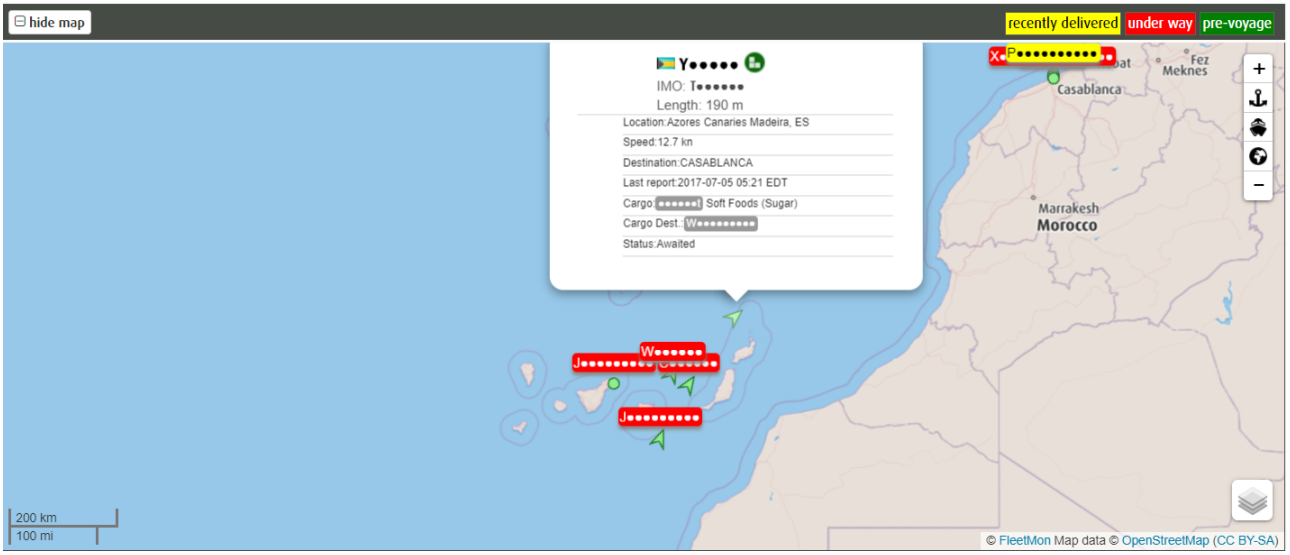


Figura 3.9: Agricultural Commodities Map

Pese a que es posible visualizar los datos actuales de esta aplicación sin necesidad de obtener ninguna licencia no existe la posibilidad de acceder a datos históricos sobre estos navíos a no ser que se obtenga la licencia para ello, esta no se consigue comprando alguna de las licencias de la tabla 3.5, sino que se debe adquirir por separado, lo que conlleva un coste adicional de 395€/mes.

Campo	Descripción
Comgroup	Tipo de mercancía (líquido, grano, fertilizantes)
Qty	Peso de la mercancía (en toneladas)
Comdesc	
Origin 1	Primer puerto de origen
Origin 2	Segundo puerto de origen
Origin 3	Tercer puerto de origen
Vessel	Nombre de la embarcación
IMO	Identificador del navío, sólo dentro de la IMO
Status	Estado actual del navío (navegando, amarrado, cargando)
Location ais	Posición actual
Destination ais	Destino
Eta ais	
Opetype	Estado de la carga
OpeCountry	País en el que se ha cargado el navío
Opecountryiso	Siglas que identifican el país de manera única
Opestart	
Eta	
Opeend	
Depcountryiso	Identificación del país de destino de la carga
Depcountry	País al que se realiza el envío
Depcode	Código del puerto
Depport	Nombre del puerto
Descountryiso	Identificador del país de destino del navío
Descountry	Nombre del país de destino del navío
Descode	Identificador del puerto de destino del navío
Desport	Nombre del puerto de destino del navío
Update	

Tabla 3.4: Campos del fichero de Agricultural Commodities Trading Database

A continuación se podrán ver las diferencias existentes entre las licencias que oferta esta plataforma.

Servicio	Free	Unlimited	Unlimited Sat 15	Unlimited Sat 50
Vessel Database	Access all vessel	Access all vessel	Access all vessel	Access all vessel
Terrestrial AIS	Sí	Sí	Sí	Sí
Satellite AIS	No	No	Sí	Sí
FleetMon Explorer	15 min por día	Sí	Sí	Sí
FleetMon Explorer App for presentation screens	No	Sí	Sí	Sí
Seacharts, EU/US inland	Sí	Sí	Sí	Sí
Seacharts global	No	Opcional	Sí	Sí
Traffic Density	No	No	Sí	Sí
Historial Tracks	No	Sí	Sí	Sí
Wind Conditions	No	Sí	Sí	Sí
Wave Height	No	Sí	Sí	Sí
Ocean Currents	No	Sí	Sí	Sí
My Fleet	Max 10	Max 100	ilimitado	ilimitado
My Alerts	Sí	Sí	Sí	Sí
Satellite AIS Hostlist	No	No	15 navíos y 5 cambios/mes	50 navíos 15 cambios/mes
Schedule Incl.container vessel schedules (vessel)	Los próximos 2 días	Los próximos 30 días	Los próximos 30 días	Los próximos 30 días
Event Log	Las últimas 24 horas	Los últimos 30 días	Los últimos 30 días	Los últimos 30 días
Port Call Log (Vessel)	Los últimos 7 días	Los últimos 12 meses	Los últimos 30 días	Los últimos 30 días
Speed Statistics	No	Sí	Sí	Sí
Draught Statistics	No	Sí	Sí	Sí
Advanced Filters	No	Sí	Sí	Sí
Ex-names history (AIS)	No	Sí	Sí	Sí
Port Coverage	Sí	Sí	Sí	Sí
Scheduled Arrivals Incl.container vessel schedules (port)	Los próximos 2 días	Los próximos 30 días	Los próximos 30 días	Los próximos 30 días
Port Call Log	Los últimos 7 días	Los últimos 12 meses	Los últimos 12 meses	Los últimos 12 meses
Vessel Search API	No	Sí	Sí	Sí
My Fleet API	No	Actualizado cada hora	Actualizado cada hora	Actualizado cada hora
Futher API Endpoints	No	Disponible	Disponible	Disponible
Public Profile	Sí	Sí	Sí	Sí
PhotoBox	Sí	Sí	Sí	Sí
Private Messages	Sí	Sí	Sí	Sí
Free Credit Points	No	30/mes	2000/mes	100000/mes
FleetMon Mobile enabled	Sí	Sí	Sí	Sí
Maritime News	Sí	Sí	Sí	Sí
Without external ADS	No	Sí	Sí	Sí
Support	No	e-mail, teléfono y helpdesk	e-mail, teléfono y helpdesk	e-mail, teléfono, helpdesk y account manager
Precio	Gratis	54€/mes	269€/mes	539€/mes

Tabla 3.5: Tipos de Licencias de FleetMon

Es posible acceder a esta aplicación a través del siguiente enlace: https://www.fleetmon.com

MarineTraffic. Aplicación web que permite la visualización, vía mapa, de navíos y puertos a lo largo del mundo gracias al uso de datos AIS obtenidos vía satélite. Como podemos ver en la figura 3.10 es una de las aplicaciones que almacena información asociada a un mayor número de navíos y puertos.

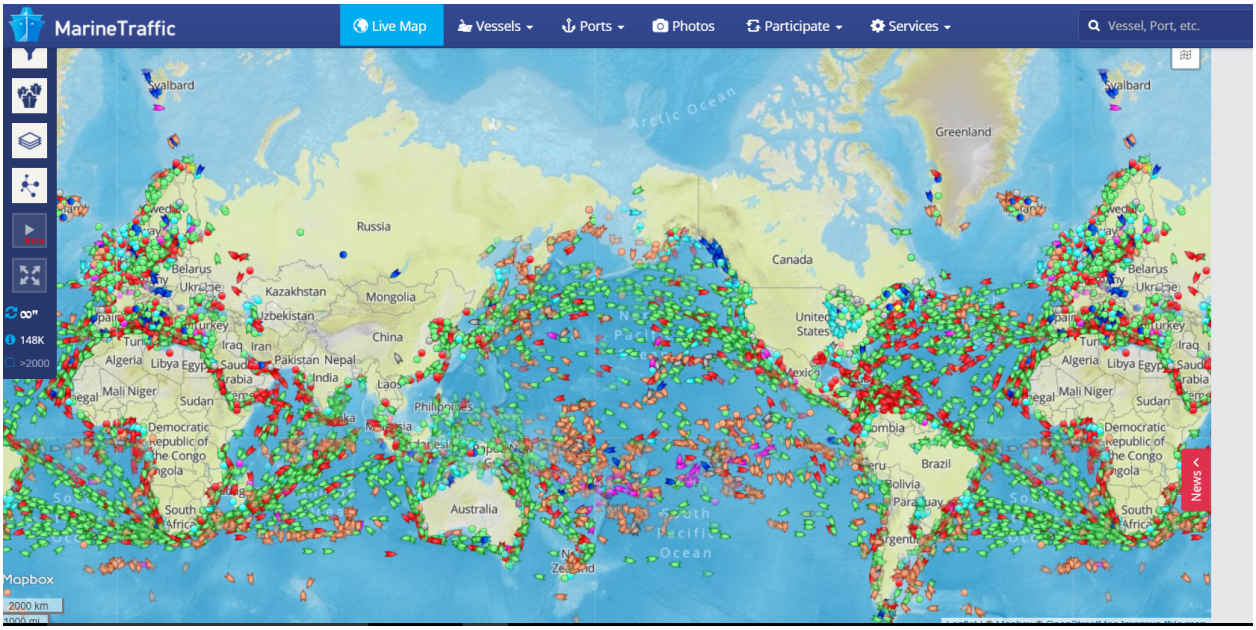


Figura 3.10: Mapa de MarinneTraffic

Uno de los motivos por los que tiene acceso a tantos datos es que los obtiene vía satélite, al encontrarse asociados con Orbcomm, uno de los principales proveedores de satélites AIS. Al igual que las anteriores aplicaciones es posible acceder a listados de navíos y puertos a partir de los que se podrá acceder a la información detallada de los elementos seleccionados. Sin embargo esta aplicación, en su parte pública, no permite visualizar estos listados, sólo acceder al mapa y a la información de algunos navíos vía Pop-Up.

En esta plataforma existen distintos tipos de cuentas privadas en función de la licencia con la que decidas registrarte, para comenzar se utilizó la licencia gratuita (en la tabla 6 se pueden ver las diferencias existentes entre cada licencia). Este tipo de licencia ya sí que permite acceder a listados de navíos y puertos, así como a parte de la información detallada de cada uno de estos. Esta aplicación nos permite visualizar el tipo de navío, el GRT y el DWT, así como gran cantidad de información sobre la localización de los mismos. Para una mayor información de la carga se debería acceder al campo Tonnage/Capacity, sin embargo como vemos en la figura 3.11 no se encuentra disponible con la licencia actual. El principal problema es que no ofrece información sobre la licencia que hay que obtener para acceder a dicha información.

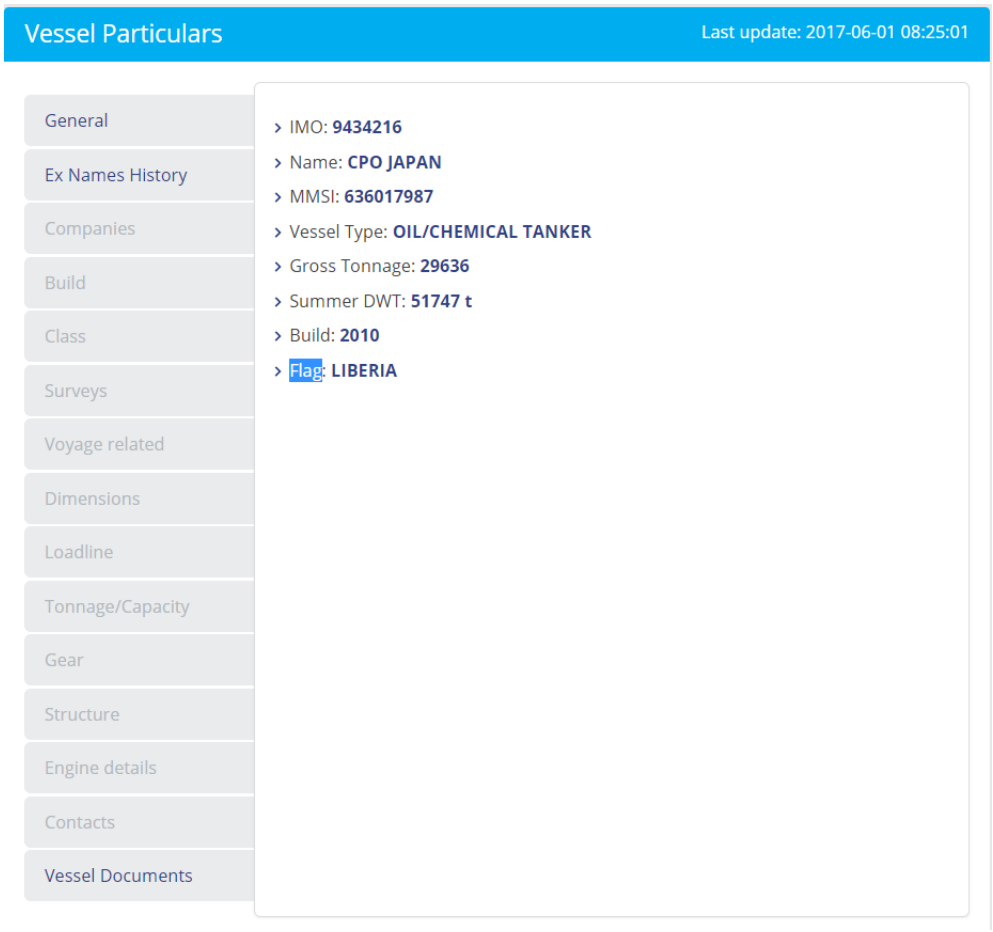


Figura 3.11: Información detallada de un navío

Es posible descargar los datos actuales de los navíos, aunque esto puede acarrear dos problemas. El primero reside en que los datos obtenidos no tienen el mismo nivel de detalle que los que se pueden ver a través de la herramienta, es posible que con alguna otra licencia los datos descargados sean mayores. Los datos que se pueden obtener vía descarga son los que se muestran en la figura 3.12.

Flag	Imo	MMSI	Vessel Name	Latest Position	Current Port	Last Known Por	Area Name	Destination	ETA	Course	Heading
Liberia	9434216	636017987	CPO JAPAN	2017-07-05 10:17		LA PAMPILLA	West Coast Central A	YOSU	2017-07-25 10:00	310	311
USA	6727868	367177810	GRACE MORAN	2017-07-05 10:17	MOREHEAD CITY	MOREHEAD CITY	US East Coast	MOREHEAD NC	1900-01-01 00:00	290	511

Figura 3.12: Datos descargados en la versión *Basic (Free)*

Otro problema que se puede encontrar al obtener datos es el número de registros que es posible exportar, esto se debe a que en función de la licencia adquirida se permite descargar un determinado número de registros mensuales. En la tabla 3.6 se puede ver cuantos se pueden descargar en función de cada licencia.

A continuación se mostrarán las diferencias existentes entre las licencias que oferta la plataforma. Los precios mostrados se corresponden con el coste anual de la licencia.

Service	Basic (Free)	Premium	Pro	Sat
Live Map Usage	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Terrestrial AIS Coverage	Global	Global	Global	Global
Satellite AIS Coverage	No	No	No	Global, 15 navíos sin retraso y el resto con 12 horas.
Density Map	Global	Global	Global y por tipo de navío	Global y por tipo de navío
Open Maps	Sí	Sí	Sí	Sí
Nautical Maps	No	No	Sí	Sí
Auto-Refresh Map	5min	1 min	1 min	1 min
Custom vessel icons	No	Sí	Sí	Sí
Full Screen Map	No	Sí	Sí	Sí
Wind Forecast	24 horas	72 horas	72 horas	72 horas
Multiple vessel tracks	1	2	10	10
Advanced Filters	No	No	Sí	Sí
My Fleet	Max 5	Max 50	Max 500	Max 500 y 15 por satélite
E-mail Notifications	30/mes	300/mes	1000/mes	1500/mes
SMS Notifications	0	10/mes	50/mes	75/mes
Mobile Push	30/mes	300/mes	1000/mes	1500/mes
My Custom areas	10	Ilimitado	Ilimitado	Ilimitado
Voyage Forecast	No	No	Sí	Sí
Vessel Master Data	No	No	Sí	Sí
Vessel Timeline	No	No	Sí	Sí
Vessel Database	Sí	Sí	Sí	Sí
Voyage History	3 días	30 días	90 dias	90 días
Add your vessel	Sí	Sí	Sí	Sí
Export Data	5 registros/mes	500 registros/mes	3000 registros/mes	3000 registros/mes
API Services	Coste Extra	Coste Extra	Coste Extra	Coste Extra
Rate Photos	Sí	Sí	Sí	Sí
Upload Photos	Sí	Sí	Sí	Sí
No Third-Party ads	No	Sí	Sí	Sí
Support	Knowledge base	Knowledge base + helpdesk	Knowledge base + helpdesk	Knowledge base + helpdesk
Precio	Gratis	9€/mes	79€/mes	269€/mes

Tabla 3.6: Licencias de MarineTraffic

Una de las características principales de esta plataforma reside en la posibilidad de descargar un histórico de datos y posiciones tanto de navíos como de puertos. Los datos que se obtienen se pueden ver en la tabla 3.8.

Tipo	Nombre	Descripción
Navíos	LON	Campo ecargado de posicionar al navío en el momento de envío de la señal
	LAT	Campo encargado de posicionar al navío en el momento de envío de la señal
	VESSEL MMSI	Identificador único del servicio móvil marítimo
	STATUS	Estado del navío en el momento de envío de la señal
	SPEED	Velocidad del navío
	COURSE	Rumbo sobre el fondo
	HEADING	Rumbo del navío
	TIMESTAMP (UTC)	Fecha en la que se envía el mensaje
Puertos	PORT ID	Identificador del puerto
	PORT NAME	Nombre asociado al puerto
	VESSEL MMSI	Identificador único del servicio móvil marítimo
	TIMESTAMP	Fecha y hora en la que se envía el mensaje
	ARR/DEP	Indicador de si el navío entra o sale del puerto

Tabla 3.7: Datos históricos

Sin embargo, al igual que en la aplicación VesselFinder, los datos de este histórico son de pago, variando el precio en función del número de posiciones del navío,o de llamadas a puerto, que se desean obtener . Además, como en la aplicación anteriormente mencionada, sólo es

posible acceder a la información de alguno de los navíos, ya que solicita que indiques mediante el MMSI los navíos de los que quieres obtener información.

En las tablas 3.8 y 3.9 se pueden ver los precios asociados al número de registros que se desea obtener. Es posible acceder a dicha aplicación mediante el siguiente enlace: https://www.marinetraffic.com

Nº de posiciones del navío	Precio
1.000	133€
10.000	327€
100.000	802€
1.000.000	1969€

Tabla 3.8: Precio sobre histórico de datos de navíos

Nº de llamadas a puerto	Precio
100	109€
1.000	292€
10.000	787€
100.000	2119€

Tabla 3.9: Precio sobre el histórico de los puertos

VTExplorer. Aplicación web encargada de proporcionar el programa VTExplorer, gracias al cual es posible visualizar en un mapa los diferentes navíos y puertos, como se puede observar en la figura 3.13, además de gestionar alertas y acceder a un histórico de la información de los navíos.

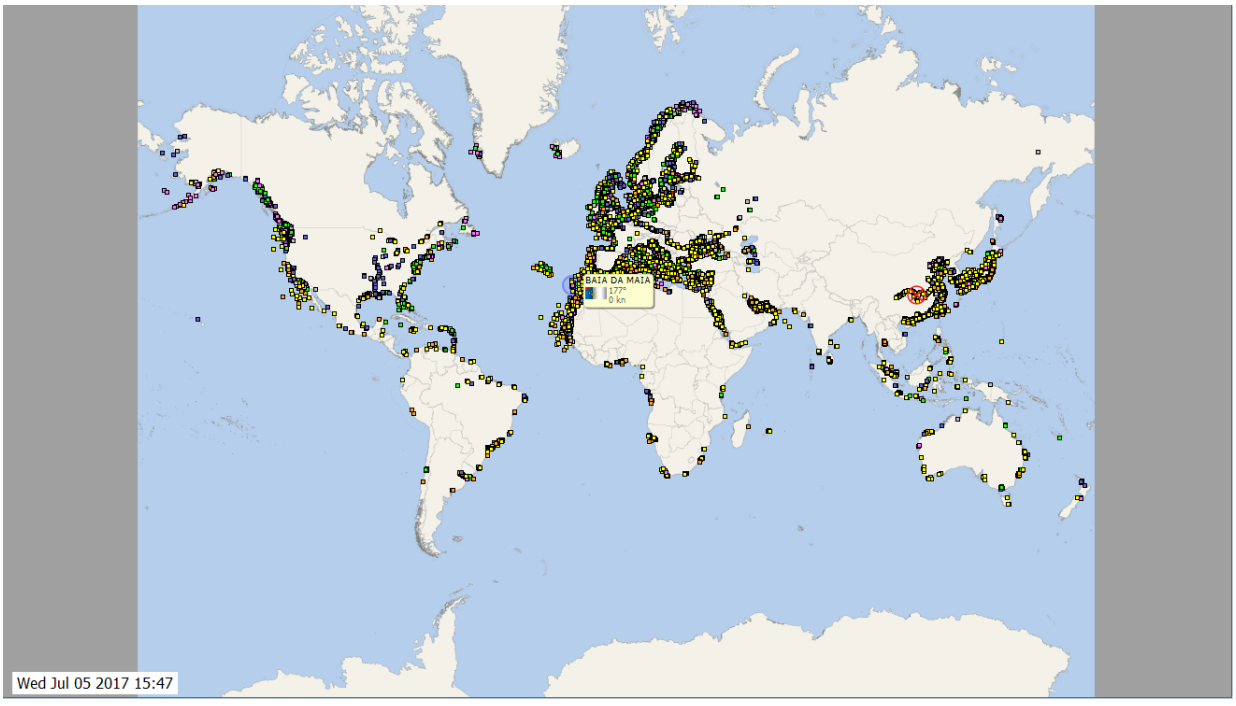


Figura 3.13: Mapa VTExplorer

La versión de la aplicación descargada es una demo fácilmente instalable en el ordenador, sin embargo esta aplicación no ofrece una gran cantidad de información ni se ha encontrado forma alguna de descargarla. Además dicha demo no permite realizar análisis sobre los movimientos de los navíos ni acceder a la información detallada de navíos y puertos, por ello no es posible estar seguros de que almacene información sobre la carga de los navíos que sea de utilidad. Existe la posibilidad de crear tu propia flota, sin embargo no detalla de cuantos barcos estará compuesta ni la información que nos aportara.

En caso de querer adquirir una versión que no sea la demo es necesario suscribirse a la plataforma, lo que conlleva un coste económico en función de la licencia elegida, esto se detalla en la tabla 3.10.

Servicio	Full Coverage	Danube River
Nº Usuarios	1	1
Coste 1 mes	50€	No hay datos
Coste 3 meses	143€	90€
Coste 6 meses	270€	160€
Coste 12 meses	480€	280€

Tabla 3.10: Costes VTExplorer

En la aplicación web indica que existe la posibilidad de obtener datos históricos sobre los navíos, de manera similar a como se obtenían en la aplicación VesselFinder, sin embargo tampoco se conoce el coste asociado a estos.

La utilidad de esta herramienta supone un interrogante, ya que no se ha podido encontrar diferencias entre ambas licencias.Es posible acceder a la plataforma usando el siguiente enlace: http://www.vtexplorer.com/

MarineCadastre. MarineCadastre es una aplicación web que ofrece datos de navíos, puertos y movimientos de navíos en las costas de EEUU entre los años 2009 y 2014. Esta aplicación utiliza datos AIS que pese a estar desactualizados pueden ser de utilidad para hacerse a la idea de los campos existentes en los datos AIS. Hay que tener en cuenta que además de estar desactualizados estos ficheros no almacenan ni el nombre ni el callsign de los navíos, el cual sería muy interesante conocer.

Estos datos son problemáticos, ya que para su visualización es necesario usar ArcGis, siendo esta herramienta de pago. Pese a ser una opción para obtener una idea de los datos que se podrían obtener no es útil por su compleja visualización. Sin embargo el análisis de dicha plataforma nos ha llevado a descubrir otra plataforma de interés, ChoroChronos, de la que se hablará más adelante, gracias a la cual se podrán obtener datos de ejemplo que ayuden a la realización de una primera versión del modelo conceptual.Es posible descargar estos datos mediante el siguiente enlace: https://marinecadastre.gov/ais/

AisHub. Esta es la aplicación más interesante para obtener datos que se ha encontrado. Esta aplicación permite recibir datos AIS, ya que el objetivo de esta plataforma es el de liberar el uso de datos AIS. El principal problema encontrado en esta plataforma reside en que antes de poder recibir datos AIS primero se tiene que haber aportado a dicha comunidad datos AIS sin procesar (NMEA), por lo que es necesario el uso de un receptor en caso de no tener dichos datos.

AisHub oferta también una aplicación de escritorio, tanto para sistemas Windows, como Linux o Mac, llamada AIS Dispatcher. Esta herramienta se encarga de recibir datos AIS, eliminar mensajes NMEA duplicados, calcular estadísticas sobre mensajes o realizar el muestreo descendiente de los mensajes NMEA. Existen herramientas como freenmea.net que permiten transformar datos AIS a ficheros con formato JSON o CSV.

ChoroChronos. ChoroChronos es una aplicación web que facilita la obtención de una gran cantidad de Datasets reales y cuya descarga es gratuita. En el caso que nos ocupa esta aplicación permite la descarga de varios Dataset sobre la navegación marítima en regiones de Francia y Grecia. Pese a que estos datos no están muy actualizados pueden permitir una primera aproximación al problema y a través de ellos realizar un posible modelo conceptual.

A continuación, en la tabla 3.11 podremos ver los distintos campos que componen el fichero, así como que describe cada uno.

Campo	Descripción
MMSI Number	Identificador único del servicio móvil marítimo (ver glosario)
Time	Campo compuesto por la fecha y la hora en la que se recibe la señal del navío
Longitude	Campo encargado de posicionar el navío en el momento de envío de la señal
Latitude	Campo encargado de posicionar el navío en el momento de envío de la señal
Heading	Dirección o rumbo del navío
Speed	Velocidad del navío
COG	Rumbo sobre el fondo
ROT	Velocidad de giro
shipCode	

Tabla 3.11: Campos que componen el fichero obtenido a través de esta aplicación

3.2. Conclusiones

En esta sección se persigue realizar un breve resumen en forma de tabla en el que se trata de trazar las características que en un inicio se consideran de mayor interés con las diversas aplicaciones analizadas, con el que hacer una comparativa entre las mismas. En esta sección también se explicará la aplicación elegida para la obtención de datos, así como el motivo de su elección. Es posible ver esta comparativa en la tabla 3.12.

Aplicación	Visualización	Obtencion de datos	Descarga	Cargamento
MyShipTracking	Mapas+Listados	Terrestrial AIS	Si, dando datos AIS	No
ShippingExplorer	Mapas+Listados	Terrestrial AIS ¿+ Satellite AIS?	Aportando datos AIS	Sí
ShipFinder	Mapa	Terrestrial+Satellite AIS	No	No
VesselFinder	Mapas+Listados	Terrestrial AIS	Pago, históricos+tiempo real	Algún dato en premium
FleetMon	Mapas+Listados	Terrestrial+Satellite AIS	De pago, créditos en función de la licencia. Históricos + tiempo real	Sí, con Agricultural Commodities Trading Database. Es de pago
MarineTraffic	Mapas+Listados	Terrestrial+Satellite AIS	De pago, históricos+tiempo real	Sí, en función de la licencia
VTExplorer	Mapas	Terrestrial AIS	De Pago	No
MarineCadastre	No	Terrestrial AIS	Gratis y desactualizados	No
AisHub	Listados	Nos se sabe, problema en los datos de prueba	Formato NMEA, si aportas datos la descarga es gratis	No se sabe
ChoroChronos	No	Terrestrial AIS	Sí, aunque no son muchos datos ni están muy actualizados	No
EMSA	No	Satellite LRIT	Permiten la descarga en función de una normativa de su página	
IMDatE	No, aún en desarrollo	Terrestrial+Satellite AIS y LRIT	No, esto lo gestiona EMSA	No se sabe

Tabla 3.12: Resumen de las aplicaciones previas

La aplicación seleccionada para la obtención de datos ha sido AISHub. El principal motivo de esta elección radica en la posibilidad de haber accedido a una licencia mensual gratuita, gracias al hecho de ser estudiante. Esta licencia permite la descarga de datos sobre el tráfico marítimo mundial, limitado a los mensajes recibidos por estaciones costeras. Para ello AISHub permite el acceso a su API a través de una url dada pero siempre para una misma IP. Con el fin de obtener esta información se ha desarrollado una aplicación java de descarga de datos, la cual se encarga de descargar ficheros con formato csv de dicha API cada 90 segundos.

Capítulo 4

Big Data

En la actualidad es un hecho que los datos son el motor del mundo, debido a esto, se han producido una gran cantidad de avances, tanto en arquitecturas como en tencnologías, con el fin de satisfacer las capacidades de análisis y almacenamiento derivadas de la cada vez más creciente generación de datos. Estos avances se engloban dentro del paraguas que ofrece Big Data. Se considera Big Data a todas las arquitecturas y tecnologías encargadas de almacenar y gestionar datos cuyo tamaño, complejidad y escalabilidad son elevados. Existen una gran cantidad de definiciones que tratan de explicar qué es Big Data, por ejemplo, Oxford English Dictionary (OED) ha definido Big data como *"data of a very large size, typically to the extent that its manipulation and management present significant logistical challenges"* .

Pese a las definiciones tradicionales de Big Data, como la vista anteriormente, es prácticamente imposible definir Big Data sin pensar en las 3 Vs, término utilizado con el fin de identificar sus principales características. Normalmente se identifica a las 3 Vs como:

1. **Volumen**: En la actualidad es posible acceder a enormes conjuntos de datos, debido principalmente al grán volumen de información que es posible captar gracias a satélites o a las redes sociales. Este hecho supone un gran problema a la hora de almacenar esta información, debido, principalmente, a que los sistemas tradicionales tienen problemas para tratar la gran escalabilidad asociada a los datos actuales, además de problemas asociados a la capacidad de almacenamiento de los hardware. Ante la necesidad de grandes capacidades de almacenamiento, a poder ser con bajo coste, nace Apache Hadoop, plataforma tecnológica opensource utilizada para el tratamiento de grandes volúmenes de datos y su almacenamiento distribuido.

2. **Velocidad**: La posibilidad de obtener datos en streaming, es decir, en tiempo real, unido con la imperiosa necesidad de obtener datos en el menor tiempo posible, puede dar lugar a problemas a la hora de procesar los datos en las base de datos tradicionales. Big Data permite mejorar los flujos de entrada de información, así como procesar grandes volúmenes de información a gran velocidad, eliminando los cuellos de botella comunes en las bases de datos tradicionales.

3. **Variedad**: En la actualidad los datos no provienen únicamente de sistemas relacionales, sino que son enviados a través de otros sistemas distintos, como sensores o redes sociales. La gran variedad de datos da lugar a problemas de procesamiento de los mismos, es sabido que los datos estructurados son los más fáciles de procesar. Estos datos se dividen, según su estructura, en 3 tipos:

- **Datos estructurados**: Se denominan datos estructurados a todos aquellos con una estructura totalmente definida, delimitando, por ejemplo, la longitud y el formato de los datos. Suelen estar sujetos a una gran cantidad de normas bastante estrictas. Estos datos suelen ser almacenados en bases de datos relacionales, en las que los datos se almacenan en tablas, con un formato previo.
- **Datos semi-estructurados**: Estos datos, pese a no presentar una estructura perfectamente definida, si que cuentan con una ligera organización interna que facilita su tratamiento. Esta organización se basa principalmente en los metadatos, encargados de describir los objetos y sus relaciones. Dentro de este tipo de datos destacan XML y JSON.
- **Datos no estructurados**: Estos datos no se ajustan a los formatos convencionales, al no estar dotados de estructura alguna. La generación de este tipo de datos se lleva a cabo a gran velocidad, debido principalmente a las redes sociales, ya que los tweets, imágenes o vídeos entran dentro de este tipo de datos. En la actualidad la mayor parte de la información generada no tiene estructura interna

Aunque las 3 Vs están mundialmente aceptadas, actualmente han surgido nuevos autores y empresas, como IBM, que sostienen que estas pueden ampliarse a 5 Vs, añadiendo para ello las siguientes características:

4. **Veracidad**: Uno de los problemas de las ingentes cantidades de datos que se pueden captar en la actualidad es su grado de veracidad. Esta característica está estrechamente relacionada con la limpieza de datos y el análisis de los datos obtenidos, tratando de mejorar la calidad de la información, con el objetivo de alcanzar una correcta toma de decisiones. A menudo los problemas de veracidad de los datos son compensados con el gran volumen de datos obtenido.

5. **Valor**: La posibilidad de almacenar grandes volúmenes de información no es de mucha utilidad, a no ser que sea posible dar valor a estos datos. Esta característica es fundamental en estos sistemas, ya que el hecho de almacenar información poco precisa puede dar lugar a graves problemas una vez sea utilizada. La importancia del valor de los datos se puede ver en las empresas que persiguen obtener un mayor conocimiento sobre su mercado, gracias al análisis de estos datos. En estos casos la toma de decisiones se puede ver afectada enormemente por la inexactitud de los datos captados. Las empresas están destinando cada vez más recursos al análisis de los datos obtenidos, ya que almacenar datos de los que no se puede extraer valor puede ser muy costoso.

Otro aspecto a destacar dentro de las soluciones Big Data es el Gobierno de Datos (*Data Governance*), capacidad de una organización para gestionar el conocimiento que tiene sobre su información. Este aspecto proporciona un enfoque completo para mejorar y administrar la información obtenida. Esto es de vital importancia en las organizaciones, ya que un mal gobierno de datos forma un camino directo al descontrol de los datos, el cual puede alcanzar grandes proporciones al trabajar con un gran volumen de información. Esta es una de las principales causas de muerte por éxito de los proyectos Big Data y las empresas que los llevan a cabo, ya que la gestión tradicional en proyectos de este tamaño es muy complicada.

Gracias al Big Data, junto con los grandes avances actuales en el campo de la Inteligencia Artificial, es posible crear soluciones a problemas que no nos habíamos planteado. Sin embargo

el Big Data no es perfecto, como se puede ver en el artículo publicado por Forbes *3 Massive Big Data Problems Everyone Should Know About* [18], existe una creciente preocupación por la privacidad de los datos y su seguridad así como las posibles ambigüedades presentes en estos.

El auge del Big Data ha supuesto una revolución tecnológica de gran dimensión, favoreciendo al desarrollo de una gran cantidad de arquitecturas, como Data Warehouse y Data Lake, así como la creación de una gran cantidad de puestos de trabajo. En la siguiente sección se explicará la arquitectura que se propone como solución en este proyecto, el Data Lake.

4.1. Data Lake

La preocupación actual, tanto de empresas como de particulares, por la gestión de sus datos es enorme. Como previamente se ha comentado, el volumen de datos que se generan en la actualidad está sufriendo un crecimiento exponencial, principalmente debido al auge de las redes sociales. Este gran crecimiento está llevando de la mano la creación de nuevas arquitecturas encargadas de almacenar cualquier tipo de dato, estructurado, semi-estructurado y no estructurado. Cabe destacar el crecimiento de los datos no estructurados, el cual ronda un 63 % por año [14].

Es un hecho que gran parte de la información que se capta carece de un valor inmediato, es decir, se desconoce su utilidad inmediata. Sin embargo no conocer la utilidad actual de los datos no significa que en el futuro no adquieran una gran importancia en los procesos de negocio de las diferentes empresas. Ante esto, grandes empresas como Google almacenan toda la información que adquieren, tratando de buscar un valor a futuro. Esto es posible verlo gracias al artículo ¿Qué es un Data Lake? [3], en el cual se comenta que un ingeniero de Google dijo "*En Google guardamos todos los datos aunque no les veamos ningún valor hoy en día. No sabemos si dentro de 1, 5 o 10 años se nos ocurrirá una idea para explotarlos y perderlos sería un grave error*".

Hechos como los anteriormente mencionados, unidos a la gran heterogeneidad de las nuevas fuentes de datos, han evidenciado la necesidad de crear nuevas arquitecturas para almacenar y gestionar la información obtenida. En este ámbito comienza a surgir una nueva arquitectura, llamada Data Lake, mediante la cual se persigue almacenar y procesar cualquier tipo de información, estructurada, semi-estructurada y no estructurada, tratando de mejorar el tratamiento de la información para prevenir problemas de amigüedades en los datos. En las siguientes secciones se explicará en detalle qué es un Data Lake.

4.1.1. Definición

Un Data Lake es un repositorio donde se almacena cualquier tipo de datos, estructurados, semi-estructurados y no estructurados, sin ningún tipo de pre procesamiento ni de esquema, ofreciendo además mecanismos para refinar y explorar estos datos. En los Data Lake los datos se almacenan en bruto, es decir, son cargados directamente desde las fuentes originales sin descartar ningún elemento, almacenándose en su estado original. Al no existir un pre procesamiento, no existe ningún tipo de agregación ni de transformación, lo que permite disponer de toda la información potencial contenida en los datos. En la figura 4.1 es posible ver un pequeño resumen de lo que es un Data Lake así como alguno de sus usos, complementando lo explicado anteriormente.

El concepto Data Lake suele encontrarse íntimamente ligado con Apache Hadoop, plataforma open-source que permite el despliegue de esta arquitectura con un coste reducido. Sin

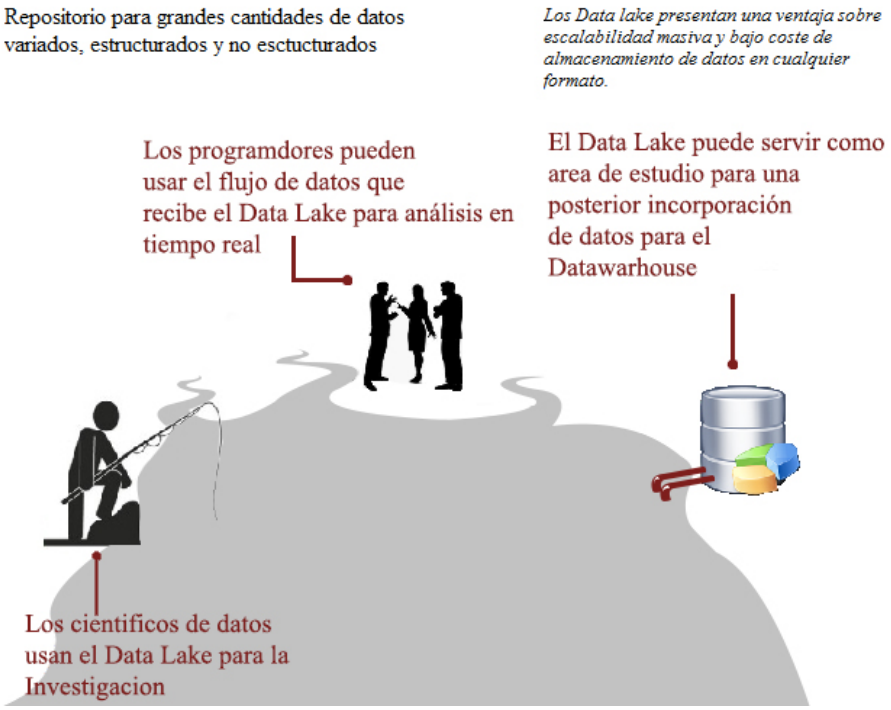


Figura 4.1: Resumen y ejemplo de usos de un Data Lake

embargo no es la única vía de desarrollo de un Data Lake, ya que es posible desarrollar este tipo de arquitecturas sin necesidad de Hadoop, aunque normalmente es lo más utilizado.

Esta arquitectura tiene grandes puntos de ruptura con las arquitecturas tradicionales, tales como los Data Warehouse. Un aspecto a destacar, dentro de los Data Lake, es el momento en el que se realiza el formateo de los datos. Esta arquitectura rompe en este punto con las arquitecturas tradicionales, ya que en esta los datos son formateados, dotados de estructura, únicamente en el momento en que son necesarios para satisfacer una necesidad de análisis. Esta propiedad, conocida como "*schema on read*", es una de las más destacadas de esta arquitectura, al distinguirla totalmente de las arquitecturas tradicionales, las cuales utilizan un "*schema on write*". El schema on read ha supuesto una gran innovación en el mundo de Big Data, ya que el hecho de poder almacenar los datos sin esquemas permite añadir y modificar grupos de datos fácilmente, reduciendo los costos asociados a su transformación, aunque también es posible contar con repositorios de esquemas que nos faciliten el acceso a los datos.

Además de la anterior existe una gran cantidad de características asociadas a los Data Lakes, sin embargo ninguna tiene la gran relevancia de los ELT (Extract, Load, Transform). En este tipo de arquitectura se sigue este sistema, según el cual los datos son extraídos de las fuentes y cargados como *Raw Data* en un sistema de archivos distribuido, a diferencia de las arquitecturas tradicionales, las cuales siguen un proceso ETL (Extract, Transform, Load). Esta característica es de gran importancia, ya que como tal es la que permite el almacenamiento de raw data. En la figura 4.2 se puede ver la diferencia existente entre los dos sistemas previamente explicados.

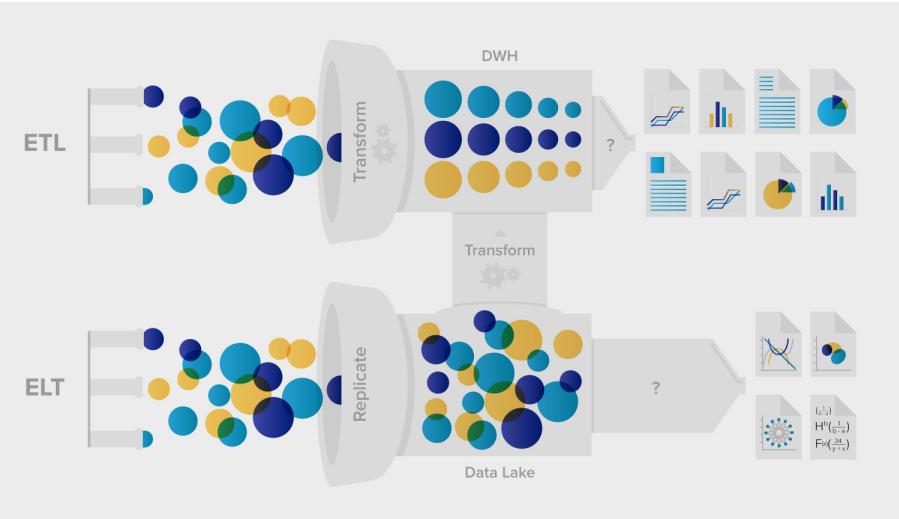


Figura 4.2: Diferencia existente entre ETL y ELT

La posibilidad de almacenar la información sin haberla dotado de un esquema da lugar a diversas ventajas, siendo un elemento decisivo a la hora de afrontar problemas de escalabilidad y heterogeneidad de los datos. Algunas de las principales ventajas de este tipo de sistemas son:

- La posibilidad de cargar los datos en bruto, directamente desde las fuentes originales y sin ningún tipo de preprocesamiento, facilita a los analistas procesar los datos con una mayor flexibilidad.

- Los datos no estructurados y semi-estructurados son soportados fácilmente y cargados a gran velocidad.

Sin embargo no todo son ventajas en este tipo de sistemas, por ejemplo, el hecho de almacenar los datos sin ningún esquema imposibilita su análisis, es decir, no es posible analizar datos a no ser que se les dote de un esquema.

Una de las preocupaciones más extendidas asociadas al uso de los Data Lakes deriva de la posible acumulación de datos incomprensibles. Esta preocupación surge debido a la falta de esquema de los datos almacenados, siendo la principal causa de convertir un Data Lake en un pantano de datos. Para intentar mitigar esta preocupación se creó un repositorio de metadatos, el cual se encarga de almacenar información de alto nivel sobre las entidades de datos a almacenar, como por ejemplo el tipo, el tiempo o el creador.

Otro de los principales problemas que pueden tener lugar en este tipo de arquitecturas radican en la potencial ambigüedad de los datos. Este hecho deriva de la heterogeneidad de las fuentes encargadas de captar datos, las cuales pueden obtener los mismos datos y así dar lugar a problemas de ambigüedad. Ante este problema adquiere una enorme importancia el gobierno de datos, ya que un buen gobierno de datos puede evitar en gran medida los problemas de ambigüedad en los datos.

El desarrollo de estos sistemas suele encontrarse muy ligado a la Inteligencia Artificial, la cual es capaz de potenciar las capacidades de los Data Lakes. Es muy común el desarrollo de inteligencias artificiales encargadas de facilitar la integración de las diversas fuentes de datos, analizar los conjuntos de datos, automatizando los procesos, así como utilizar los datos almacenados mediante *Machine Learning* para mejorar los diferentes flujos empresariales.

4.1.2. Arquitectura

La arquitectura de un Data Lake suele estar compuesta por 3 capas, compuestas a su vez por 3 niveles. Las capas que conforman este tipo de arquitectura son la capa de Gobierno de Datos y Seguridad (Data Governance and Security layer), capa de Metadatos (Metadata layer) y la capa de gestión del ciclo de vida de la información (Information Lifecycle Management layer). Sin embargo esta no es la única interpretación de la arquitectura de un Data Lake, ya que existen múltiples interpretaciones sobre la misma en función de los requisitos tanto empresariales como tecnológicos. Algunos de los principales factores que afectan a la arquitectura de un Data Lake son los siguientes:

- La forma de obtención de datos utilizada, como por ejemplo en tiempo real.
- La forma de almacenamiento de los datos.
- El gobierno de datos que se desea llevar a cabo.
- La profundidad de los metadatos.

La arquitectura por capas, mencionada anteriormente, es la más tradicional que se puede encontrar en un Data Lake, además de la más utilizada en la actualidad. Esta arquitectura a su vez también se encuentra caracterizada por 3 niveles distintos, en las cuales se agrupan funcionalidades de mayor similitud, es decir, con un nivel de abstracción menor. El flujo existente entre los 3 niveles es secuencial, mientras que los datos son transmitidos entre los niveles, las capas se encargan del procesamiento de los mismos. Estos niveles son:

- Nivel de entrada (Intake Tier)
- Nivel de gestión (Management Tier)
- Nivel de consumo (Consumption Tier)

En la figura 4.3 se muestra en detalle la arquitectura previamente explicada.

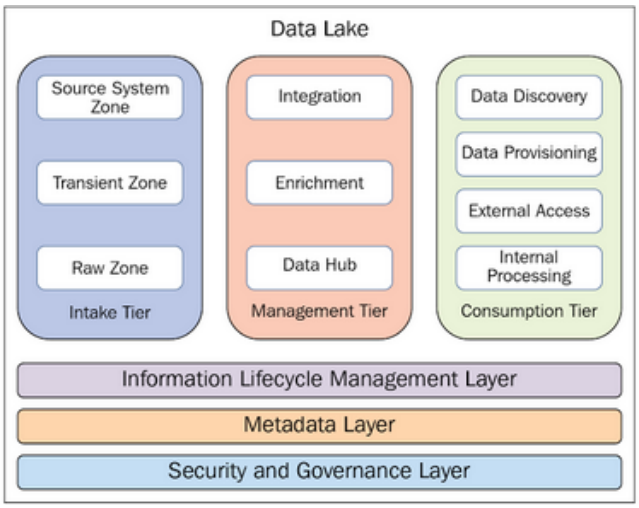


Figura 4.3: Arquitectura de un Data Lake

A continuación se explicarán las diversas capas y niveles que forman esta arquitectura.

Gobierno de Datos y Seguridad (Data Governance and Security layer). Capa encargada de regular el acceso y modificación a los datos de los Data Lakes. Para alcanzar la consecución de este objetivo esta capa documenta todo el proceso para el cambio y acceso a todos los datos. Este sistema realiza un seguimiento de todas las modificaciones llevadas a cabo en los distintos niveles de un Data Lake, combinándose con las normas de seguridad. Estas normas se encargan de garantizar la integridad de los datos, así como el control del acceso y autorización a los mismos. Un ejemplo de sistema que gestiona este tipo de capas es Kerberos, solución utilizada en Hadoop con el fin de garantizar el autocontrol de los usuarios. En la figura 4.4 se puede ver la funcionalidad de la capa de Gobierno de Datos y Seguridad, obtenida del libro "*Data Lake development with Big Data : explore architectural approaches to building Data Lakes that ingest, index, manage, and analyze massive amounts of data using Big Data technologies*"[25].

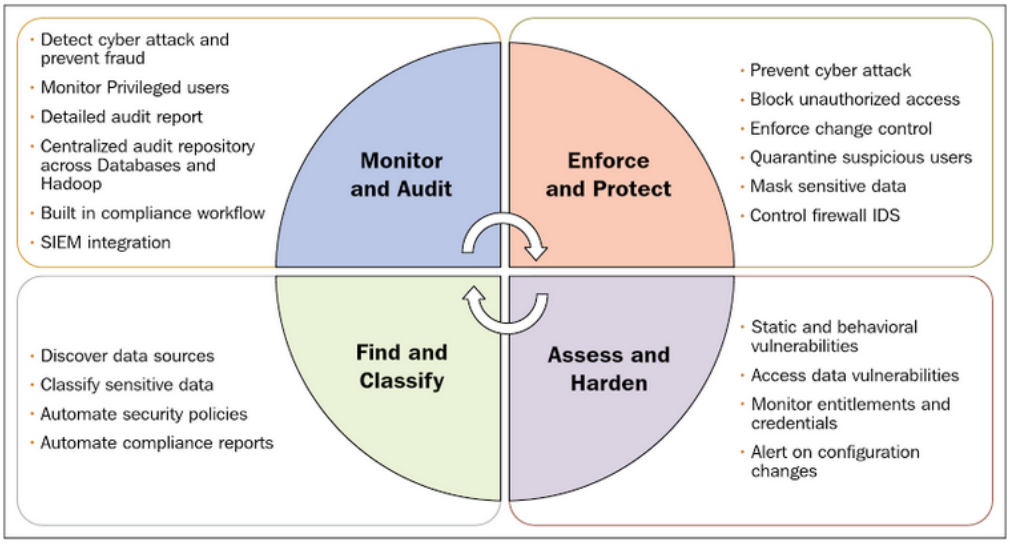


Figura 4.4: Gobierno de Datos y Seguridad

Capa de gestión del ciclo de vida de la información (Information Lifecycle Management layer). Esta capa se encarga de asegurar las reglas que gobiernan el almacenamiento de datos en un Data Lake. Estas reglas suelen utilizarse para gestionar la información almacenada a lo largo de su ciclo de vida. Esto se debe a que la información suele perder valor con el paso del tiempo, lo que eleva el riesgo de uso de la misma.

Esta capa por tanto trata de definir la estrategia y políticas a utilizar, tratando así de clasificar los datos de valor y determinar cuando deberían dejar de ser almacenados. Las herramientas, encargadas de realizar estas tareas, son automáticas y permiten limpiar y archivar los datos en función de las políticas a seguir.

En la figura 4.5, obtenida también a gracias al libro escrito por Pradeep Pasupuleti y Beulah Salome Purra, se puede ver las funcionalidades asociadas a esta capa.

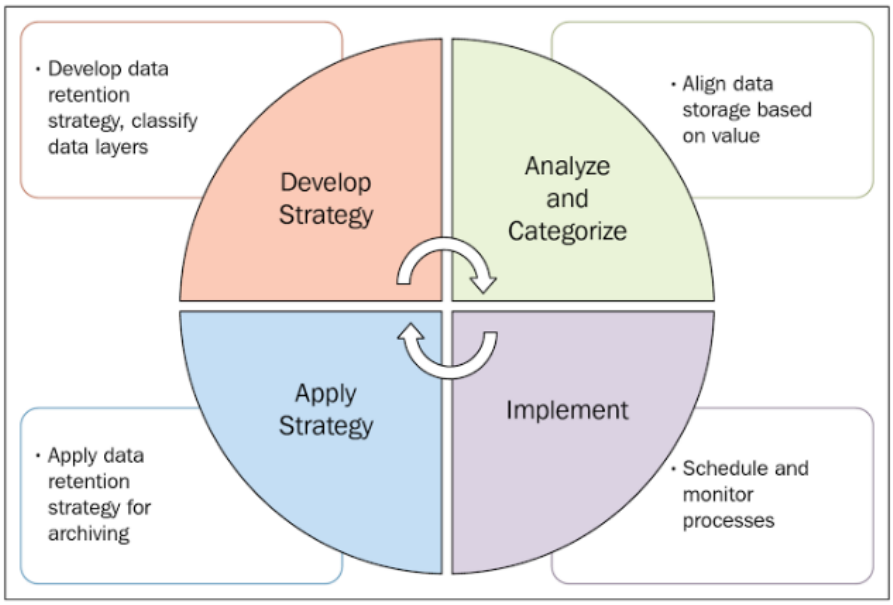


Figura 4.5: Funcionalidades de la Capa de gestión del ciclo de vida de la información

Capa de Metadatos (Metadata layer). La capacidad de un Data Lake de almacenar cualquier tipo de datos se encuentra especialmente ligada con esta capa. Esta capa trata de ofrecer un mecanismo mediante el cual se puedan encontrar los vínculos entre la información que se almacena y quién puede acceder a esta. Esta capa, por tanto, suele considerarse el corazón de un Data Lake.

En esta capa se almacena la información sobre los datos, a medida que son obtenidos, y la indexa con el fin de poder acceder antes a los metadatos que a los propios datos, facilitando en gran medida la accesibilidad a estos. La posibilidad de indexar esta información es de gran importancia en un Data Lake, ya que posibilita el acceso a la información almacenada, por mucho que los datos se almacenen en su formato original o incluso no tengan estructura, permitiendo así almacenar los datos en su formato original para después ser transformados en el momento en que vayan a ser usados. Es por tanto muy importante para definir la estructura de los ficheros almacenados en el *Raw Zone*, describiendo también las entidades existentes en los ficheros.

Esta capa es capaz de proveer de información de vital importancia al Data Lake, como por ejemplo sobre el significado de la información almacenada en el mismo. Una buena construcción de la capa de metadatos aumenta el potencial del Data Lake, permitiendo realizar gran cantidad de análisis mediante sistemas como *Machine Learning as a Service (MLaaS)* o *Data as a Service (DaaS)*. En la figura 4.6 se pueden ver las algunas de las capacidades asociadas a la capa previamente explicada.

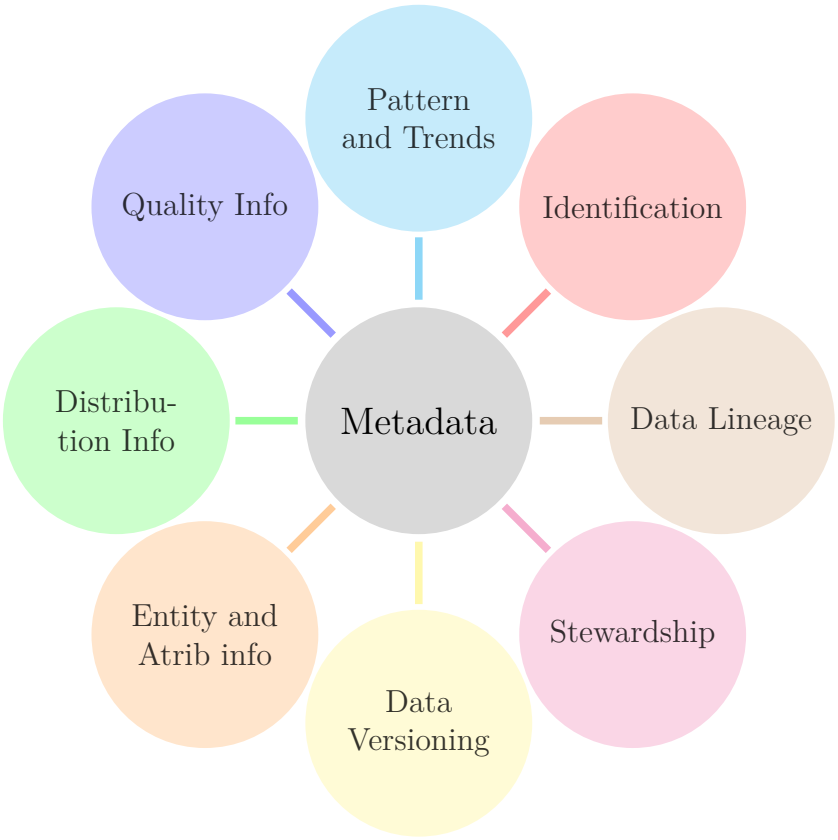


Figura 4.6: Capa de Metadatos

Nivel de Entrada (Intake tier). Nivel encargado de gestionar todos los servicios que se conectan a fuentes externas, así como del área de almacenamiento de los datos obtenidos. A su vez este nivel se encuentra divido en 3 zonas. Estas zonas son:

1. *The Source System Zone*: Zona encargada de gestionar las conexiones con las fuentes externas y la obtención de datos de dichas fuentes. Un hecho a tener en cuenta en esta zona es el tiempo de adquisición de la información, el cual depende de los requisitos de la aplicación a crear, pudiendo ser en tiempo real, en batch o en micro batch. En la figura 4.7 es posible ver los diferentes flujos de información que puede captar un Data Lake.

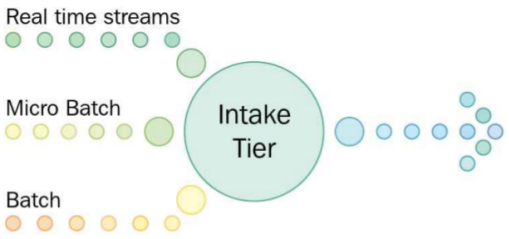


Figura 4.7: Tipos de flujo de entrada

2. *The Transient Zone*: Zona intermedia entre *The Source System Zone* y *The Raw Zone* encargada de almacenar la información los datos obtenidos, por fuentes, antes de ser enviados a la *Raw Zone*. Esta zona además se encarga de calcular el tamaño de los datos

obtenidos. Al realizar unas comprobaciones mínimas de validación de datos su inexistencia podría ocasionar bajadas en la calidad de los datos almacenados en la *Raw Zone*. En la figura 4.8 se pueden ver algunas de las funcionalidades asociadas a esta zona.

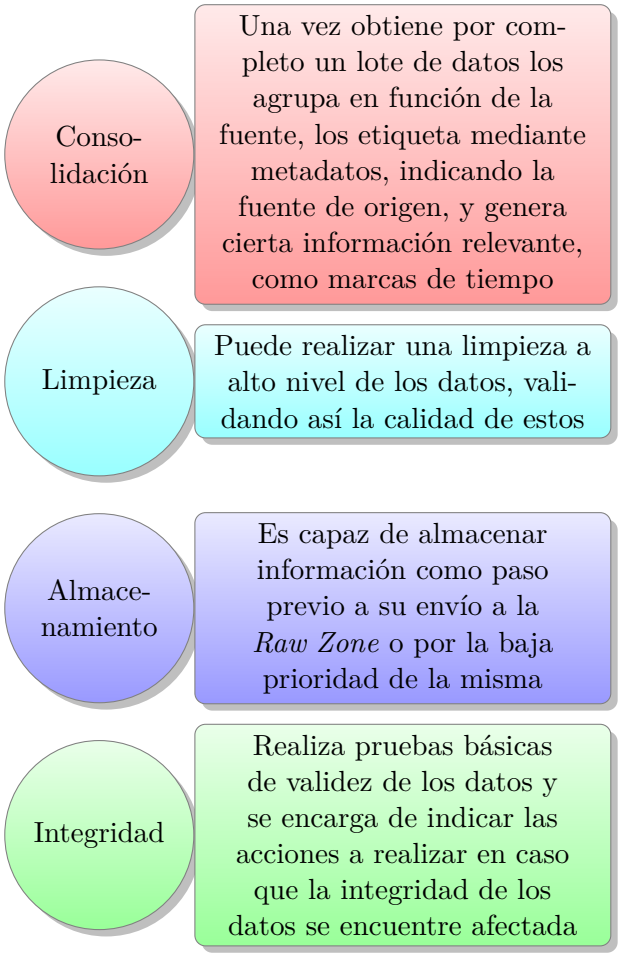


Figura 4.8: Funcionalidades de *Transient Zone*

3. *Raw Zone*: Zona encargada de almacenar la información obtenida. Para implementar esta zona se utiliza un sistema de archivos distribuidos, normalmente HDFS, incluyendo un área en el que se conservan los datos en su formato original. Este área varía en función de si los datos son obtenidos en tiempo real o por lotes. En la figura 4.9 es posible ver a alto nivel el esquema de uso de la *Raw Zone*.

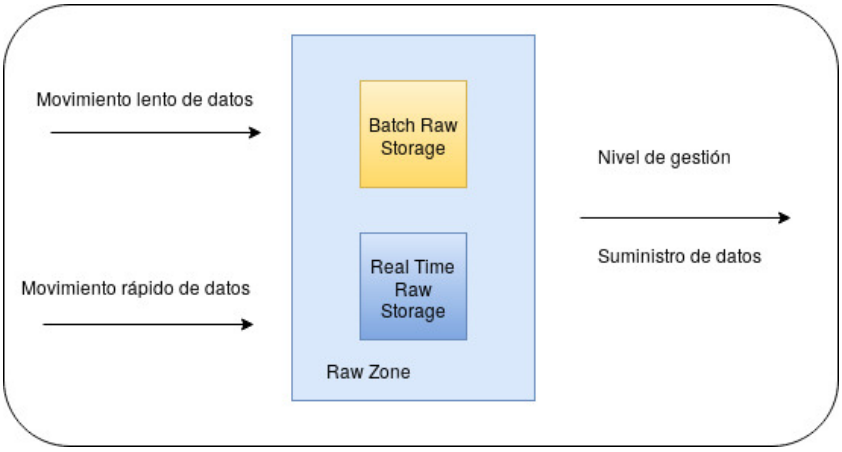


Figura 4.9: Funcionalidades de alto nivel asociadas a la Raw Zone

Nivel de Gestión (Management Tier). Nivel encargado del tratamiento y preparación de los datos, previamente almacenado en su formato original, para su análisis y uso futuro. Este nivel, al igual que el anterior, se encuentra divido en 3 zonas, a través de las cuales fluyen los datos de manera secuencial, hasta almacenarse en el *Data Hub*, zona donde se almacenan tanto datos estructurados como no estructurados. En este nivel se añaden y adjuntan los metadatos a cada fichero, creando así un identificador de los mismos, permitiendo seguir todos los cambios que tengan lugar en cada registro individual. La información asociada a los metadatos de cada fichero es de gran importancia, permitiendo mejorar la gestión de la calidad de los datos, realizar un seguimiento de la progresión de estos e incluso analizar las anomalías y correcciones a realizar a mayor velocidad. Las zonas que componen este nivel son las siguientes:

1. *The Integration Zone*: En esta zona se lleva a cabo la integración de las diversas fuentes de datos, sobre los que se aplican una serie de transformaciones creando una estructura estandarizada. En la figura 4.10 es posible ver las principales funcionalidades asociadas a esta zona, así como el flujo seguido por los datos en este nivel.

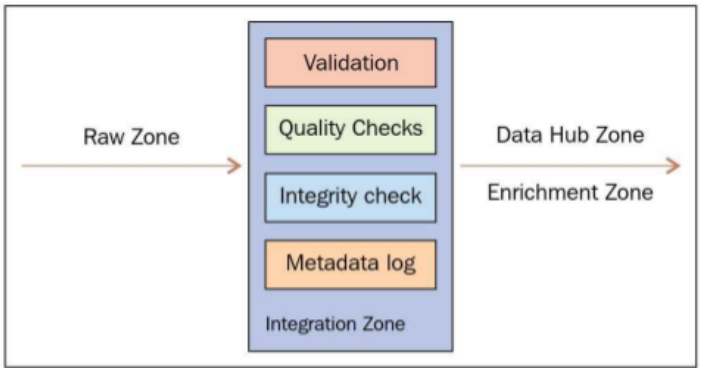


Figura 4.10: Funcionalidades asociadas a *The Integration Zone*

2. *The Enrichment Zone*: Zona encargada de proporcionar procesos para el tratamiento y gestión de los datos, pudiendo incluir procesos que permitan añadir nuevos atributos a los registros existentes a través de fuentes externas e internas. En esta zona se utiliza

un sistema de archivos distribuido, como puede ser HDFS, permitiendo así alcanzar una mayor flexibilidad en el tratamiento de estos datos. Esta flexibilidad radica en que estos sistemas, principalmente HDFS, permiten el almacenamiento sin necesidad de esquemas o índices, por lo que no es necesario que los datos sean tratados antes de ser usados. En la figura 4.11 es posible ver las diferentes capacidades asociadas a esta zona:

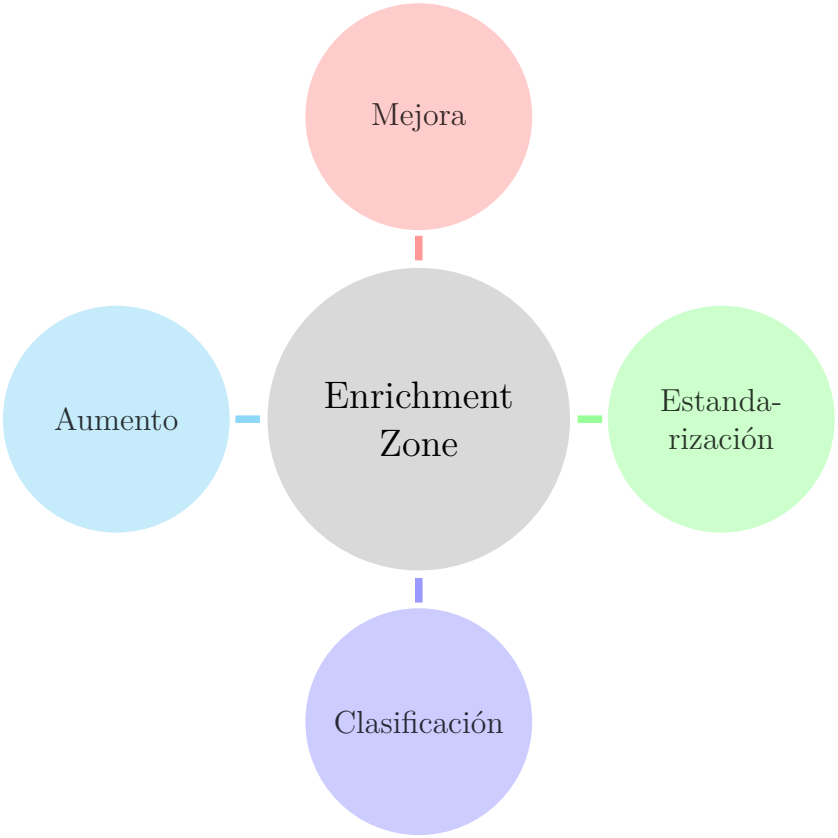


Figura 4.11: Funcionalidades asociadas a *The Enrichment Zone*

3. *The Data Hub Zone*: Zona encargada del almacenamiento de los datos una vez han sido limpiados y procesados, constituyendo el final del flujo de datos dentro del Nivel de Gestión. En esta zona se llevan a cabo diversos procesos de búsqueda y recuperación de la información, a través de diversas herramientas, como puede ser Apache Lucene. Estos procesos hacen posible llevar a cabo algunos descubrimientos, basándose para ello en los grandes conjuntos de metadatos almacenados con anterioridad. En esta zona es posible almacenar la información utilizando bases de datos relacionales, como Oracle o MS SQL Server, así como utilizando arquitecturas no relacionales, tales como Mongo DB o HBase.

Nivel de Obtención (The Data Consumption Tier). En este nivel es posible acceder a los datos almacenados tanto en *The Data Hub Zone* como en la *Raw Zone*, siendo capaces de llevara cabo una gran variedad de análisis sobre los mismos. Estos datos pueden ser usados también mediante herramientas de visualización, algo muy usado en la actualidad para realizar seguimientos de mercancías, o incluso por aplicaciones externas conectadas a través de servicios Web. El acceso a estos datos se encuentra altamente regulado, siguiendo una gran cantidad de

controles de seguridad cuya función es evitar el acceso a usuarios sin los suficientes permisos. Al igual que en los anteriores niveles, este nivel se encuentra dividido en diversas zonas:

1. *The Data Discovery Zone*: Zona considerada como la principal puerta de entrada para los usuarios externos que desean acceder al Data Lake. En esta zona se lleva a cabo un registro de los eventos que ocurren sobre los datos, el cual, unido al seguimiento que se realiza gracias a los metadatos, facilita a los usuarios el análisis de los mismos.

2. *The Data Provisioning Zone*: Zona en la que los usuarios pueden obtener los datos almacenados en el Data Lake, pudiendo suministrarse tango del *Data Hub Zone* como de la *Raw Zone*.

En la figura 4.12 es posible ver la estructura que forma el Nivel de Obtención previamente explicado.

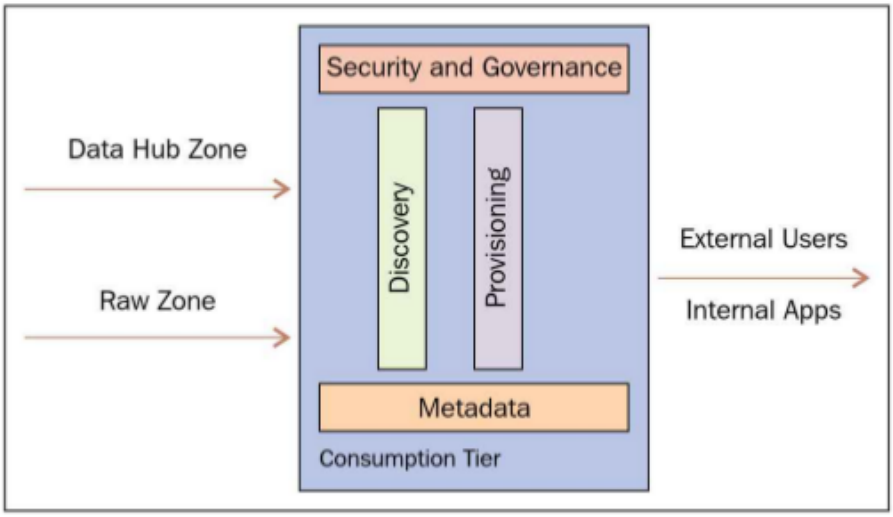


Figura 4.12: Estructura del Nivel de Obtención

4.1.3. Comparativa Data Lakes vs Data Warehouses

El concepto de Data Lake es utilizado, en algunas ocasiones, de manera equivocada como sustitutivo de los tradicionales Data Warehouses. Sin embargo existen grandes diferencias entre estas arquitecturas, las cuales radican principalmente en el tipo de datos que son capaces de almacenar, así como en el momento en que estos datos son formateados. Un Data Warehouse es una base de datos corporativa tradicional, la cual destaca por integrar y depurar la información antes de su almacenamiento, siguiendo procesos ETL, facilitando así su análisis desde un gran número de perspectivas a gran velocidad. Este hecho constituye una de las principales diferencias existentes entre los Data Lakes y los Data Warehouses, ya que de esta forma en los Data Warehouses los datos son almacenados con una estructura previa definida, mientras que los Data Lakes los datos son almacenados sin ser formateados, los datos se formatean únicamente en el momento en que son usados. En la figura 4.13 se puede ver el flujo que siguen los datos en los Data Warehouses.

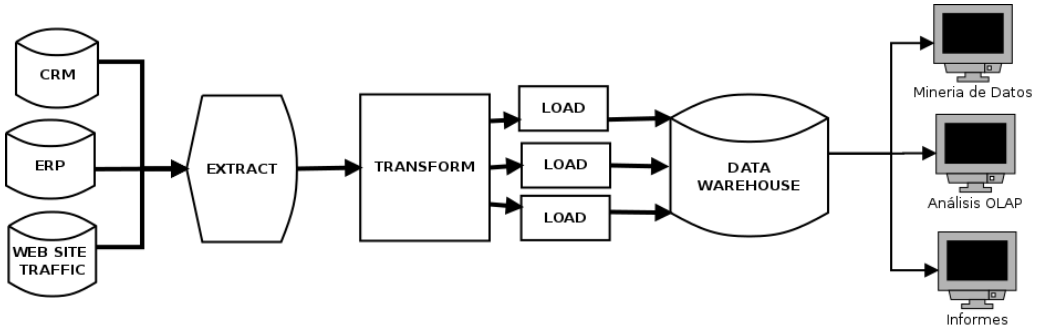


Figura 4.13: Flujo a seguir por los datos en un Data Warehouse, ETL

Otra gran diferencia existente entre estas arquitecturas, como previamente se ha comentado, radica en el tipo de datos que son capaces de almacenar. Este aspecto se encuentra fuertemente relacionado con la anterior diferencia, ya que el proceso ETL seguido en los Data Warehouse limita en gran parte los datos a almacenar, siendo almacenados de una única fuente y dotados de una estructura, perdiendo así gran cantidad de información. Frente a esto, los Data Lakes siguen un *scheme on read*, pudiendo así almacenar cualquier tipo de datos. Este hecho constituye una gran ventaja de este tipo de arquitecturas, ya que en la actualidad el 80 % de los datos que se recogen son no estructurados o semi-estructurados, por lo que no podrían ser almacenado en un Data Warehouse sin un procesamiento previo, el cual aumentaría los costes, reduciría la velocidad de carga de los datos e incluso podría ocasionar la perdida de gran cantidad de información.

Otra de las principales diferencias que se pueden apreciar entre estas arquitecturas radica en las preguntas a realizar, es decir, si se conoce el valor de los datos previamente a su almacenamiento, como en los Data Warehouse, o si no se conoce y es necesario su estudio, como en los Data Lakes.

En cuanto al precio asociado al almacenamiento de estas arquitecturas cabe destacar el bajo coste de los Data Lakes, para los que es posible conseguir grandes capacidades de almacenamiento a bajo coste, gracias principalmente a soluciones open-source, como Hadoop, permitiendo un gran ahorro en licencias. En la tabla 4.1 se pueden ver algunas de las principales diferencias existentes entre estas arquitecturas.

Data Warehouse	vs	Data Lakes
Estructurados, procesados	DATOS	Estructurados, semi-estructurados y no estructurados
Orientado a escritura, scheme on write	PROCESAMIENTO	Orientado a lectura, scheme on read
Es caro almacenar grandes volúmenes de datos	ALMACENAMIENTO	Grandes capacidades de almacenamiento a bajo coste
Normalmente una	FUENTES	Múltiples fuentes
Estructura en estrella y en copo de nieve	ESTRUCTURAS OPTIMIZADAS	No
Madura	SEGURIDAD	Madurando (se encuentra en desarrollo)
Profesionales de los negocios	USUARIOS	Científicos de datos

Tabla 4.1: Diferencias entre Data Lakes y Data Warehouse

En definitiva los Data Lakes no surgen como una arquitectura sustitutiva de los Data Warehouse, ya que como se ha podido observar pueden trabajar de manera conjunta, permitiendo a las empresas conseguir grandes mejoras en la gestión, almacenamiento y tratamiento de los enormes volúmenes de datos que son capaces de captar en la actualidad.

Capítulo 5

Modelo de Datos

En este capítulo se persigue realizar un seguimiento completo de la información desde su extracción hasta su posterior carga en una aplicación final. Para ello se analizarán y describirán en detalle las diversas fuentes de información a partir de las cuales son obtenidos los datos que se utilizarán a lo largo de este proyecto *Raw Data*. Una vez analizadas esas fuentes de información se desarrollará el modelo conceptual, con el cual se pretende representar la realidad en función de los datos previamente analizados. Por último se llevará a cabo un análisis de los diferentes cambios que se realizan sobre los datos con el fin de facilitar su uso en las aplicaciones finales que los utilicen, utilizando para ello un Mapa de Datos, con el fin de mantener un historial de cambios sobre los mismos.

5.1. Raw Data

En esta sección se persigue describir las diversas fuentes de información, en su formato original (*Raw Data*), de las cuales se van a obtener datos que son susceptibles de ser utilizados para dotar de contenido a la arquitectura a desarrollar. Una vez analizada la información de la que se dispone se seleccionarán los campos cuya información será utilizada en este proyecto, justificando su elección.

Las fuentes de información gracias a las que es posible dotar de contenido a dicha arquitectura pueden ser divididas en información asociada a los navíos e información portuaria.

5.1.1. Información asociada a los navíos

La información asociada a los navíos es obtenida a través de AISHub [2], el cual es un *AIS Data Exchange* cuyo objetivo es almacenar *Raw NMEA Data* y dotarla de valor. Esta empresa facilita la obtención de mensajes AIS utilizando el sistema AIS costero, es decir, no gestionan datos vía satélite. La elección de esta fuente de información se fundamenta en su precio, al ser gratuita siempre y cuando se colabore con ella para dotarla a su vez de información. Para este proyecto se consiguió una licencia gratuita de un mes, al ser utilizada la información con fines académicos. Por tanto los datos obtenidos inicialmente se encuentran acotados a los mensajes en tiempo real captados por AISHub entre los días 15 de septiembre y 15 de octubre de 2017.

Cabe destacar que para captar la información ofertada por AISHub, a través de su API, ha sido necesario desarrollar una pequeña aplicación Java. Con el fin de captar esta información se ha tenido que dar a AISHub la dirección IP del primer servidor utilizado en el proyecto, dedicado

a la captación y almacenamiento de dicha información. Una vez obtenida la licencia de AISHub e indicada la IP de la máquina encargada de hacer peticiones a su API es posible desplegar el programa de descargas de datos. A groso modo, ya que en el capítulo "Implementación" se entrará en más detalle, este programa se conecta a la API de AISHub a través de una url creada usando diferentes parámetros, como el nombre de usuario de la licencia, y en cada conexión se descargará un fichero, bit a bit, con el formato elegido en la url. Este programa se ejecutará en segundo plano en el servidor, haciendo una nueva petición cada 90 segundos, con el fin de descargar automáticamente los ficheros. La realización de la petición se lleva a cabo dentro de un hilo y se realiza en ese espacio de tiempo al ser el mínimo recomendado por la propia API para descargar información, ya que en caso de realizar un mayor número de conexiones puede llevar a problemas a la API y no retornar ninguna información.

Como se ha comentado en el capítulo 2 los mensajes emitidos por el sistema AIS se envían utilizando el protocolo NMEA-0183, altamente utilizado para la comunicación entre los diversos componentes del ecosistema marítimo. La información encapsulada en cada mensaje puede dividirse en 3 tipos: estática, dinámica o sobre el viaje. Cabe destacar que toda esta información no es emitida en un único mensaje, es decir, cada mensaje final AIS obtenido ha sido conseguido a través de la unión de los 3 tipos de mensajes anteriormente mencionados. Por ejemplo los de la información dinámica suele ser emitida con una gran frecuencia, entre 2 y 10 segundos de diferencia entre cada mensaje, mientras que la información estática suele enviarse cada 3 minutos.

Gracias a la API que proporciona AISHub es posible acceder a gran cantidad de información sobre los datos capturados, que dividiremos en datos dinámicos, relativos al viaje y estáticos.

Datos dinámicos. En la tabla 5.1 se podrán ver los diferentes campos que componen la información dinámica del navío y que son susceptibles de ser utilizados.

Campo	Long	Nombre	Miembro	Tipo	Explicación
38-41	4	Estado de Navegación	Estado	Enum	Campo encargado de describir el estado actual del navío
42-49	8	Velocidad de giro (ROT)	Giro	Float	Campo encargado de medir la velocidad de giro, en los datos obtenidos no se almacena
50-59	10	Velocidad sobre el fondo (SOG)	Velocidad	Float	Campo encargado de mostrar la velocidad sobre el fondo del navío, la cual varía desde de 0 a 102 nudos. El valor 1022 indica que el navío navega más rápido de los 102 nudos y el valor 1023 indica que la velocidad no esta disponible
60-60	1	Precisión de la posición	Precisión	Boolean	Campo al cual no tenemos acceso en los datos que actualmente se han captado
61-88	28	Longitud	Longitud	Float	Campo encargado de describir la longitud en la que se posiciona el navío. Minutes/10000
89-115	27	Latitud	Latitud	Float	Campo encargado de describir la latitud en la que se posiciona el navío. Minutes/10000
116-127	12	Dirección sobre el fondo (COG)	Dirección	Float	Campo encargado de mostrar el rumbo sobre el fondo del navío. El valor 3600 indica que no hay información disponible sobre dicho campo.
128-136	9	Rumbo Real	Rumbo	Unsigned Integer	Campo encargado de describir el rumbo del navío. Esta información varía entre los 0 y los 359 grados, si el valor de dicho campo es de 511 indica que no está disponible
137-142	6	TimeStamp	Segundos	Unsigned Integer	Momento en el que se envía el mensaje, en formato UTC
143-144	2	Indicador de maniobra	Maniobra	Enum	Campo al cual no tenemos acceso en los datos que actualmente se han captado
145-147	3	Repuestos		Bit reservado	Campo al cual no tenemos acceso en los datos que actualmente se han captado
148-148	1	RAIM flag	RAIM	Boolean	Campo al cual no tenemos acceso en los datos que actualmente se han captado
149-167	19	Radio Status	radio	Unsigned Integer	Campo al cual no tenemos acceso en los datos que actualmente se han captado

Tabla 5.1: Datos dinámicos ofrecidos por AISHub

El campo "Estado de Navegación" es utilizado con el fin de indicar el estado del navío en cada momento. Los diferentes códigos que pueden ser utilizados en este campo se encuentran descritos en la tabla 5.2.

Código	Descripción
0	Under way using engine
1	At anchor
2	Not under command
3	Restricted manoeuverability
4	Constrained by her draught
5	Moored
6	Aground
7	Engaged in Fishing
8	Under way sailing
9	Reserved for future amendment of Navigational Status for HSC
10	Reserved for future amendment of Navigational Status for HSC
11	Reserved for future use
12	Reserved for future use
13	Reserved for future use
14	AIS-SART is active
15	Not defined (default)

Tabla 5.2: Estado de navegación

Una vez analizados los mensajes que se han podido captar a través de esta plataforma se ha decidido utilizar ciertos campos en el desarrollo de este proyecto, en la tabla 5.3 se describen los diversos campos, asociados a la información dinámica de los navíos, que se ha decidido utilizar en el proyecto, así como una pequeña justificación de su elección.

ID	Nombre	Justificación
C1	Estado de navegación	Este campo puede aportar información relevante sobre el estado del navío en el momento de emisión
C2	Velocidad sobre el fondo (SOG)	Este campo aporta información relevante sobre la trayectoria que realiza el navío
C3	Longitud	Este campo aporta una informacion imprescindible para así poder localizar al navío en todo momento
C4	Latitud	Este campo aporta una informacion imprescindible para así poder localizar al navío en todo momento
C5	Dirección sobre el fondo (COG)	Este campo aporta información relevante sobre la trayectoria que realiza el navío
C6	Rumbo (Heading)	Este campo aporta información relevante sobre la trayectoria que realiza el navío
C7	TimeStamp	Este campo aporta información de gran importancia, ya que permite asociar a cada navío con el momento en el que envía cada mensaje

Tabla 5.3: Campos asociados a los datos dinámicos seleccionados

Datos estáticos. En la tabla 5.4 se muestran los diferentes campos que componen la información estática del navío y que son susceptibles de ser utilizados.

Campo	Lon	Nombre	Miembro	Tipo	Codificación
0-5	6	Tipo de mensaje	Tipo	Unsigned Integer	Constante: 5
6-7	2	Indicador de repetición	repetición	Unsigned Integer	Contador de mensajes repetidos
8-37	30	MMSI	mmsi	Unsigned Integer	9 Dígitos
38-39	2	Versión AIS	ais_version	Unsigned Integer	0=[ITU1371], 1-3 = future editions
40-69	30	Número IMO	imo	Unsigned Integer	Identificador de la organización marítima internacional
70-111	42	Call Sign	callsign	String	7 six-bit characters (es usado para identificar el navío que emite)
112-231	120	Nombre del navío	nombre	String	Caracteres de 20 six-bit
232-239	8	Tipo de navío	shiptype	Enum	Mirar la tabla *Códigos por navío*
240-248	9	Dimensin de proa	A	Unsigned Integer	Metros
249-257	9	Dimensión de popa	B	Unsigned Integer	Metros
258-263	6	Dimensión de babor	C	Unsigned Integer	Metros
264-269	6	Dimensión de estribor	D	Unsigned Integer	Metros
294-301	8	Calado (Draught)	Draught	Float	Metros/10

Tabla 5.4: Datos estáticos

Al igual que ocurre en el caso del campo "*Estado de Navegación*", previamente explicado, en esta información nos encontramos con otro atributo de tipo enumerado, "*Tipo de Navío*". La tabla 5.5 muestra los diferentes tipos de navío que se identifican mediante el sistema AIS.

Code	Ship & Cargo Classification
0	Not available (default)
1-19	Reserved for future use
20	Wing in ground (WIG) all ships of this type
21	Wing in ground (WIG) Hazardous category A
22	Wing in ground (WIG) Hazardous category B
23	Wing in ground (WIG) Hazardous category C
24	Wing in ground (WIG) Hazardous category D
25	Wing in ground (WIG) Reserved for future use
26	Wing in ground (WIG) Reserved for future use
27	Wing in ground (WIG) Reserved for future use
28	Wing in ground (WIG) Reserved for future use
29	Wing in ground (WIG) Reserved for future use
30	Fishing
31	Towing
32	Towing: length exceeds 200m or breadth exceeds 25m
33	Dredging or underwater ops
34	Diving ops
35	Military ops
36	Sailing
37	Pleasure Craft
38	Reserved
39	Reserved
40	High speed craft (HSC) all ships of this type
41	High speed craft (HSC) Hazardous category A
42	High speed craft (HSC) Hazardous category B
43	High speed craft (HSC) Hazardous category C
44	High speed craft (HSC) Hazardous category D
45	High speed craft (HSC) Reserved for future use
46	High speed craft (HSC) Reserved for future use
47	High speed craft (HSC) Reserved for future use
48	High speed craft (HSC) Reserved for future use
49	High speed craft (HSC) No additional information
50	Pilot Vessel
51	Search and Rescue vessel
52	Tug
53	Port Tender
54	Anti-pollution equipment
55	Law Enforcement
56	Spare - Local Vessel
57	Spare - Local Vessel
58	Medical Transport
59	Noncombatant ship according to RR Resolution No. 18

Code	Ship & Cargo Classification
60	Passenger all ships of this type
61	Passenger Hazardous category A
62	Passenger Hazardous category B
63	Passenger Hazardous category C
64	Passenger Hazardous category D
65	Passenger Reserved for future use
66	Passenger Reserved for future use
67	Passenger Reserved for future use
68	Passenger Reserved for future use
69	Passenger No additional information
70	Cargo all ships of this type
71	Cargo Hazardous category A
72	Cargo Hazardous category B
73	Cargo Hazardous category C
74	Cargo Hazardous category D
75	Cargo Reserved for future use
76	Cargo Reserved for future use
77	Cargo Reserved for future use
78	Cargo Reserved for future use
79	Cargo No additional information
80	Tanker all ships of this type
81	Tanker Hazardous category A
82	Tanker Hazardous category B
83	Tanker Hazardous category C
84	Tanker Hazardous category D
85	Tanker Reserved for future use
86	Tanker Reserved for future use
87	Tanker Reserved for future use
88	Tanker Reserved for future use
89	Tanker No additional information
90	Other Type all ships of this type
91	Other Type Hazardous category A
92	Other Type Hazardous category B
93	Other Type Hazardous category C
94	Other Type Hazardous category D
95	Other Type Reserved for future use
96	Other Type Reserved for future use
97	Other Type Reserved for future use
98	Other Type Reserved for future use
99	Other Type no additional information

Tabla 5.5: Códigos por barco

Una vez analizados los mensajes que se han podido captar a través de esta plataforma se ha decidido descartar ciertos campos para el desarrollo de este proyecto. En la tabla 5.6 se describen los diversos atributos, asociados a la información estática de los navíos, que se ha decidido utilizar en el proyecto, así como una breve explicación de los mismos y del por qué de su elección.

ID	Nombre	Descripción	Justificación
C8	MMSI	Número de identificación del servicio móvil marítimo, pudiendo ser este un navío (actuando como transpondedor) o una estación costera. Se encuentra formado por 9 dígitos y es obligatorio en todos los navíos. Los 3 primeros dígitos de cada MMSI permiten identificar el país de bandera del navío	Se ha decidido almacenar este campo al permitir identificar al navío de manera inequívoca a gran velocidad, así como facilitar la obtención de su nacionalidad o comprobar si se corresponde con una aeronave SAR, *Search and Rescue*
C9	IMO	Atributo encargado de comunicar si el navío se encuentra adscrito a la Organización Marítima Internacional (IMO). Si el navío es de navegación interior el valor de este campo se reducirá a 0	Se ha decidido almacenar esta información ya que puede ser de ayuda para ver qué tipo de barcos suelen estar adscritos a la IMO
C10	Callsign	Este atributo consiste en un conjunto de 7 caracteres (de 6 bits) que emiten los navíos y permite identificarlos de manera única. Al igual que el MMSI permite identificar el país de bandera de cada navío gracias a su prefijo.	Se ha decidido almacenar este campo al permitir identificar al navío de manera inequívoca a gran velocidad (en algunos casos puede que no haya información sobre el mismo)
C11	Nombre	Nombre que persigue identificar a cada navío de manera menos técnica, puede darse el caso de que diferentes navíos tengan el mismo nombre, o algún nombre similar	Se ha decidido almacenar esta información al facilitar a los usuarios la identificación de cada navío
C12	Tipo de navío	Este atributo permite a los usuarios identificar el tipo de navío que emite cada mensaje. Para identificarlo se cuenta con un enumerado previamente explicado	Se ha decidido almacenar esta información al ser de ayuda en el posible caso de desear saber que transporta cada barco, permitiendo acotar el cerco en una primera versión del proyecto
C13	Dimensión de proa	Atributo encargado de aportar informacion sobre la proa del navío, parte delantera en que se unen las amuras de un barco formando el canto o roda que al avanzar va cortando las aguas en que navega. El valor 511 indica 511 metros o más	Se ha decidido almacenar esta información al ser de ayuda para,conocer,en los puertos en los que es capaz de atracar cada navío, ya que,en función de sus dimensiones es posible que pueda dañarse al atracar,en puerto
C14	Dimensión de popa	Atributo encargado de aportar informacion sobre la popa del navío, terminación posterior de la estructura de un barco. El valor 511 indica 511 metros o más	Se ha decidido almacenar esta información al ser de ayuda para conocer,en los puertos en los que es capaz de atracar cada navío, ya que en función de sus dimensiones es posible que pueda dañarse al atracar en puerto
C15	Dimensión de babor	Atributo encargado de aportar informacion sobre las dimensiones de babor asociadas al navío, parte izquierda del navío mirando hacia proa. El valor 63 indica que la proa es de 63 metros o más	Se ha decidido almacenar esta información al ser de ayuda para,conocer,en los puertos en los que es capaz de atracar cada navío, ya que,en función de sus dimensiones es posible que pueda dañarse al atracar,en puerto
C16	Dimensión de estribor	Atributo encargado de aportar informacion sobre las dimensiones de estribor asociadas al navío, parte derecha del navío mirando hacia proa. El valor 63 indica que la proa es de 63 metros o más	Se ha decidido almacenar esta información al ser de ayuda,para,conocer,en los puertos en los que es capaz de atracar cada navío„ya que,en función de sus dimensiones es posible que pueda dañarse al,atracar,en puerto
C17	Calado (Draugth)	Atributo encargado de describir la distancia vertical entre un punto de la linea de flotación y la base del navío, incluido el casco	Se ha decidido almacenar esta información al ser de,ayuda,para,conocer,en los puertos en los que es capaz de atracar cada,navío„ya que,en función de sus dimensiones es posible que pueda dañarse,al,atracar,en puerto

Tabla 5.6: Campos asociados a los datos estáticos seleccionados

Datos relativos al viaje. En la tabla 5.7 se muestran los diferentes campos que componen la información del viaje asociada a este tipo de mensajes.

Campo	Lon	Descripción	Miembro	Tipo	Codificación
270-273	4	Posición fija	epfd	Enum	Campo no usado
274-277	4	ETA mes (UTC)	mes	Unsigned Integer	1-12, 0=N/A (default)
278-282	5	ETA días (UTC)	días	Unsigned Integer	1-31, 0=N/A (default)
283-287	5	ETA horas (UTC)	horas	Unsigned Integer	0-23, 24=N/A (default)
288-293	6	ETA minutos (UTC)	minutos	Unsigned Integer	0-59, 60=N/A (default)
302-421	120	Destino	Destino	String	20 6-bit characters (ISOCODE)
422-422	1	DTE	dte	Boolean	0=Data terminal ready, 1=Not ready(default).
423-423	1	Spare		Bits reservados	No usa codificación

Tabla 5.7: Datos del viaje

Una vez analizados los mensajes que se han podido captar a través de esta plataforma se ha decidido descartar ciertos campos para el desarrollo de este proyecto. En la tabla 5.8 se describen los diversos atributos, asociados a la información estática de los navíos, que se han decidido utilizar en el proyecto, así como una breve explicación de los mismos y del por qué de su elección.

ID	Nombre	Descripción	Justificación
C18	Destino	Mediante este atributo es posible conocer el puerto de destino al que se dirige cada navío	Se ha decidido almacenar esta información al considerar que aporta información descriptiva de la ruta a seguir por el navío, pudiendo incluso realizar trazas de los puertos en los que para dicho navío
C19	ETA	Mediante este atributo se permite conocer el momento en el que se espera que el navío llegue a puerto. Este atributo se forma gracias a la unión de los 4 atributos ETA vistos en la tabla 5.7	Se ha considerado que puede aportar información de gran interés tanto para navío como para el puerto en el que se espera que atraque.

Tabla 5.8: Campos asociados a los datos relativos al viaje seleccionados

Todos los datos explicados anteriormente que no pertenezcan a los campos seleccionados, los mostrados en las tablas con el identificador CXX, no se utilizarán en este proyecto al exceder el alcance del mismo. En muchos casos la información descartada no se encuentra disponible a través de la plataforma de intercambio de datos utilizado, al ser gratuito. Es posible acceder a un mayor conjunto de información, así como a datos vía satélite, a través de otras plataformas, como las descritas en el capítulo “Estado del Arte”.

5.1.2. Información asociada a los puertos

La información asociada a los puertos, a nivel mundial, es obtenida a través de una base de datos *MS Access* ofertada por *National Geospatial Intelligence Agency (NGA)*[20]. Al igual que en el caso anterior, esta información lleva consigo una serie de limitaciones, siendo la principal la fecha, ya que son datos del año 2014. Pese a estas limitaciones se consideró que al ser una versión inicial podría ser interesante analizar esta información, con vistas a que en un futuro sea viable adquirir alguna licencia de plataformas como Fleetmon o MarineTraffic, las cuales, además de información sobre navíos también son capaces de proveer de información sobre puertos. La elección de esta fuente de información radica en la gran variedad de datos disponibles sobre los puertos, los cuales se explican en la tabla 5.9.

ID	Nombre	Descripción	Tipo
P1	Index	El listado de puertos se encuentra ordenado de manera consecutiva a través de esta columna, facilitando así la busqueda de estos	Int
P2	Zone_Code	Este campo, unido al campo Country_Code, permite identificar el puerto. De esta manera los puertos se listan secuencialmente en función de su situación geográfica	Int
P3	Country_Code	Este campo, unido al campo Zone_Code,„permite identificar el puerto.,De esta manera los puertos se listan secuencialmente en función de su,situación geográfica	Int
P4	Port_Name	Este campo permite la identificación de un puerto de manera coloquial	String
P5	Latitude	Este campo facilita la localización de un puerto, junto con la longitud, al describir la latitud en la que se encuentra el puerto. Su obtención divide la latitud en 3 columnas, los cuales deben unirse para evitar problemas de localización	Int
P6	Longitude	Este campo facilita la localización de un puerto, junto con la latitud„al describir la longitud en la que se encuentra el puerto. Su obtención,divide la latitud en 3 columnas, los cuales deben unirse para evitar,problemas de localización	Int
P7	Sailing_Directions	Esta información permite conocer el número de publicación de las instrucciones de navegación de la NGA, en el que se describe el puerto o la zona donde esta situado. En algunos puertos se pueden mostrar publicaciones con diferentes abreviaturas	(Enum o String?)
P8	Chart	Este campo permite conocer el número de la carta de mejor escala emitida por la NGA	String
P9	Harbor_Size	Mediante esta información se trata de conocer el tamaño del puerto, para lo que se miden el área, las instalaciones y el espacio del muelle	Enum
P10	Type_Harbor	Mediante esta información es posible determinar el tipo de puerto del que se dispone información	Enum
P11	Sheltered_Afforded	Este campo permite conocer la calidad del área de fondeo de cada puerto, zona en la que se suelen realizar las operaciones portuarias normales	Enum
P12	Entrance_Restrictions	Información relativa a los factores naturales que pueden ocasionar problemas en la entrada de navíos a puerto, como el hielo. Esta información se encuentra distribuida entre varios campos (tide, swell, ice, other)	Cada campo es de tipo boolean
P13	Overhead_limitations	Campo encargado de determinar si existen cables aéreos o de puente	Boolean? hay campos vacíos
P14	Depths	Información asociada a la profundidad de diferentes elementos del puerto, como son el canal principal o el ancho principal, entre otros. Esta información suele ir dada en 5 pies y se divide en varios campos	Por determinar
P15	Tides	(No lo entiendo bien)	Int
P16	Max_Size_Vessel	Tamaño máximo del navío que puede entrar en puerto	Enum
P17	Good_holding_ground	Campo encargado de notificar si se han comunicado las condiciones reales de anclaje	Boolean (huecos en blanco)
P18	Turning_area	Campo encargado de notificar si en el puerto existe un área de giro para los navíos	Boolean(huecos en blanco)

ID	Nombre	Descripción	Tipo
P19	First_Port_of_Entry	Campo encargado de indicar si en el puerto es posible gestionar mercancías y personal extranjero a través de Aduanas e Inmigración	Boolean(huecos en blanco)
P20	US_representative	Campo encargado de indicar si USA tiene representación civil o militar en ese puerto	Boolean(huecos en blanco)
P21	ETA	Campo encargado de indicar si se necesita un mensaje ETA para el puerto	Boolean(huecos en blanco)
P22	Pilotage	Información asociada a la capacidad de asistencia a los navíos desde puerto. Esta información se encuentra dividida en 4 campos, Compulsory, available, local assist y advisable	Boolean(huecos en blanco)
P23	Tugs	Información dedicada a indicar si hay remolcadores disponibles para la asistencia en el atraque o en el anclaje. Esta información se divide en 2 campos	Boolean (huecos en blanco)
P24	Quarantine	No se dispone de información que explique la division de esta en campos	Boolean (huecos en blanco)
P25	Communications	Información que permite conocer los diferentes tipos de comunicaciones que se encuentran disponibles en el puerto y/o en un área cercana. Esta información da lugar a 6 campos	Boolean (huecos en blanco)
P26	Load/Offload	Información referente a la zona donde se llevan a cabo las operaciones portuarias. Esta información se divide en diversos campos, como los campos anchor o Med Moor	Boolean (huecos en blanco)
P27	Medical Facilities	Campo encargado de indicar si el puerto dispone de instalaciones médicas para ayudar a los marineros	Boolean (huecos en blanco)
P28	Garbage Disposal	Campo encargado de indicar si la basura se puede desechar en el muelle o en el fondeadero	Boolean (huecos en blanco)
P29	Degaussing	Indica si esta disponible la desmagnetización	Boolean(huecos en blanco)
P30	Dirty_Ballast	Campo encargado de indicar si el puerto cuenta con instalaciones suficiones para recibir productos contaminados químicamente	Boolean (huecos en blanco)
P31	Cranes/Lifts	Indica si hay grúas disponibles, así como su tipo y su capacidad de elevación. Esta información da lugar a varios campos en donde es detallada	Boolean (huecos en blanco)
P32	Services	Información encargada de indicar si los servicios portuarios normales se encuentran disponibles. Esta información se encuentra descrita en diversos campos	Boolean (huecos en blanco)
P33	Supplies	Información encargada de enumerar la disponibilidad de provisiones, agua y combustible, enumerando por separado el gasoleo y el fuel. Esta información se encuentra descrita en diversos campos	Boolean (huecos en blanco)
P34	Repairs	Campo encargado de clasificar las reparaciones que se pueden llevar a cabo en los buques	Enum (huecos en blanco)
P35	Drydock/Marine Railway	Campo encargado de clasificar el tamaño y tipo de instalaciones de reparación subacuática del puerto	Enum(huecos en blanco)

Tabla 5.9: Explicación a alto nivel sobre datos portuarios

Como se observa en la tabla 5.9 existen varios bloques de información que dan lugar a un gran número de campos. Con el fin de evitar crear una tabla totalmente ilegible, por el gran número de filas, se ha optado por describir estos campos dividiéndolos por su bloque de información. A continuación explica cada uno de estos campos.

Campos asociados al bloque "*Entrance Restrictions*". Este bloque de información trata de gestionar la información relativa a los diversos factores naturales que pueden ocasionar problemas en la entrada de navíos a puerto. Esta información se encuentra dividida en varios campos, los cuales se encuentran descritos en la tabla 5.10.

ID	Nombre	Descripción	Tipo
P12.1	Tide	Este campo persigue indicar si hay o no marea en el momento de entrada a puerto	Boolean
P12.2	Swell	Este campo persigue indicar si hay o no oleaje en el momento de entrada a puerto	Boolean
P12.3	Ice	Este campo persigue indicar si hay o no hielo en el momento de entrada a puerto	Boolean
P12.4	Other	Este campo persigue indicar si existen problemas para la entrada a puerto de navíos por culpa de factores naturales distintos a los previamente descritos	Boolean

Tabla 5.10: Campos derivados del bloque Entrance Restrictions

Campos asociados al bloque "*Depths*". Los datos obtenidos ofrecen información sobre la profundidad de distintos elementos del navío. En el fichero original esta información se encuentra divida en varios campos, los cuales se encuentran descritos en la tabla 5.11.

ID	Nombre	Descripción	Tipo
P14.1	Channel	Campo encargado de clasificar la profundidad de control del canal principal (o más profundo). El canal seleccionado debe llevar al fondeadero o hasta el muelle	Enum
P14.2	Anchorage	Este campo persigue clasificar la profundidad del aclaje principal, en un punto de bajío aislado	Enum
P14.3	Cargo Pier	Este campo persigue clasificar la profundidad máxima del muelle de mayor profundidad del puerto, ingresándola en el Índice Mundial de Puertos	Enum
P14.4	Oil Termina	Este campo persigue clasificar la profundidad máxima de la instalación diseñada para la carga o descarga de productos asociados al petróleo asociada a puerto más profunda.	Enum

Tabla 5.11: Campos derivados del bloque Depths

Para más información sobre los valores que toma como referencia para realizar las clasificaciones descritas en la tabla 5.11 se puede visualizar en un documento realizado por la NGA[21].

Campos asociados al bloque "*Pilotage*". Los datos obtenidos ofrecen información sobre la capacidad de asistencia a los navíos desde puerto. En el fichero original esta información se encuentra divida en varios campos, los cuales se encuentran descritos en la tabla 5.12.

ID	Nombre	Descripción	Tipo
P22.1	Compulsory	Este campo indica si es obligatoria la asistencia desde puerto	Boolean
P22.2	Available	Este campo se encarga de indicar si se encuentra disponible la asistencia desde puerto	Boolean
P22.3	Local Assist	Este campo se encarga de indicar si existe asistencia local desde puerto	Boolean
P22.4	Advisable	Este campo se encarga de indicar si es aconsejable la asistencia desde puerto	Boolean

Tabla 5.12: Campos derivados del bloque Pilotage

Campos asociados al bloque "*Communications*". Los datos obtenidos ofrecen información sobre las diferentes formas de comunicación que puede utilizar el puerto. En el fichero original esta información se encuentra divida en varios campos, los cuales se encuentran descritos en la tabla 5.13.

ID	Nombre	Descripción	Tipo
P25.1	Telephone	Este campo indica si el puerto es capaz de comunicarse vía telefónica	Boolean
P25.2	Telefax	Este campo indica si el puerto es capaz de comunicarse mediante el uso de un fax	Boolean
P25.3	Radio	Este campo indica si el puerto es capaz de comunicarse vía radio	Boolean
P25.4	Radio Tel	Este campo indica si el puerto es capaz de comunicarse vía radio teléfono	Boolean
P25.5	Air	Este campo indica si el puerto posee equipos de comunicaciones que les permita comunicarse con aeronaves	Boolean
P25.6	Rail	No se ha encontrado información sobre este tipo de comunicación	Boolean

Tabla 5.13: Campos derivados del bloque Communications

Campos asociados al bloque "*Load/Offload*". Los datos obtenidos ofrecen información sobre zona en la que se llevan a cabo las operaciones portuarias. En el fichero original esta información se encuentra divida en varios campos, los cuales se encuentran descritos en la tabla 5.14.

ID	Nombre	Descripción	Tipo
P26.1	Wharves	Este campo indica si las operaciones portuarias se realizan en muelle	Boolean
P26.2	Anchor	Este campo indica si las operaciones portuarias se realizan en el momento de anclar el navío	Boolean
P26.3	Med Moor	Este campo indica si las operaciones portuarias se realizan en un amarre medio	Boolean
P26.4	Beach Moor	Este campo indica si las operaciones portuarias se realizan en el amarre en una playa	Boolean
P26.5	Ice Moor	Este campo indica si las operaciones portuarias se realizan en el amarre en hielo	Boolean

Tabla 5.14: Campos derivados del bloque Load/Offload

Campos asociados al bloque "*Cranes/Lifts*". Los datos obtenidos ofrecen información sobre las grúas disponibles, así como su capacidad de elevación y los tipos de grúas existentes. En el fichero original esta información se encuentra divida en varios campos, los cuales se encuentran descritos en la tabla 5.15 y 5.16.

ID	Nombre	Descripción	Tipo
P31.1	Fixed	Este campo indica si existen grúas fijas disponibles	Boolean
P31.2	Mobile	Este campo indica si existen grúas móviles disponibles	Boolean
P31.3	Floating	Este campo indica si existen grúas flotantes disponibles	Boolean

Tabla 5.15: Campos derivados del bloque Crane

ID	Nombre	Descripción	Tipo
P31.4	100 tons plus	Este campo indica si en el puerto hay gruas con fuerza para elevar cargas de mas de 100 toneladas	Boolean
P31.5	50-100 tons	Este campo indica si en el puerto hay gruas con fuerza para elevar cargas de entre 50 y 100 toneladas	Boolean
P31.6	25-49 tons	Este campo indica si en el puerto hay gruas con fuerza para elevar cargas de entre 25 y 49 toneladas	Boolean
P31.7	0-24 tons	Este campo indica si en el puerto hay gruas con fuerza para elevar cargas de entre 0 y 24 toneladas	Boolean

Tabla 5.16: Campos derivados del bloque Lifts

Campos asociados al bloque "*Services*". Los datos obtenidos ofrecen información sobre la disponibilidad de los servicios portuarios básicos. En el fichero original esta información se encuentra divida en varios campos, los cuales se encuentran descritos en la tabla 5.17.

ID	Nombre	Descripción	Tipo
P32.1	Longshore	Desconocido	Boolean
P32.2	Elect	Este campo indica si se da servicios a barcos eléctricos	Boolean
P32.3	Steam	Este campo indica si se da servicios a barcos de vapor	Boolean
P32.4	Naving Equip	Desconocido	Boolean
P32.5	Elect Repair	Este campo indica si el puerto ofreces servicios de reparación eléctricos	Boolean

Tabla 5.17: Campos derivados del bloque Services

Campos asociados al bloque "*Supplies*". Los datos obtenidos ofrecen información sobre la disponibilidad de provisiones, agua y combustible, de diversos tipos. En el fichero original esta información se encuentra divida en varios campos, los cuales se encuentran descritos en la tabla 5.18.

ID	Nombre	Descripción	Tipo
P33.1	Provisions	Este campo indica si hay provisiones disponibles en puerto	Boolean
P33.2	Water	Este campo indica si hay agua disponible en puerto	Boolean
P33.3	Fuel oil	Este campo indica si hay gasolina en puerto	Boolean
P33.4	Diesel oil	Este campo indica si hay gasóleo en puerto	Boolean
P33.5	Deck	Se desconoce a qué hace referencia este campo	Boolean
P33.6	Engine	Se desconoce a qué hace referencia este campo	Boolean

Tabla 5.18: Campos derivados del bloque Supplies

Campos seleccionados. Una vez analizados los datos portuarios obtenidos a través de la NGA se ha decidido seleccionar ciertos campos para el desarrollo del proyecto. En la tabla 5.19 se explican los diferentes atributos seleccionados para este proyecto, así como una breve justificación de su elección.

ID	Nombre	Descripción	Justificación
C20	Index	Campo utilizado para la ordenación de los puertos en el listado	Se ha decidido seleccionar este campo al facilitar la ordenación de los puertos
C21	Zone_Code	Campo utilizado, junto con Country_Code, para identificar el puerto	Se ha decidido almacenar este campo al permitir así organizar los puertos por zona geográfica
C22	Country_Code	Campo utilizado, junto con Zone_Code, para identificar el puerto	Se ha decidido almacenar este campo al permitir así organizar los puertos por zona geográfica
C23	Port_Name	Campo utilizado para identificar el puerto	Se ha decidido seleccionar esta información al aportar información descriptiva importante a la hora de identificar el puerto
C24	Latitude	Este campo facilita la localización de un puerto, junto con la longitud, al describir la latitud en la que se encuentra el puerto.	Se ha decidido almacenar esta información ya que aporta información importante a la hora de localizar los diferentes puertos.
C25	Longitude	Este campo facilita la localización de un puerto, junto con la latitud„al describir la longitud en la que se encuentra el puerto.	Se ha decidido almacenar esta información ya que aporta información importante a la hora de localizar los diferentes puertos.
C26	Harbor_size	Mediante esta información se trata de conocer el tamaño del puerto, para lo que se miden el área, las instalaciones y el espacio del muelle	Se ha decidido almacenar esta información al permitir conocer el tamaño del puerto, pudiendo así saber que navíos pueden atracar en cada puerto
C27	Harbor_type	Mediante esta información es posible determinar el tipo de puerto del que se dispone información	Se ha decidido almacenar esta información al aportar información descriptiva de importancia sobre los puertos

Tabla 5.19: Campos asociados a los datos portuarios seleccionados para este proyecto

Los campos que no aparecen en la tabla 5.19 son descartados para el desarrollo de este proyecto al no entrar en el alcance del mismo.

5.2. Refined Data

En esta sección se lleva a cabo el diseño del Modelo Conceptual, a partir del cual se pretende describir el mini-mundo que afecta a este proyecto. Este objetivo lleva a que en esta sección se trate de representar el ámbito marítimo real de manera independiente a la implementación de la arquitectura a desarrollar. El diseño del Modelo Conceptual se encuentra soportado por documentación que incluye tanto el diagrama Entidad-Relación como el Diccionario de Datos generado. En las siguientes subsecciones se llevará a cabo una aproximación a estos dos grandes bloques del Modelado Conceptual.

5.2.1. Modelo Entidad-Relación

El Modelo Entidad-Relación persigue representar el mundo real, de una manera sencilla y libre de ambigüedades, con independencia de los aspectos físicos relacionados con la implementación de la arquitectura a desarrollar. Este modelo sigue una estructura top-down para realizar el diseño conceptual de cualquier base de datos.

Las primeras labores de análisis del problema han derivado en la creación del siguiente modelo entidad-relación, ver figura 5.1, gracias al cual se persigue mostrar de manera gráfica y sencilla el problema a solventar.

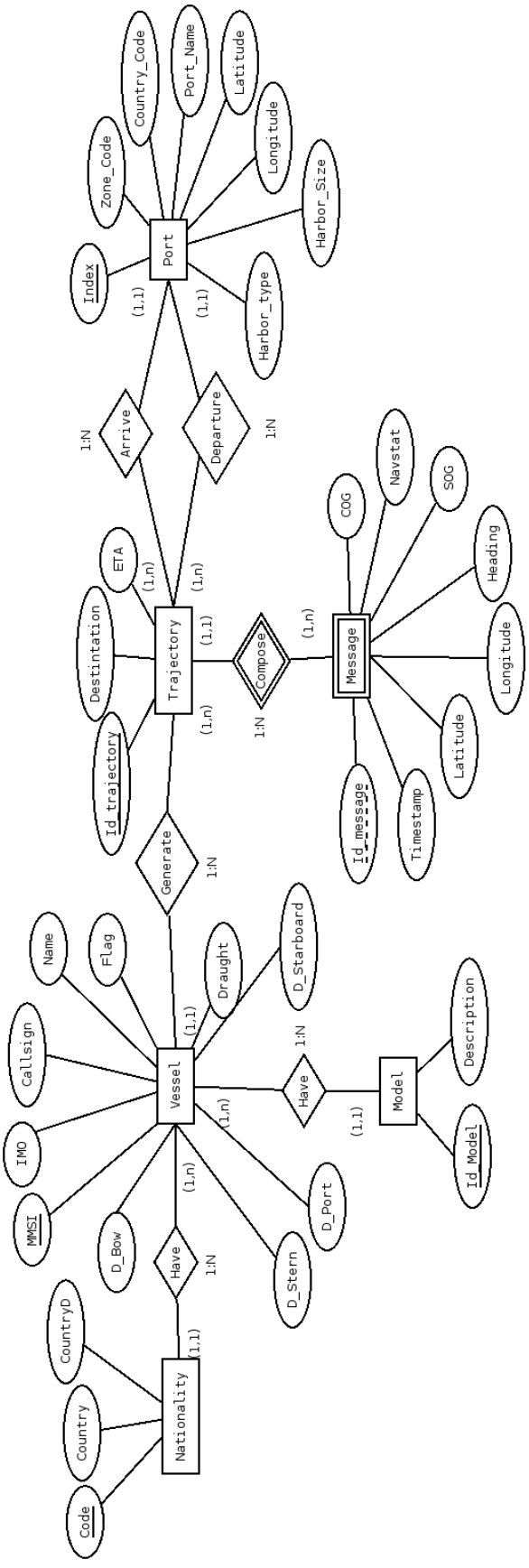


Figura 5.1: Diagrama Entidad-Relación

5.2.2. Diccionario de Datos

El Diccionario de Datos trata de proveer de una descripción detallada de los datos gestionados por la Base de Datos, los metadatos. Es imposible pensar en una base de datos, o cualquier arquitectura Big Data, sin los metadatos, ya que sin un buen uso de ellos es muy complicado caracterizar los datos y conocer su significado. A continuación se puede ver el diccionario de datos asociado a este proyecto dividido por entidades.

E1. Entidad Vessel. Entidad encargada de representar los datos que describen al navío en cuestión, correspondiéndose estos con los campos seleccionados asociados a la información estática de los mensajes anteriormente descrita. Para acceder a la información asociada a la descripción de este tipo de datos ver la tabla 5.6, en la sección Raw data.

ID	Nombre	Tipo	Restricciones	Fuente de Información	ID Fuente
E1.1	MMSI	Int	Primary key	AISHub	C8
E1.2	IMO	Int	Nullable		C9
E1.3	Callsign	String	Nullable		C10
E1.4	Nombre	String	Nullable		C11
E1.5	D_Stern	Int	Nullable		C13
E1.6	D_Bow	Int	Nullable		C14
E1.7	D_Port	Int	Nullable		C15
E1.8	D_Starboard	Int	Nullable		C16
E1.9	Flag	Int	Not Null		C8
E1.10	Draught	Float	Nullable		C17

Tabla 5.20: Entidad Vessel

E2. Entidad Model. Entidad encargada de representar los datos que describen el tipo de navío. Esta entidad permite asociar el tipo de navío, en formato entero, obtenido en los mensajes AIS con el nombre de su tipo, con el fin de almacenar así una información más descriptiva de cada navío.

ID	Tipo	Restricciones	Fuente de Información	ID Fuente
E2.1	Id_Model	Int	Primary key	C12
E2.2	Description	String	Not Null	Stakeholder

Tabla 5.21: Entidad Model

E3. Entidad Nationality. Entidad encargada de almacenar los datos relativos a la conversión de los MID code a lenguaje natural.

ID	Tipo	Restricciones	Fuente de Información	ID Fuente
E3.1	Code	Int	Primary key	C8
E3.2	Country	String	Not null	C8
E3.3	CountryD	String	Not null	C8

Tabla 5.22: Entidad Nationality

E4. Entidad Trajectory. Entidad encargada de representar las trayectorias seguidas por cada navío. Estas trayectorias se obtienen a través de los datos asociados a la información sobre el viaje, obtenida a través de los diferentes mensajes AIS y que se explica previamente en el apartado Raw data, más concretamente en la tabla 5.8.

ID	Nombre	Tipo	Restricciones	Fuente de Información	ID Fuente
E4.1	Id_Trajectory	String	Primary key	AISHub	New data
E4.2	Destination	String	Nullable		C18
E4.3	ETA	TimeStamp	Nullable		C19

Tabla 5.23: Entidad Trajectory

E5. Entidad Message. Entidad encargada de representar la información dinámica obtenida a través de los mensajes AIS previamente explicados. A partir de esta información es posible seguir las trayectorias que realiza cada navío. Es posible a la descripción detallada de los campos que la componen observando la tabla 5.3, perteneciente a la sección Raw Data.

ID	Nombre	Tipo	Restricciones	Fuente de Información	ID Fuente
E5.1	NavStat	Int	Not null	AISHub	C1
E5.2	SOG	Float	Nullable		C2
E5.3	Longitude	Float	Nullable		C3
E5.4	Latitude	Float	Nullable		C4
E5.5	COG	Float	Nullable		C5
E5.6	Heading	Int	Not null		C6
E5.7	TimeStamp	TimeStamp	Weak key, Not null		C7

Tabla 5.24: Entidad Message

E6. Entidad Port. Entidad encargada de representar la información asociada a los puertos que se gestionarán en este proyecto. Al igual que en las entidades anteriores para acceder a la descripción de cada campo se debe acceder a la sección Raw Data, más concretamente a la tabla 5.19.

ID	Nombre	Tipo	Restricciones	Fuente de Información	ID Fuente
E6.1	Index	Int	Primary key	NGA	C20
E6.2	Zone_Code	Int	Not null		C21
E6.3	Country_Code	String	Not null		C22
E6.4	Port_Name	String	Not null		C23
E6.5	Latitude	String	Not null		C24
E6.6	Longitude	String	Not null		C25
E6.7	Harbor_size	String	Nullable		C26
E6.8	Harbor_type	String	Nullable		C27

Tabla 5.25: Entidad Port

En la entidad Port existen dos campos de tipo enumerado, para conocer en detalle el valor de estos campos observar la tabla 5.25.

ID	Nombre	Valor	Equivalencia
E5.7	Harbor_size	V	Very small
		S	Small
		M	Medium
		L	Large
E5.8	Harbor_type	Cn	Coastal Natural
		Cb	Coastal Breakwater
		Ct	Coastal Tide Gate
		Rn	River Natural
		Rb	River Basin N-NONE
		Rt	River Tide Gate
		Lc	Lake or Canal
		Or	Open Roadstead
		Th	Typhoon Harbor

Tabla 5.26: Descripción de los atributos enumerados

R1. Relación Have. Esta relación se produce entre las entidades 'Model' y 'Vessel' y surge para permitir asociar cada navío con su modelo.

La cardinalidad es la siguiente: un navío solo puede ser de un modelo (1,1), mientras que un modelo puede ser el mismo para 1 o varios navíos (1,N), propagando así su clave primaria de la entidad Model a la entidad Vessel.

R2. Relación Generate. Esta relación se produce entre las entidades 'Vessel' y 'Trajectory' permitiendo asociar a cada navío con la ruta que sigue.

La cardinalidad existente en esta relación es la siguiente: un navío puede seguir 1 o varias trayectorias (1,N), mientras que una trayectoria puede ser seguida por un solo navío (1,1).

R3. Relación Compose. Esta relación se produce entre las entidades 'Message' y 'Trajectory', permitiendo asociar las rutas con la información dinámica de los navíos obtenida directamente de los mensajes.

La cardinalidad existente en esta relación es la siguiente: una trayectoria se compone de 1 o varios mensajes (1,N), mientras que un mensaje solo puede pertenecer a una trayectoria (1,1), al ser únicos por cada navío y en cada momento estos mensajes.

R4. Relación Arrive. Esta relación se produce entre las entidades 'Port' y 'Trajectory', permitiendo asociar las rutas de entrada a puerto seguidas por cada navío.

La cardinalidad existente en esta relación es la siguiente: una trayectoria solo puede llegar a un puerto (1,1), mientras que a un puerto pueden llegar una o varias (1,N).

R5. Relación Departure. Esta relación se produce entre las entidades 'Port' y 'Trajectory', permitiendo asociar las rutas de salida de puerto seguidas por cada navío.

La cardinalidad existente en esta relación es la siguiente: una trayectoria solo puede salir de un puerto (1,1), mientras que de un puerto pueden salir una o varias (1,N).

5.3. Transformación de Datos

En esta sección se persigue describir las transformaciones que afectarán a los datos y que conllevarán la transformación de estos de *Raw Data* a *Refined Data*, permitiendo así ahondar un paso más en el proceso ETL previamente descrito.

En los siguientes apartados se realizará una aproximación detallada sobre cada una de las transformaciones realizadas sobre el conjunto de datos originales de los que disponemos, siendo estas divididas por fase de limpieza.

5.3.1. Fase 0

Fase inicial de procesamiento de datos en la que se persigue, principalmente, generar los identificadores asociados a la trayectoria y los mensajes. Además de la generación de estos identificadores se persigue eliminar las comillas existentes a lo largo de todos los ficheros, así como insertar valores nulos en los registros que no dispongan de ningún valor. Esta fase inicial de limpieza solo se llevará a cabo sobre los datos procedentes de AISHub, ya que los datos portuarios tienen un mayor nivel de refinamiento. A continuación se detallan las 5 transformaciones realizadas sobre los datos en esta fase de procesamiento.

- Eliminación de comillas: En esta fase de procesamiento se eliminarán todas las comillas existentes en los ficheros de AISHub, dado que son susceptibles de ocasionar problemas.
- Inserción de valores nulos: Se insertarán valores nulos siempre que el campo carezca de valor.
- Generación del identificador de trayectoria: En esta fase se generará un identificador único para cada trayectoria que sigue cada navío. Este identificador estará formado por la concatenación entre el MMSI y la fecha del primer mensaje enviado por el navío.
- Generación del identificador de mensaje: En esta fase se generará un identificador único para cada mensaje que envíe un navío y consistirá en la concatenación entre el MMSI del mismo y el momento en el que es enviado el mensaje.

- Conversión de la fecha a formato Unix (EPOCH): Al ser necesario concatenar la fecha y el identificador del navío, para la generación de ambos identificadores, además de la posible necesidad de almacenar los valores asociados a la fecha en formato Unix se ha decidido adelantar dicha conversión a esta fase de procesamiento.

5.4. Fase 1

En esta segunda fase de procesamiento de la información se persigue realizar el alineamiento de dominio de los datos previamente procesados, llevando a cabo la adaptación de estos al modelo conceptual desarrollado en la sección 5.2.

A continuación se podrán ver en detalle las diferentes transformaciones realizadas en esta fase ordenadas en función de la fuente de información a la que pertenecen y la entidad en la que se desean almacenar.

Entidad Vessel. En una primera aproximación a los datos con los que deseamos poblar esta entidad nos encontramos con una serie de problemas derivados de los ficheros procesados en la fase anterior. A modo de ejemplo, la tabla 5.27 muestra los datos procesados en la fase 0 y que se desea insertar en dicha tabla:

MMSI	IMO	Callsign	Name	a	b	c	d	draught
367422040	0	WCX5096	BLUE HORIZON	0	0	0	0	0.0
211590710	0	DA5898	FEUERLOESCHBOOT 1	0	0	0	0	0.2
244740200	0	PC2246	MAAIKE	40	15	9	2	0.0
205593000	9569009	ORPV	WESTERSCHELDE PILOT	14	11	6	6	0.0
413500010	0	BMEI	YUAN SHUN FA	75	23	8	8	5.9
412000001	8914934	DLMU	DLMU IMTS01	100	20	19	20	0.0
367177840	7338808	WBR4464	BART TURECAMO	10	23	2	3	4.0
367479630	8103004	WDF6958	MIAMI	62	21	5	21	4.0
413901677	0	NULL	YUESHAOGUANHUO0016	28	7	4	4	0.0
244050842	0	NULL	STORM	0	0	0	0	0.0

Tabla 5.27: Raw Data Vessel

En estos datos es necesario realizar diversas transformaciones, con el fin de adecuarse al modelo conceptual previamente descrito. En esta fase las transformaciones estarán asociadas al formato de los datos con los que nos encontramos, siendo el principal problema la existencia de valores nulos en los campos “a”,“b”,“c”, “d” y “draught”, ya que los problemas asociados a los campos nulos de los atributos “Callsign” y “Name” ya han sido solucionados en la fase 0.

En esta fase de procesamiento de información se pretende solucionar el problema anteriormente descrito. Para ello se transformarán todos los datos cuyo valor sea 0, o 0.0 en el caso del atributo “Draught”, por el valor NULL. Es posible ver el resultado de las transformaciones anteriormente descritas, para el ejemplo de la tabla 5.27, en la tabla 5.28.

Entidad Model. Los datos utilizados para poblar esta entidad se corresponden con el atributo tipo de los ficheros obtenidos. En su formato original este atributo presenta ciertos problemas, al tener valores fuera del rango indicado en la sección 5.1, es decir fuera del rango

MMSI	IMO	Callsign	Name	a	b	c	d	draught
367422040	0	WCX5096	BLUE HORIZON	NULL	NULL	NULL	NULL	NULL
211590710	0	DA5898	FEUERLOESCHBOOT 1	NULL	NULL	NULL	NULL	0.2
244740200	0	PC2246	MAAIKE	40	15	9	2	NULL
205593000	9569009	ORPV	WESTERSCHELDE PILOT	14	11	6	6	NULL
413500010	0	BMEI	YUAN SHUN FA	75	23	8	8	5.9
412000001	8914934	DLMU	DLMU IMTS01	100	20	19	20	NULL
367177840	7338808	WBR4464 1	BART TURECAMO	10	23	2	3	4.0
367479630	8103004	WDF6958	MIAMI	62	21	5	21	4.0
413901677	0	NULL	YUESHAOGUANHUO0016	28	7	4	4	NULL
244050842	0	NULL	STORM	NULL	NULL	NULL	NULL	NULL

Tabla 5.28: Datos entidad Vessel transformados, fase 1

existente entre el valor 0 y el 99 así como valores NULL. A continuación se podrá ver un pequeño listado con ejemplos de este atributo en su formato original.

Type
NULL
15
255
99
35

Tabla 5.29: Raw Data Model

Para poblar esta tabla se ha decidido eliminar todos los valores nulos, así como todos los valores que no se encuentran dentro del rango 0-99 anteriormente descrito.

Entidad Trajectory. Como se ha visto anteriormente, tanto en la sección 5.1 como en la 5.2, los datos con los que se pretende poblar esta entidad se corresponden con la información relativa al viaje del navío, los cuales pueden verse en la tabla 5.8. En esta fase de procesamiento sólo se realizará una transformación en los datos, debido a la generación previa del identificador de trayectoria, así como la inserción de valores nulos en el atributo "Dest" llevada a cabo en la fase 0. Por ello en esta fase de procesamiento de la información los esfuerzos se centrarán en gestionar los posibles valores nulos existentes en el atributo "ETA" (*Estimated Time of Arrival*), así como en la conversión de este atributo a formato Unix (EPOCH).

A continuación se puede observar un ejemplo de los datos que poblarían la entidad Trajectory, tras su paso por la fase 0.

Id_Trajectory	Dest	ETA
204209890-2017-09-15	NULL	07-18 05:00
204601240-2017-09-15	HORTA	09-09 06:00
204670450-2017-09-15	NULL	00-00 00:00
204670870-2017-09-15	NULL	00-00 00:00

Tabla 5.30: Datos después de fase 0 Trajectory

En esta fase de procesamiento se pretende solucionar los problemas relativos al campo ETA, así como convertir su formato a Unix. Para la generación de valores nulos se tomarán como erróneos los valores '00-00 00:00'y '00-24 60:00'. En la tabla 5.31 es posible ver los datos de la tabla 5.30 una vez se ha realizado las transformaciones anteriormente descritas.

Id_Trajectory	Dest	*ETA*
204209890-2017-09-15	NULL	1531882800
204601240-2017-09-15	HORTA	1504929600
204670450-2017-09-15	NULL	NULL
204670870-2017-09-15	NULL	NULL

Tabla 5.31: Fase 1 Trajectory

Entidad Message. Los datos con los que se pretende poblar esta entidad se corresponden con los datos dinámicos del navío, ver tabla 5.3. A continuación se mostrará un pequeño ejemplo en el que se pueden ver los datos referentes a esta entidad una vez han pasado la fase 0 de limpieza:

Id_Message	NavStat	Latitude	Longitude	SOG	COG	Heading	TimeStamp
367422040-1505472175	0	33.76183	-118.19601	0	360	511	1505472175
211590710-1505472189	0	51.4449019	6.734630	0	257.69999999	281	1505472189
244740200-1505472188	0	52.3950499	4.8256800	0	0	511	1505472188
205593000-1505472191	0	51.2266800	2.93493	0	239.69999999	125	1505472191
413500010-1505472050	5	22.57339	113.4931	0	0	511	1505472050

Tabla 5.32: Fase 0 Message

Conociendo los valores especiales existentes en cada registro es posible identificar 2 transformaciones básicas sobre estos campos:

- Inserción de nulos: Los valores por defecto 360.0, 1023 y 511 indican que no hay información disponible sobre los atributos "SOG", "COG" y "Heading", respectivamente, en dichos registros. Se transformarán por tanto esos valores a NULL, así como todos los valores fuera de los rangos 0-359, 0-102.2 y 0-359 respectivamente. También se insertarán valores nulos en "latitude" y "longitude", sustituyendo así a los valores por defecto 91 y 181 por el valor NULL.
- Redondeo a 1 decimal: Se redondearán los valores relativos al COG y al SOG a 1 decimal.

A continuación se puede ver como quedaría este conjunto de datos una vez se han realizado los cambios anteriormente mencionados, así como los cambios realizados en la fase 0.

Id_Message	NavStat	Latitude	Longitude	SOG	COG	Heading	TimeStamp
367422040-1505472175	0	33.76183	-118.19601	0	NULL	NULL	1505472175
211590710-1505472189	0	51.4449019	6.734630	0	257.7	281	1505472189
244740200-1505472188	0	52.3950499	4.8256800	0	0	NULL	1505472188
205593000-1505472191	0	51.2266800	2.93493	0	239.7	125	1505472191
413500010-1505472050	5	22.57339	113.4931	0	0	NULL	1505472050

Tabla 5.33: Datos de la entidad Vessel transformados, fase 1

Entidad Port. En esta entidad los datos que se almacenan no provienen de AISHub, a diferencia de la información de las anteriores, sino de la NGA, en la tabla 5.19 se pueden ver los campos con la información que pretende almacenarse en dicha entidad. A continuación se puede ver un ejemplo de los datos a almacenar en su formato original:

Index	Zone_Code	Country_Code	Port_Name	Latitude	Field5	Combo353	Longitude	Field8	Combo214	Harbor_size	Harbor_type
70	60	IS	KEFLAVIC	60	0	N	22	33	W	V	OR
75	60	IS	STRAUMSVIK	64	3	N	22	3	W	V	CN
80	60	IS	HAFNARFJORDUR	64	4	N	21	57	W	V	CN
90	60	IS	SKERJAFJORDUR	64	9	N	22	1	W	V	CN
100	60	IS	REYKJAVIK	64	9	N	21	56	W	M	CB

Tabla 5.34: Raw Data Ports

Como a simple vista se puede observar tanto la latitud como la longitud se encuentran en diferentes campos. Para gestionar esta información de manera eficiente se deberán unir los campos “Latitude”, “Field5” y “Combo353”, así como los campos “Longitude”, “Field8” y “Combo214”, creando así dos campos Latitude y Longitude con la información completa. Sin embargo estos no son los únicos cambios a realizar en este fichero, ya que existen valores en blanco en los atributos Harbor_size y Harbor_type, los cuales serán cambiados por el valor NULL.

En la tabla 5.35 se podrán ver los mismos registros una vez se han realizado los cambios anteriormente descritos.

Index	Zone_Code	Country_Code	Port_Name	Latitude	Longitude	Harbor_size	Harbor_type
70	60	IS	KEFLAVIC	640N	2233W	V	OR
75	60	IS	STRAUMSVIK	643N	223W	V	CN
80	60	IS	HAFNARFJORDUR	644N	2157W	V	CN
90	60	IS	SKERJAFJORDUR	649N	221W	V	CN
100	60	IS	REYKJAVIK	649N	2156W	M	CB

Tabla 5.35: Datos entidad Port transformados, Fase 1

5.4.1. Fase 2

En esta fase se persigue extraer valor sobre los datos ya almacenados, permitiendo así entre otras cosas generar nuevos atributos o eliminar mensajes duplicados o erróneos. A continuación se describen en detalle las diferentes transformaciones realizadas en esta fase, ordenadas en función de la entidad sobre la que se realizan.

Entidad Vessel. Como se explica en el apartado 5.1 es posible conocer la nacionalidad del navío en función de los 3 primeros dígitos que forman el "MMSI", así como conocer si los mensajes son emitidos por equipos *Search and Rescue*, *SAR*. Por tanto para esta entidad en esta fase se ha decidido generar un nuevo atributo "FLAG", encargado de almacenar la información de la bandera del navío. En este atributo se almacenara por tanto los 3 primeros dígitos de cada MMSI. Es posible conocer la relación entre los datos almacenados referentes a la bandera y el país que identifican gracias a *VTExplorer* [27]. En la tabla 5.36 se puede ver un ejemplo de los datos almacenados en la entidad vessel una vez se ha realizado la transformación anteriormente descrita. (Posiblemente eliminar los mensajes con MMSI empezando por 200)

MMSI	IMO	Callsign	Flag	Name	a	b	c	d	Draught
367422040	0	WCX5096	367	BLUE HORIZON	NULL	NULL	NULL	NULL	NULL
211590710	0	DA5898	211	FEUERLOESCHBOOT 1	NULL	NULL	NULL	NULL	0.2
244740200	0	PC2246	244	MAAIKE	40	15	9	2	NULL
205593000	9569009	ORPV	205	WESTERSCHELDE PILOT	14	11	6	6	NULL
413500010	0	BMEI	413	YUAN SHUN FA	75	23	8	8	5.9
412000001	8914934	DLMU	412	DLMU IMTS01	100	20	19	20	NULL
367177840	7338808	WBR4464 1	367	BART TURECAMO	10	23	2	3	4.0
367479630	8103004	WDF6958	367	MIAMI	62	21	5	21	4.0
413901677	0	NULL	413	YUESHAOGUANHUO0016	28	7	4	4	NULL
244050842	0	NULL	244	STORM	NULL	NULL	NULL	NULL	NULL

Tabla 5.36: Datos entidad vessel transformados, paso a fase 2

5.5. Mapa de Datos

En esta sección se persigue describir las transformaciones, previamente explicadas, de forma gráfica, permitiendo identificar de un vistazo todas las transformaciones que permiten convertir el *Raw Data* a *Refined Data*.

Para describir estas transformaciones se suele utilizar un mapa de datos, gráfico visual en forma de matriz que permite relacionar el raw data y el refined data. El uso de este tipo de herramientas de documentación deriva de la gran variedad de fuentes de datos que son almacenadas en las actuales arquitecturas Big Data, cada cual con unas características particulares que deben ser consideradas para la construcción del proceso ETL. El mapa de datos se considera por tanto básico para la construcción de las acctividades que componen el proceso ETL, al aportar información fundamental para la obtención de metadatos y para la gestión de calidad.

En la tabla 5.37 se describe el mapa de datos que se utilizará para la construcción del proceso ETL asociado a este proyecto.

Origen		Transformaciones								Destino	
Fuente	Atributo	T0	Descripción	T1	Descripción	T2	Descripción	T3	Descripción	Entidad	Atributo
AISHub	NavStat				Sin cambios					Message	NavStat
	SOG			✓	Redondeo a 1 decimal y conversión de valores erróneos					Message	SOG
	Longitude			✓	Nullable					Message	Longitude
	Latitude			✓	Nullable					Message	Latitude
	COG			✓	Redondeo a 1 decimal					Message	COG
	Heading			✓	Insertar nulos					Message	Heading
	TSTAMP	✓	Eliminación de comillas y conversión a EPOCH time							Message	Timestamp
	MMSI				Sin cambios	✓	Generación del nuevo atributo Flag, utilizando los 3 primeros dígitos del MMSI			Vessel	MMSI
	IMO				Sin cambios					Vessel	IMO
	Callsign	✓	Eliminación de comillas, nullable							Vessel	Callsign
	Name	✓	Eliminación de comillas, nullable							Vessel	Name
	A			✓	Nullable					Vessel	D_Bow
	B			✓	Nullable					Vessel	D_Stern
	C			✓	Nullable					Vessel	D_Port
	D			✓	Nullable					Vessel	D_Starboard
	Draught			✓	Nullable					Vessel	Draught
	Type			✓	Eliminación de nulos y valores erróneos					Model	Id_Type
	ETA	✓	Eliminación de comillas	✓	Conversión a EPOCH time, nullable					Trajectory	ETA
	Dest	✓	Eliminación de comillas nullable							Trajectory	Destination
NGA	Index				Sin cambios					Port	Index
	Zone_Code				Sin cambios					Port	Zone_Code
	Country_Code				Sin cambios					Port	Country_Code
	Port_Name				Sin cambios					Port	Port_Name
	Latitude			✓	Se unen los campos Latitude, Field5 y Combo353					Port	Latitude
	Field5			✓	Se unen los campos Latitude, Field5 y Combo353						
	Combo353			✓	Se unen los campos Latitude, Field5 y Combo353						
	Longitude			✓	Se unen los campos Longitude, Field8 y Combo214					Port	Longitude
	Field8			✓	Se unen los campos Longitude, Field8 y Combo214						
	Combo214			✓	Se unen los campos Logitude, Field8 y Combo214						
	Harbor_size			✓	Nullable					Port	Harbor_Size
	Harbor_Type			✓	Nullable					Port	Harbor_Type
Generación	Id_Trajectory	✓	Se genera un id de trayectoria							Trajectory	Id_Trajectory
	Id_Message	✓	Se genera un id de mensaje							Message	Id_Message

Tabla 5.37: Mapa de Datos

Capítulo 6

Análisis

En este capítulo se persigue realizar un análisis detallado sobre el sistema desarrollado en este proyecto, partiendo de la aplicación de captación de datos y continuando con la aplicación web de visualización desarrollada. Pese a estar desarrollando un proyecto más orientado a la investigación, el hecho de desarrollar una aplicación web, con el fin de facilitar la visualización de las trayectorias y así contar con una herramienta más a la hora de analizar la gran cantidad de información de la que se dispone, hace que sea plausible poder realizar una pequeña fase de análisis de la misma. Por esto a lo largo de este capítulo se llevará a cabo una especificación de los diferentes tipos de requisitos existentes en un proyecto de desarrollo, así como el desarrollo de las historias de usuario asociadas a los requisitos de usuario especificados, al trabajar siguiendo una metodología ágil. El principal elemento en torno al que gira todo el análisis del proyecto son los requisitos de usuario, los cuales se encargan de describir las interacciones entre los usuarios y la aplicación desarrollada.

Al encontrarse especificados los objetivos asociados al proyecto en la sección 1.2 se pasará a iniciar la fase de análisis describiendo las diversas limitaciones, tanto hardware como software e incluso en la información disponible para trabajar, con las que nos encontraremos a lo largo del proyecto.

6.1. Limitaciones y restricciones de implementación

En el desarrollo de este proyecto se han podido encontrar ciertas limitaciones, siendo la principal el hecho de que sólo se dispone de datos de un mes concreto de datos, entre el 15 de septiembre y el 15 de octubre, impidiendo así realizar un seguimiento del ecosistema marítimo en tiempo real. El hecho de contar información únicamente en el rango temporal mencionado no constituye la única limitación del sistema, ya que unido a ese problema se encuentra el ámbito de los datos obtenidos, al ser datos de tipo costero y por tanto no conseguir dar soporte a la trayectoria completa de un navío, al no disponer de la información que emite el navío al estar en alta mar. Este hecho se deriva de que en la licencia utilizada para obtener la información no daba soporte a la captación de datos vía satélite. Pese a las limitaciones ya descritas se ha decidido continuar con el desarrollo del proyecto creando las diferentes fases de captación y procesamiento de los datos por si en un futuro se puede tener acceso a licencias de pago más completas. Además de las limitaciones anteriormente descritas se han podido encontrar limitaciones derivadas del hardware del que se dispone, al disponer de un clúster con un único nodo con una capacidad de almacenamiento menor que la cantidad de datos obtenidos.

De una manera más específica sobre la aplicación se han podido encontrar ciertas restricciones, derivadas del software al que se tiene acceso, al contar con una versión *CDH 5.13 Standalone* de Cloudera como entorno altamente desactualizada, ya que las versiones posteriores son de pago. El hecho de contar con una versión Cloudera antigua lleva consigo el hecho de contar con una versión del sistema operativo CentOS 6.9, la cual a su vez también se encuentra desactualizada y que entre otras cosas no permite una correcta integración con GitHub[12], imposibilitando la realización de pruebas de integración continua, al ser necesario para el uso de GitHub, al cual se conecta una aplicación externa como es Travis[4] mediante la cual se automatizan los test.

Una vez descritas las limitaciones, tanto a nivel de proyecto como a nivel de aplicación, se pasará a realizar la especificación, anteriormente mencionada, comenzando por realizar una descripción de los actores.

6.2. Actores del Sistema

En esta sección se realizará la especificación asociada a los diferentes actores que interactúan con la aplicación web desarrollada. Se denominan como actores a todas las entidades externas al sistema que interactúan con el mismo. Siguiendo esta definición es posible determinar que los actores pueden ser tanto personas físicas como otros sistemas con los que interactúa el sistema a desarrollar.

ACT-01	**Usuario General**
Versión	1.0 (16/05/2018)
Descripción	Se considera como usuario general a todo usuario capaz de acceder a la aplicación y disfrutar de sus servicios de visualización, no existe la necesidad de ningún tipo de registro
Comentarios	Ninguno

Tabla 6.1: Actor Usuario General

ACT-02	**AISHub Api**
Versión	1.0 (16/05/2018)
Descripción	Actor secundario a través del que se interactúa con el fin de obtener los datos de navegación marítima
Comentarios	Ninguno

Tabla 6.2: Actor AISHub API

6.3. Product Backlog

En esta sección se persigue especificar y describir las diferentes historias de usuario que deberán ser cumplidas al final del desarrollo de este proyecto. Se puede definir una historia de

usuario como cualquier actividad que podrá realizar el usuario mediante interacciones con el sistema a desarrollar. En definitiva las historias de usuario materializan progresivamente los objetivos de negocio. Las historias de usuario se identificarán en el product backlog, el cual consiste en una lista priorizada de requisitos que representa la visión y expectativas del cliente respecto a los objetivos que tienen sobre el proyecto.

A continuación, en la tabla 6.3, se podrá ver el Product Backlog de las historias de usuario que se tendrán en cuenta en el desarrollo de la herramienta desarrollada en este proyecto, así como una breve descripción de las mismos.

ID	Descripción
US-01	El usuario general podrá visualizar un listado de los diferentes navíos, de los que se tienen datos, en el que se mostrarán los datos estáticos del navío
US-02	El usuario general podrá buscar buscar los datos de un navío
US-03	El usuario general podrá visualizar un mapa en el que aparezcan los navíos de un tipo determinado y en un momento indicado
US-04	El usuario general podrá filtrar los navíos a mostrar en el mapa en función del tipo
US-05	El usuario general podrá filtrar los navíos a mostrar en función de la fecha
US-06	El usuario general podrá filtrar los navíos a mostrar en función del nombre
US-07	El usuario general podrá ver un mapa de calor creado por las posiciones de los navíos de un tipo previamente indicado y en un momento específico
US-08	El usuario general podrá visualizar la trayectoria seguida por un navío
US-09	El usuario general podrá visualizar un conjunto de métricas derivadas de los datos globales de navíos almacenados, así como algunas limitadas a ciertos puertos de interés

Tabla 6.3: Product Backlog

Existen diferentes técnicas utilizadas para entender los requisitos de usuario previamente descritos, destacando las técnicas de casos de uso, utilizada en desarrollos tradicionales, y de historias de usuario, enfocadas a su uso en proyectos que utilicen metodologías ágiles. A continuación se detallarán las diferentes historias de usuario, mediante las que se persigue especificar las historias de usuario listadas en el product backlog.

6.3.1. Historias de Usuario

En este apartado se realizará una especificación detallada de los historias de usuario, anteriormente listadas en el Product Backlog. Esta técnica es altamente utilizada en aquellos proyectos, que como este, son planificados utilizando metodologías ágiles. Una historia de usuario es la unidad de trabajo más pequeña que se puede identificar en el marco de las metodologías ágiles, constituyendo un objetivo final expresado desde el punto de vista de un usuario final del sistema a desarrollar. Este tipo de técnicas han adquirido una gran importancia en los últimos años, comenzando a ser puestas en práctica en una gran cantidad de proyectos.

Las historias de usuario se pueden agrupar en bloques, de mayor tamaño, surgiendo así las épicas y las iniciativas. A continuación, en la figura 6.1, se podrá ver la jerarquía existente entre los elementos anteriormente descritos.

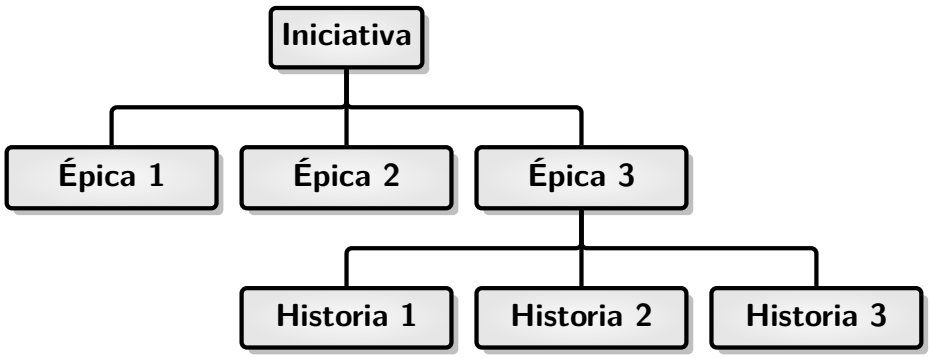


Figura 6.1: Jerarquía historias de usuario

Una vez conocida la jerarquía existente entre los diferentes elementos utilizados para tratar de entender los requisitos de usuario utilizando historias de usuario se procederá a especificar en detalle los requisitos de usuario anteriormente descritos. El proyecto a desarrollar puede dividirse en 3 épicas y una historia de usuario independiente a estas, las cuales se detallan a continuación, al igual que las historias de usuario.

ID Épica	Descripción Épica				
EP-01	Como usuario de la plataforma deseo visualizar la información referente a los diferentes navíos disponibles en la plataforma, para así poder acceder rápidamente a la información de los navíos				
ID Historia	**Descripción "Historias de usuario"**	**Estatus**	**Prioridad**	**Estimaciones**	**Propietario**
US-01	Como usuario de la plataforma deseo tener acceso a un listado con la información asociada a los diferentes navíos disponibles en la plataforma, para así facilitar su visualización				
Criterios de aceptación					
1	Listar navíos				
Dado que	Utilizo la aplicación web				
Cuando	Accedo a la plataforma o navego a la opción "Explorer"				
Entonces	Se muestra un listado en forma de tabla con la información estática asociada a los diferentes navíos disponibles en la plataforma				
ID Historia	**Descripción "Historias de usuario"**	**Estatus**	**Prioridad**	**Estimaciones**	**Propietario**
US-02	Como usuario de la plataforma deseo poder buscar navíos en función de sus características				
Criterios de aceptación					
1	Listar navíos				
Dado que	Me encuentro en la opción "Explorer"				
Cuando	Utilizo el buscador asociado a la tabla				
Entonces	Se realiza un filtrado, con independencia del atributo de la tabla en el que se desee buscar y se muestran los resultados asociados a la búsqueda				

Tabla 6.4: Épica 01

ID Épica	Descripción Épica				
EP-02	Como usuario de la plataforma deseo poder visualizar un mapa en el que se pueda observar el tráfico marítimo en un momento determinado asociado a un tipo de navío determinado				
ID Historia	**Descripción "Historias de usuario"**	**Estatus**	**Prioridad**	**Estimaciones**	**Propietario**
US-03	Como usuario de la plataforma quiero poder seleccionar el tipo de navío que deseo analizar				
Criterios de aceptación					
1	Seleccionar tipo de navío				
Dado que	Me encuentro en la opción "Dashboard Explorer"				
Cuando	Selecciono el tipo de navío, a través del desplegable				
Entonces	Se muestra un mapa con todos los navíos del tipo seleccionado				
ID Historia	**Descripción "Historias de usuario"**	**Estatus**	**Prioridad**	**Estimaciones**	**Propietario**
US-04	Como usuario de la plataforma quiero poder seleccionar el momento en el que deseo analizar el tráfico marítimo				
Criterios de aceptación					
1	Seleccionar fecha				
Dado que	Me encuentro en la opción "Dashboard Explorer"				
Cuando	Selecciono el momento en que deseo analizar el tráfico marítimo				
Entonces	Se muestra el tráfico marítimo de todos los navíos del tipo anteriormente seleccionado en el momento indicado				
ID Historia	**Descripción "Historias de usuario"**	**Estatus**	**Prioridad**	**Estimaciones**	**Propietario**
US-05	Como usuario de la plataforma deseo poder visualizar un mapa con la información marítima derivada de las elecciones previas realizadas para facilitar el análisis del tráfico marítimo				
Criterios de aceptación					
1	Ver mapa				
Dado que	Me encuentro en la opción "Dashboard Explorer"				
Cuando	Indico que deseo ver el mapa asociado a los filtrados realizados				
Entonces	Se muestra un mapa con el tráfico marítimo de todos los navíos del tipo anteriormente seleccionado en el momento indicado				
ID Historia	**Descripción "Historias de usuario"**	**Estatus**	**Prioridad**	**Estimaciones**	**Propietario**
US-06	Como usuario de la plataforma deseo poder visualizar un mapa de calor con la información marítima derivada de las elecciones previas realizadas para facilitar el análisis del tráfico marítimo				
Criterios de aceptación					
1	Ver mapa				
Dado que	Me encuentro en la opción "Dashboard Explorer"				
Cuando	Indico que deseo ver el mapa de calor asociado a los filtrados realizados				
Entonces	Se muestra un mapa calor asociado tráfico marítimo de todos los navíos del tipo anteriormente seleccionado en el momento indicado				

Tabla 6.5: Épica 02

ID Épica	Descripción Épica				
EP-03	Como usuario de la plataforma deseo poder visualizar la trayectoria seguida por un navío determinado, así como las estadísticas asociadas a la misma con el fin de dotar de valor a los datos almacenados				
ID Historia	**Descripción "Historias de usuario"**	**Estatus**	**Prioridad**	**Estimaciones**	**Propietario**
US-07	Como usuario de la plataforma quiero poder seleccionar el tipo de navío que deseo analizar				
Criterios de aceptación					
1	Seleccionar tipo de navío				
Dado que	Me encuentro en la opción "Dashboard Routes"				
Cuando	Selecciono el tipo de navío, a través del desplegable				
Entonces	Se permite seleccionar un navío a través de su nombre en un desplegable asociado, ya que sólo oferta los nombres de los navíos pertenecientes al tipo seleccionado				
ID Historia	**Descripción "Historias de usuario"**	**Estatus**	**Prioridad**	**Estimaciones**	**Propietario**
US-08	Como usuario de la plataforma quiero poder seleccionar el nombre del navío cuya ruta deseo analizar				
Criterios de aceptación					
1	Seleccionar nombre				
Dado que	Me encuentro en la opción "Dashboard Routes" y he seleccionado el tipo de navío a buscar				
Cuando	Selecciono el nombre del navío cuya trayectoria deseo analizar				
Entonces	Se muestra un mapa con la trayectoria seguida por el navío seleccionado, así como un conjunto de estadísticas derivadas de dicha trayectoria				

Tabla 6.6: Épica 03

A su vez también existe una historia de usuario independiente de ninguna épica, su especificación se encuentra en la tabla 6.7.

ID Historia	Descripción "Historias de usuario"	Estatus	Prioridad	Estimaciones	Propietario
US-9	Como usuario de la plataforma deseo poder visualizar métricas sobre el conjunto de valores disponibles con el fin de dotar de valor a dicha información				
Criterios de aceptación					
1	Ver métricas				
Dado que	Me encuentro en la opción "Metrics"				
Cuando	Accedo a la opción "Metrics"				
Entonces	Se muestra un conjunto de estadísticas derivadas del conjunto de datos disponible				

Tabla 6.7: Historia de Usuario 09

Una vez especificadas las diferentes historias de usuario que se desarrollarán en nuestro proyecto se pasara a realizar un listado de las diferentes tareas realizadas en cada sprint del proyecto, gracias al Sprint Backlog.

6.4. Sprint Backlog

En esta sección se persigue detallar las diferentes tareas que se han llevado a cabo en cada uno de los sprint del proyecto. Esta lista hace las veces de plan mediante el que se persiguen completar los objetivos, así como demostrar su consecución a los clientes en cada iteración.

En este proyecto los sprint tendrán una duración de 2 semanas. Sin embargo cabe destacar la existencia de un gran lapsus de tiempo en el que el proyecto estuvo parado, principalmente por su sincronización con la docencia del primer cuatrimestre. Pese a que la idea inicial era comenzar el proyecto en febrero se estableció un sprint inicial, entre el 7 y el 15 de septiembre, debido a la necesidad de tener datos disponibles para posteriormente realizar el proyecto. A continuación se muestra una tabla en la que se especificarán las diversas tareas que componen el Sprint Backlog.

Sprint Backlog					
Sprint	**Tarea**	**Descripción**	**Horas Estimadas**	**P. Historia**	**Estado**
1	T-01	Investigar los protocolos y el ecosistema marítimo	12	1	Completado
	T-02	Analizar y documentar las aplicaciones que utilizan y proveen de datos marítimos	30	1	Completado
	T-03	Crear una aplicación de descarga de datos marítimos	14	0,5	Completado
	T-04	Configurar el servidor para la ejecución ininterrumpida de la aplicación de descarga, en segundo plano	3	0,5	Completado
	T-05	Investigación sobre Data Lake, fase inicial	40	2	Completado
	T-06	Documentación introducción	6	0,5	Completado

Sprint Backlog					
2	T-07	Investigación y documentación en profundidad sobre el protocolo AIS	20	1	Completado
	T-08	Investigación y documentación en profundidad sobre el protocolo LRIT	10	0,5	Completado
3	T-09	Investigación sobre Data Lake, fase final	50	2	Completado
	T-10	Búsqueda de información asociada a los puertos a nivel mundial	10	0,25	Completado
	T-11	Análisis y documentación de las fuentes de información (Raw Data)	40	1	Completado
	T-12	Desarrollo de un modelo entidad relación y el diccionario de datos asociado	30	1	Completado
	T-13	Instalación y breve formación en el uso de Cloudera	30	0,75	Completado
4	T-14	Breve formación en HDFS y Hive	15	0,5	Completado
	T-15	Instalación del entorno R en local y formación	30	1,5	Completado
	T-16	Primer análisis de la información asociada a la navegación marítima usando hive	45	2	Completado
5	T-17	Finalizar la primera fase de limpieza de información	10	2	Completado
	T-18	Instalar R y RStudio Server en el nuevo servidor Cloudera	15	0,5	Completado
	T-19	Instalación de las diferentes librerías y paquetes, de R y del sistema, desarrollar la aplicación web	35	0,5	Completado
	T-20	Instalación de los paquetes de conexión entre R y Cloudera	50	2	Completado
	T-21	Documentación y análisis inicial de la plataforma web a utilizar	25	0,5	Completado
	T-22	Gestión de riesgos	10	0,25	Completado
	T-23	Diseño de interfaces	10	0,5	Completado
	T-24	Creación arquitectura	15	1	Completado

Sprint Backlog					
6	T-25	Desarrollo interfaz de usuario utilizando shinydashboard	10	1	Completado
	T-26	Formación en leaflet	10	0,5	Completado
	T-27	Test aplicación web	1	0,25	Completado
7	T-28	Análisis II aplicación web	5	1	Completado
	T-29	Diseó II aplicación web	5	0,5	Completado
	T-30	Conexión de la aplicación con cloudera	5	0,25	Completado
	T-31	Creación del mapa de los mapas de la aplicación	20	0,75	Completado
	T-32	Creación de tabla responsive con los datos de los navíos	5	0,25	Completado
	T-33	Implementación de los filtrados en el menú lateral	10	0,25	Completado
	T-34	Pruebas sobre la aplicación desarrollada hasta ahora	20	1	Completado
8	T-35	Formación map reduce	15	1	Completado
	T-36	Segunda fase de limpieza	20	1,5	Completado
	T-37	Carga de datos	15	1,5	Completado
9	T-38	Análisis métricas	5	0,5	Completado
	T-39	Diseño métricas	10	0,5	Completado
	T-40	Implementación de la visualización de métricas en la plataforma	10	1	Completado
	T-41	Pruebas finales aplicación	10	1	Completado
10	T-42	Revisiones finales aplicación y base de datos	15	1	Completado
	T-43	Documentación final	20	1	Completado

Tabla 6.8: Sprint Backlog

6.5. Gestión de Riesgos

Es un hecho que los proyectos software, al igual que en casi todos los aspectos de la vida, se encuentran expuestos a una enorme cantidad de riesgos que pueden influir en gran medida a la consecución de los objetivos que se espera conseguir con el desarrollo del proyecto. Este hecho dota por tanto de una gran importancia a la identificación, análisis y tratamiento de los diferentes riesgos que pueden afectar a los proyectos software. El análisis y gestión de riesgos trata de ofrecer herramientas que faciliten al equipo software la comprensión y el entendimiento de la incertidumbre generada por los riesgos. Esta gestión adquiere una enorme importancia debido a que un riesgo no identificado es capaz de, en el mejor de los casos, producir un impacto negativo significativo en la fecha de entrega del proyecto o en su presupuesto. Este hecho conlleva que cada vez se destinen más recursos al establecimiento de un buen Plan de

Gestión de Riesgos.

El plan de Gestión de Riesgos se encarga de definir:

- Un organigrama para la gestión de riesgos.
- El proceso de identificación y análisis de riesgos.
- Las herramientas y técnicas a utilizar.
- Plantillas estandarizadas para la identificación y gestión de riesgos.
- Actividades de control de riesgos y periocidad de las mismas.

Al ser este un Trabajo de Fin de Grado alguno de los aspectos del Plan de Gestión de Riesgos, anteriormente descritos, no serán tenidos en cuenta, como el control de riesgos, ya que al ser un proyecto de corta duración temporal no habrá mucho espacio a cambios en los riesgos. A continuación se llevará a cabo la identificación de los principales riesgos susceptibles de afectar a este proyecto.

ID	Título
RISK-01	Retrasos en la planificación
RISK-02	Desconocimiento de las tecnologías a utilizar
RISK-03	Cambios en la especificación de los requisitos
RISK-04	Problemas hardware en los servidores disponibles
RISK-05	Falta de experiencia del equipo de desarrollo
RISK-06	Problemas de salud en el equipo de desarrollo
RISK-07	Llegar a un punto sin salida en la investigación
RISK-08	Competencia

Tabla 6.9: Listado de riesgos

La identificación de riesgos en un proyecto suele ser muy problemática, ya que es prácticamente imposible identificar todos los riesgos a los que se enfrenta el desarrollo del proyecto.

Una vez identificados los principales riesgos que se tendrán en cuenta en el proyecto, se debe pasar a realizar un análisis cualitativo de los riesgos. Una de las herramientas más utilizadas con este fin es la Matriz de Probabilidad Impacto. Esta herramienta facilita la priorización de riesgos en función de su probabilidad de ocurrencia, así como del impacto asociado a los diferentes riesgos. A continuación se mostrarán los diferentes riesgos priorizados utilizando la Matriz de Probabilidad Impacto.

El siguiente paso a seguir, una vez realizada la priorización de riesgos, consiste en llevar a cabo actividades y medidas que permitan gestionar los riesgos. Con el fin de gestionar los riesgos se establecen dos tipos de planes:

- Plan de mitigación: Definida en el PMBOK como el conjunto de acciones a través de las cuales se pretende reducir la probabilidad de ocurrencia de un riesgo.
- Plan de contingencia: Conjunto de acciones realizadas como respuesta a la producción de un riesgo. Es el ejemplo perfecto de un plan reactivo de gestión de riesgos, es decir, trata de reducir el impacto del riesgo una vez este ya ha ocurrido.

A continuación se detallaran los planes de mitigación y contingencia asociados a los riesgos anteriormente priorizados.

ID	Título	Impacto	Probabilidad	Prioridad
RISK-01	Retrasos en la planificación	Alto	Media	4
RISK-02	Desconocimiento de las tecnologías a utilizar	Alto	Alto	2
RISK-03	Cambios en la especificación de los requisitos	Medio	Baja	7
RISK-04	Problemas hardware en los servidores disponibles	Medio	Media	6
RISK-05	Falta de experiencia del equipo de desarrollo	Alto	Media	5
RISK-06	Problemas de salud en el equipo de desarrollo	Medio	Muy baja	8
RISK-07	Llegar a un punto sin salida en la investigación	Alto	Media	3
RISK-08	Desarrollo del mismo producto por la competencia	Muy Alto	Media	1

Tabla 6.10: Listado de riesgos priorizados

ID	Título	Descripción
RISK-01	Retrasos en la planificación	Es posible que existan retrasos, en cuanto a la planificación, a lo largo del desarrollo del proyecto. Estos retrasos pueden derivarse de una mala planificación inicial
Gestión del riesgo		
Estrategia		Se persigue establecer una estrategia mediante la cual se evite el riesgo
Plan de mitigación		Para evitar problemas derivados de una mala planificación temporal se ha decidido añadir una variable de holgura a la planificación, es decir, se ha hecho una planificación pesimista en la que se trata de dar 3 días a mayores al final de cada sprint
Plan de contingencia		Incrementar el número de horas diarias de trabajo

Tabla 6.11: Risk-01. Retrasos en la planificación

ID	Título	Descripción
RISK-02	Desconocimiento de las tecnologías a utilizar	Al realizar un proyecto de investigación en el que tanto el ámbito del proyecto, como el entorno tecnológico es eminentemente novedoso es posible que existan problemas derivados del trabajo con tecnologías novedosas, al no haber sido estudiadas en la carrera
Gestión del riesgo		
Estrategia		Se persigue establecer una estrategia mediante la cual se prevenga el riesgo
Plan de mitigación		Con el fin de prevenir estos problemas antes de iniciar el proyecto se ha tratado de tomar una pequeña formación en las tecnologías a utilizar, gracias a plataformas de cursos online
Plan de contingencia		Incrementar el número de horas diarias de trabajo y contactar con los tutores del proyecto, ya que ellos se encuentran familiarizados con estas tecnologías al haber realizado proyectos de similares características

Tabla 6.12: Risk-02. Desconocimiento de las tecnologías

ID	Título	Descripción
RISK-03	Cambios en la especificación de requisitos	Al realizar un proyecto con un cliente real nos encontramos con la posibilidad de la ocurrencia de cambios en los requisitos del proyecto a desarrollar, más aún tratándose de un proyecto de investigación
Gestión del riesgo		
Estrategia		Se persigue establecer una estrategia mediante la cual se prevenga el riesgo
Plan de mitigación		Con el fin de prevenir y mitigar estos problemas se ha utilizado una metodología ágil, al destacar estas por su gran adaptabilidad a modificaciones en los requisitos
Plan de contingencia		Realizar los cambios pertinentes en los requisitos del proyecto

Tabla 6.13: Risk-03. Cambios en la especificación de los requisitos

ID	Título	Descripción
RISK-04	Problemas hardware en los servidores disponibles	Para el desarrollo de este proyecto son utilizados dos servidores ofertados por la Universidad de Valladolid. La caída o pérdida de estos servidores puede derivar en una pérdida irreparable en la información obtenida o en las aplicaciones encargadas de procesarla y utilizarla
Gestión del riesgo		
Estrategia		Se persigue establecer una estrategia mediante la cual se prevenga el riesgo
Plan de mitigación		Con el fin de prevenir este riesgo, además de contar con los sistemas de backup con los que trabaja la universidad, se ha decidido almacenar en la nube, en aplicaciones como Mega o Dropbox, los avances realizados, así como en varios discos duros externos. Este hecho facilita la recuperación de información en caso de problema hardware
Plan de contingencia		Tratar de solucionar el problema hardware y volver a desplegar el ecosistema al completo a partir de las copias realizadas

Tabla 6.14: Risk-04. Problemas hardware en los servidores disponibles

ID	Título	Descripción
RISK-05	Falta de experiencia del equipo de desarrollo	La posible falta de experiencia del equipo de desarrollo del proyecto puede suponer retrasos y sobrecostes en el proyecto, así como una implementación poco eficaz del mismo
Gestión del riesgo		
Estrategia		Se persigue establecer una estrategia mediante la cual se prevenga el riesgo
Plan de mitigación		Con el fin de prevenir este tipo de problemas se incluye tiempo dedicado a la formación del personal dentro de la planificación del proyecto
Plan de contingencia		Incrementar los recursos temporales y económicos orientados a la formación del personal

Tabla 6.15: Risk-05. Falta de experiencia del equipo de desarrollo

ID	Título	Descripción
RISK-06	Problemas de salud en el equipo de desarrollo	Al igual que en cualquier trabajo la planificación temporal realizada puede verse truncada debido a la posibilidad de que el personal de trabajo tenga problemas de salud, entre otro tipo de problemas, que deriven en la baja del trabajador.
Gestión del riesgo		
Estrategia		Se ha decidido aceptar el riesgo, dado a la baja probabilidad de que el personal de trabajo del proyecto, en el corto periodo temporal en que se desarrolla, tenga que recurrir a solicitar alguna baja
Plan de mitigación		Ante los problemas que pueden dar lugar a bajas laborales no es posible disponer de ningún plan que las prevengas, más allá de promover hábitos de vida saludables
Plan de contingencia		Aceptar los retrasos producidos por la baja laboral

Tabla 6.16: Risk-06. Problemas de salud en el equipo de desarrollo

ID	Título	Descripción
RISK-07	Llegar a un punto sin salida en la investigación	Al realizar un proyecto de investigación en el que el objetivo final puede ser cambiante puede terminar dando lugar a un punto muerto, hecho muy frecuente en los proyectos de investigación debido a la falta de herramientas con las que desarrollarlos
Gestión del riesgo		
Estrategia		La estrategia seguida en este caso sería la aceptación, ya que ante un proyecto que ha llevado a un callejón sin salida no hay otra opción que aceptarlo y tratar de reenfocar el proyecto
Plan de mitigación		Es imposible prevenir que la investigación a seguir vaya a alcanzar un punto muerto
Plan de contingencia		Tratar de buscar una reorientación del proyecto, tratando así de reciclarlo y que sea útil en otro ámbito

Tabla 6.17: Risk-07. Llegar a un punto sin salida en la investigación

ID	Título	Descripción
RISK-08	Desarrollo del mismo producto por la competencia	En un mundo cada vez más informatizado y globalizado es cada vez más complejo que dos personas se encuentren desarrollando soluciones distintas a un mismo problema, un ejemplo de esto se puede ver entre Telegram y WhatsApp. Este riesgo es muy alto en este proyecto, al tener constancia de que otro equipo de trabajo se encuentra desarrollando una solución similar
Gestión del riesgo		
Estrategia		Se persigue establecer una estrategia mediante la cual se evite el riesgo
Plan de mitigación		Es imposible saber si el producto desarrollado por la competencia es superior al tuyo o si se encuentra más desarrollado, para evitar problemas de competencia lo mejor es publicitar la herramienta y tratar de sacar una versión demo del producto antes que la competencia
Plan de contingencia		Tratar de desarrollar algo que nos diferencie del producto ya disponible

Tabla 6.18: Risk-08. Desarrollo del mismo producto por la competencia

Una vez especificados los diferentes planes de mitigación y contingencia diseñados para cada riesgo se habrá completado la gestión de riesgos del proyecto a desarrollar.

Capítulo 7

Gestión del Proyecto

En este capítulo se persigue describir y detallar la metodología seguida a lo largo del proyecto a desarrollar, así como los motivos de su utilización. Una vez descrita la metodología a seguir en el proyecto se detallará su planificación temporal y el coste asociado al desarrollo del mismo. Cabe destacar que al tratarse de un proyecto de investigación la planificación será un poco especial, siendo muy complejo realizar una estimación de costes previa, al no conocer con exactitud con que nos encontraremos a lo largo del proyecto.

7.1. Metodología

En la actualidad se está viendo un gran crecimiento en el número de proyectos que utilizan metodologías ágiles, viviendo el auge de la gestión de proyectos "DevOps". El auge de estas técnicas de gestión de proyectos se encuentra muy relacionado con los grandes y extremadamente rápidos cambios realizados en el mundo de la tecnología. Estos cambios tecnológicos suelen ocasionar una gran cantidad de cambios en los requisitos de un proyecto. Ante esta situación, las técnicas de gestión de proyectos tradicionales se encuentran con una gran cantidad de problemas derivados de su reducida capacidad de adaptación a cambios. Los cambios tecnológicos, junto con la gran cantidad de empresas dedicadas a ofrecer soluciones informáticas, han aumentado la necesidad de seguir estrategias que permitan el lanzamiento temprano de productos funcionales, no completos, ocasionando este otro problema a las técnicas de gestión de proyectos tradicionales. Para poner solución a estos problemas surgen las metodologías ágiles, las cuales ofrecen una mayor adaptabilidad a cambios y siguen estrategias orientadas a obtener una versión inicial del proyecto a mayor velocidad. Las metodologías ágiles se fundamentan sobre 4 pilares:

- Valorar más el software funcional que la documentación extensiva: La posibilidad de anticipar cómo será el funcionamiento del producto final, antes de haberlo desarrollado por completo, puede ofrecer un feedback muy interesante y estimulante. Este feedback permite generar ideas que serían muy difíciles de generar en una especificación detallada de requisitos en el inicio del proyecto.

- Valorar más la colaboración con el cliente que la negociación contractual: Los proyectos ágiles se encuentran orientados a proyectos de evolución continua, en los que sería imposible definir un documento inicial de requisitos que indique cómo debería ser el producto final.

- Valorar más a los individuos y su interacción que a los procesos y las herramientas: En los proyectos que siguen metodologías ágiles se persigue alcanzar las cotas deseadas de calidad basándose en los conocimientos del equipo de trabajo, más que en la calidad de las herramientas disponibles para trabajar, las cuales no deben perder importancia.

- Valorar más las respuestas a los cambios que el seguimiento de un plan: En el desarrollo de productos con requisitos inestables, en los que es inherente el cambio y la evolución rápida y continua, es mucho más valiosa la capacidad de respuesta que la de seguimiento de planes.

En la actualidad existen una gran cantidad de metodologías ágiles de trabajo, entre las que destacan Kanban y Scrum. Para el desarrollo de este proyecto se ha decidido utilizar una metodología ágil, más en concreto se ha decidido utilizar Scrum. La elección de una metodología ágil para la gestión del proyecto radica en la inestabilidad inherente a un proyecto de investigación, además de que, al ser un proyecto también realizado durante las prácticas para una empresa que no tiene sede en España, era la única manera de gestionar los avances del proyecto. Dentro de las metodologías ágiles se ha decidido utilizar Scrum, ya que es la metodología ágil más conocida por el equipo de trabajo. Scrum es una metodología ágil caracterizada por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.

- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados.

- Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra en un ciclo secuencial o de cascada.

Los proyectos que siguen esta metodología parten de una visión general del producto que se desea desarrollar, a partir de la cual se va especificando y detallando la funcionalidad a realizar. En esta metodología cada ciclo de desarrollo, llamado sprint, finaliza con la entrega operativa del producto, siendo la duración recomendada de un sprint menor de un mes. En scrum se suelen realizar reuniones diarias de seguimiento del proyecto, de corta duración, con el fin de monitorizar el avance del mismo. En este proyecto, a diferencia de un scrum clásico, las reuniones de seguimiento son semanales, ya que los participantes en las reuniones, tutores del proyecto y mi tutor de prácticas en la empresa, trabajan en otros proyectos, imposibilitando por tiempo las reuniones diarias de seguimiento.

Al finalizar cada sprint se suele llevar a cabo una reunión, entre todas aquellas personas involucradas en el desarrollo del proyecto, con el fin de poder revisar el resultado del sprint y valorar el incremento conseguido en el mismo.

El marco de scrum se encuentra formado por 3 componentes, roles, artefactos y eventos, siempre girando en torno al sprint. A continuación se puede ver una tabla en la que se describen estos componentes.

Componente	Tipos	Descripción
Roles	Equipo scrum	Grupo de profesionales que realizan el incremento en cada sprint
	Product Owner	Persona encargada de tomar las decisiones del cliente
	Scrum Master	Responsable del cumplimiento de las reglas de un marco de scrum técnico
Artefactos	Product Backlog	El product Backlog es la lista ordenada de todo aquello que el propietario de producto cree que necesita el producto
	Sprint Backlog	El sprint Backlog es la lista de las tareas necesarias para construir las historias de usuario que se van a realizar en un sprint
	Incremento	El incremento es la parte de producto producida en un sprint, y tiene como característica el estar completamente terminado y operativo
Eventos	Sprint	El evento clave de scrum para mantener un ritmo de avance continuo es el sprint, el periodo de tiempo acotado de duración máxima de 4 semanas, durante el que se construye un incremento del producto
	Reunión de planificación del sprint	En esta reunión se toman como base las prioridades y necesidades de negocio del cliente, y se determinan cuáles y cómo van a ser las funcionalidades que se incorporarán al producto en el siguiente sprint
	Scrum diario	Reunión diaria breve, de no más de 15 minutos, en la que el equipo sincroniza el trabajo y establece el plan para las siguientes 24 horas
	Revisión del sprint	Reunión realizada al final del sprint para comprobar el incremento
	Retrospectiva del sprint	Reunión en la que se realiza un autoanálisis de la forma de trabajar e identifica fortalezas y puntos débiles

Tabla 7.1: Explicación componentes de scrum

En la figura 7.1 se explica como interactúan los artefactos entre si, mostrando también el seguimiento de los eventos anteriormente descritos.

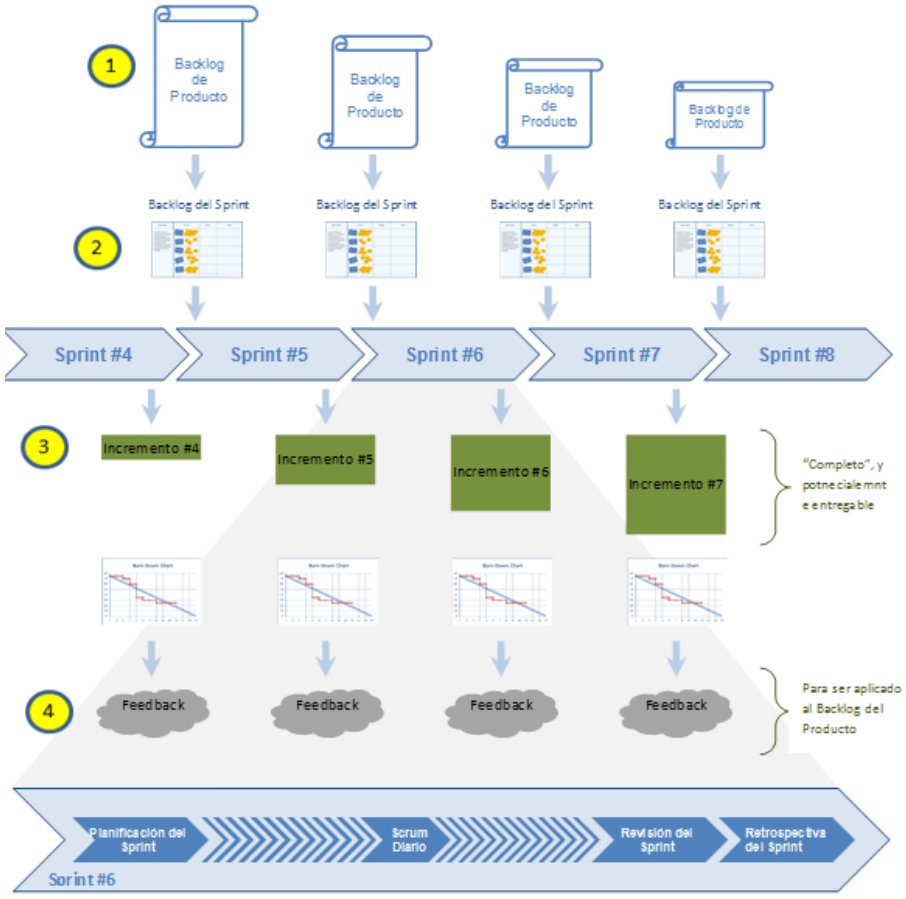


Figura 7.1: Scrum

A lo largo del Capítulo 6 se han descrito las diferentes historias de usuario de la plataforma, así como las tareas realizadas con el fin de desarrollar las diferentes herramientas del proyecto a realizar, descritas en el sprint backlog. Es posible utilizar herramientas de gestión de proyectos en las que puedan insertar las diferentes historias de usuario y tareas que componen los product y sprint backlog, generando a partir de ellas gran cantidad de gráficos que permiten realizar un seguimiento del proyecto. En la sección de planificación se detallarán los diferentes sprint del proyecto, mediante una serie de gráficos obtenidos a través de la herramienta VersionOne [5].

7.2. Planificación

Como previamente se ha comentado, la planificación del proyecto se ha realizado utilizando una metodología ágil llamada Scrum. La planificación de este proyecto se ha desarrollado en 10 sprints, cuyas tareas están especificadas en el sprint backlog (ver tabla 6.8). Una vez creado el sprint backlog y habiendo dotado a las tareas de una estimación temporal a cada una, así como los puntos de historia de usuario asociados a estas, es posible desarrollar gráficos que muestren el avance temporal del proyecto.

Como con anterioridad se ha comentado, la planificación en este proyecto es problemática. Este hecho deriva del carácter de investigación del proyecto al no tener como funcionalidad final el desarrollo de una herramienta software al uso, sino el análisis de la información marítima

mundial utilizando herramientas Big Data. El carácter de investigación asociado al proyecto, junto con la complejidad de las herramientas utilizadas, ha derivado en que la mayor parte de la planificación está destinada a la investigación del entorno marítimo y el Data Lake, así como a la formación tecnológica necesaria para analizar los datos de los que se dispone. Esto dificulta en gran parte la planificación del proyecto. Pese a haber planificado el proyecto utilizando una metodología ágil, esta no se visualiza de manera clara hasta el momento en el que se desarrolla una aplicación web de visualización de información, que sirva de complemento para facilitar el análisis de la información marítima. En los sprint dedicados al desarrollo de esta aplicación es posible ver los diferentes puntos que sigue scrum en el desarrollo de proyectos (análisis, diseño, codificación y pruebas).

Este proyecto estará dividido en 10 sprints, siendo la duración promedia de estos de 2 semanas. Cabe destacar la realización de un sprint inicial de 1 semana de duración entre el 7 y el 15 de septiembre, debido a la urgencia de captar información.

Como se ha comentado en la sección anterior la aplicación se utilizará una licencia gratuita de VersionOne con el fin de realizar la gestión y planificación del proyecto. Esta aplicación permite obtener gran variedad de gráficos, destacando el gráfico de velocidad del proyecto. Gracias a este gráfico es posible ver la velocidad de desarrollo del proyecto en función de los puntos de historia estimados en el sprint backlog. Este gráfico supone una alternativa al gráfico burndown, permitiendo analizar el conjunto de sprints del proyecto y ayudando a conocer la evolución del proyecto, así como el porcentaje de proyecto realizado en cada sprint gracias a un histograma. A continuación se mostrará dicho histograma (en la figura 7.2). Cabe destacar en ese histograma que los trozos de gráfico en azul marcan los puntos de historia que no se han completado en ese sprint.

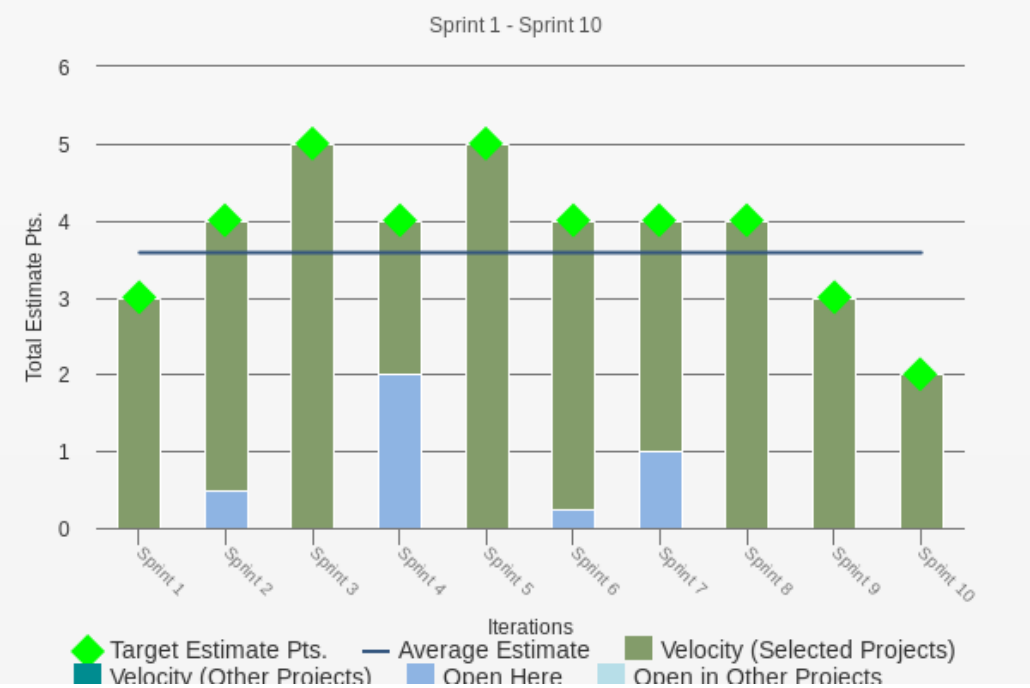


Figura 7.2: Histograma de evolución del proyecto mediante puntos de historias

A continuación se detalla la correspondencia entre los puntos de historia de usuario y el tiempo de cada sprint, con el fin de apoyar al gráfico anterior en términos temporales. Es

posible ver esto en la tabla 7.1.

Sprint	Duración total (horas)	Puntos de historia	Retraso temporal (horas)	Retraso puntos
1	58	3	0	0
2	76	4	1	0,5
3	160	5	0	0
4	90	4	15	2
5	110	5	0	0
6	82	4	0	0
7	70	4	10	1
8	50	4	5	1,5
9	30	3	0	0
10	20	2	0	0

Tabla 7.2: Detalle temporal-puntos en cada sprint

Una vez visto el gráfico de seguimiento del proyecto puede destacar el gran porcentaje de puntos de historia no completado en el sprint 4. En este caso se tomó la decisión de continuar con la tarea de limpieza de datos, tarea equivalente a ese retraso, en el siguiente sprint. Este retraso en el proyecto está relacionado con un problema en los servidores con los que se está desarrollando el proyecto. En el resto de sprint al no suponer un gran retraso se ha decido hacer horas extra los fines de semana con el fin de alcanzar a la planificación inicial del proyecto.

7.3. Presupuesto Final

Al ser un proyecto de investigación es prácticamente imposible realizar una estimación inicial sobre el presupuesto asociado al desarrollo del proyecto a realizar, por ello se ha decidido pasar directamente a explicar el presupuesto final que supone el desarrollo de este proyecto, una vez finalizado. Para la realización del presupuesto del proyecto se cuenta con una duración de 10 meses. A continuación se podrá ver un desglose del presupuesto final del proyecto.

- **Presupuesto Hardware**: El presupuesto asociado al hardware del proyecto se prorrateará en función de la duración temporal del proyecto y la vida útil de los componentes. A continuación se podrá ver una tabla con el desglose del coste asociado al hardware.

Hardware	Coste Total	Vida Útil	% de uso	Coste real
Ordenador personal	800 €	5 años	16 %	128€
Servidor Cloudera				0 €
Servidor Almacenamiento				0 €
Conexión a Internet	50 €/mes	10 meses	100 %	500 €

Tabla 7.3: Coste componentes Hardware

- **Presupuesto Software** En la tabla 7.3 se muestran los costes asociados a los productos Software utilizados para el desarrollo del proyecto realizado. Los valores de los costes software se calcularán siguiendo el mismo patrón que se ha seguido para realizar el presupuesto hardware. Cabe destacar que para la realización de este proyecto se ha optado por utilizar tecnología open-source, por lo que en su mayoría es gratuito.

Software	Coste Total	Vida Útil	% de uso	Coste real
Debian 9	0 €			0 €
CentOS 6.9	0 €			0 €
Pencil	0€			0€
Photoshop	24,19 €/mes	3 meses		72,57 €
Netbeans	0€			0€
Eclipse	0€			0€
Cloudera CDH	0€			0€
TexStudio	0€			0€
RStudio-Server	0€			0€
VersionOne	0€			0€

Tabla 7.4: Coste Software

- **Presupuesto de mano de obra**: Para realizar el presupuesto asociado a la mano de obra del proyecto se ha contado con el salario correspondiente un único ingeniero informático, de 22.646'17 €/año. Para calcular el sueldo total a cobrar sería necesario multiplicar el número de horas invertidas en el proyecto por el salario por hora a cobrar por la persona encargada de su desarrollo. A continuación se puede ver el coste asociado al personal

  $$22{,}646{,}17\ euros/\text{año} \longrightarrow 11{,}23\ euros/hora$$

  Una vez tenemos precio por hora asociado al personal, se podrá calcular el presupuesto asociado a la mano de obra final.

  $SueldoTotal = precio/hora\ * hora \longrightarrow SueldoFinal = 11{,}23\ euros/hora\ *\ 748\ horas$

  El salario final a percibir por la persona encargada del desarrollo del proyecto será de: 8400,04€

Una vez se han calculado los diferentes costes asociados al proyecto es posible obtener el presupuesto total del proyecto realizado. A continuación puede verse el cálculo del coste total del proyecto.

$Presupuesto\ Total = C.Hardware + C.Software + C.Personal$

$Presupuesto\ Total = 628 + 72{,}57 + 8400{,}04 = 9100{,}61$€

Capítulo 8

Diseño

En este capítulo se persigue desarrollar el trabajo de diseño previo al desarrollo de la aplicación web. En este capítulo por tanto se detallarán las arquitecturas lógica y física utilizadas para el desarrollo del proyecto, así como el diseño de las interfaces de la aplicación web creada.

8.1. Arquitectura lógica

En esta sección se persigue detallar los diferentes componentes lógicos que forman el sistema, así como la relación existente entre los mismos. La arquitectura lógica de este proyecto se corresponderá con la aplicación web desarrollada en R, gracias a shiny. Al ser R un lenguaje de muy alto nivel, unido al hecho de la facilidad de desarrollo de aplicaciones web utilizando shiny, ha derivado en una arquitectura lógica de grán sencillez. A continuación se podrá ver la arquitectura lógica de este proyecto, descrita en la imagen 8.2.

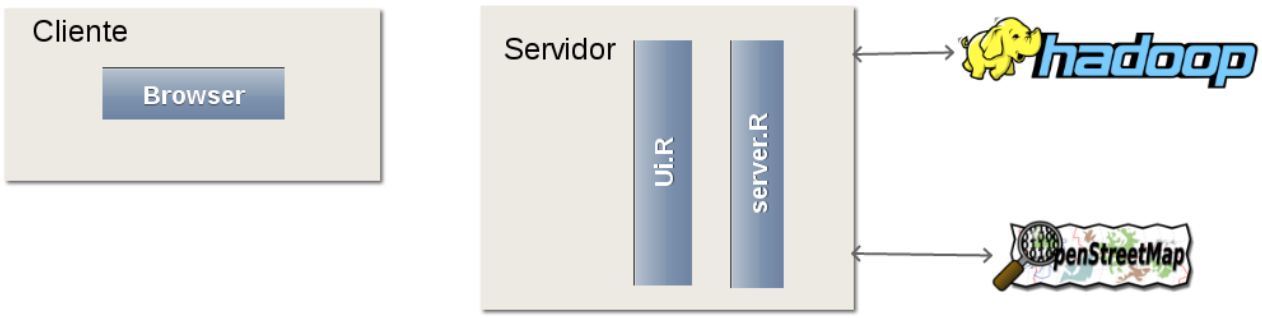


Figura 8.1: Arquitectura lógica

8.2. Arquitectura Física

En esta sección se persigue describir la arquitectura física utilizada a lo largo del proyecto. Cabe destacar que dentro de la arquitectura física debería de encontrarse la arquitectura del Data Lake, ampliamente detallada en el capítulo 3. Otro hecho a tener en cuenta son las limitaciones en cuanto a recursos con los que se ha realizado el proyecto, siendo imposible

desarrollar una arquitectura física que facilite las mejores capacidades de gestión de concurrencia o de integridad y recuperación de la información.

El proyecto se ha realizado utilizando 2 servidores de la Universidad de Valladolid, uno con el fin de captar y mantener una copia en crudo de la información y otro con un entorno Cloudera utilizado para el procesamiento de la información y el despliegue de la aplicación web. A continuación es posible ver la arquitectura física con la que se ha desarrollado este proyecto.

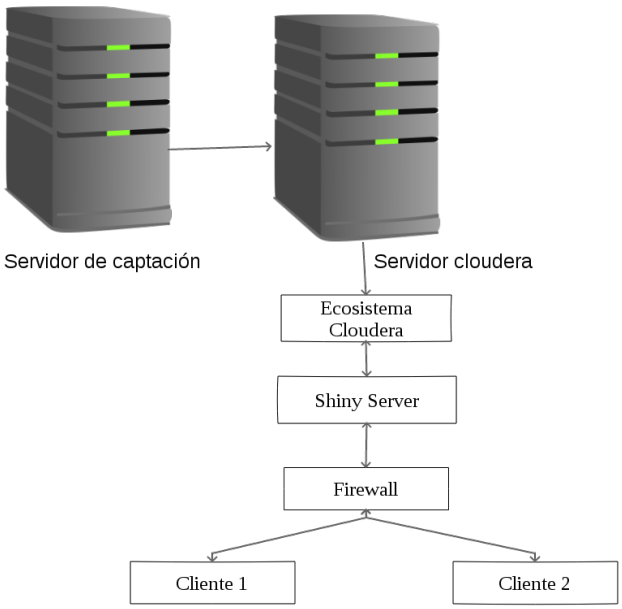


Figura 8.2: Arquitectura Física utilizada

8.3. Diseño de interfaces

En esta sección se persigue detallar las diferentes interfaces que compondrán la aplicación web desarrollada a modo de dashboard. A continuación se pueden ver estas interfaces.

La pestaña “Dashboard Explorer” diseñada a continuación cuenta con un “tab box” con dos pestañas, lo que hará que cambié por tanto la visualización del body, a continuación se podrá ver la interfaz correspondiente a ambas pestañas.

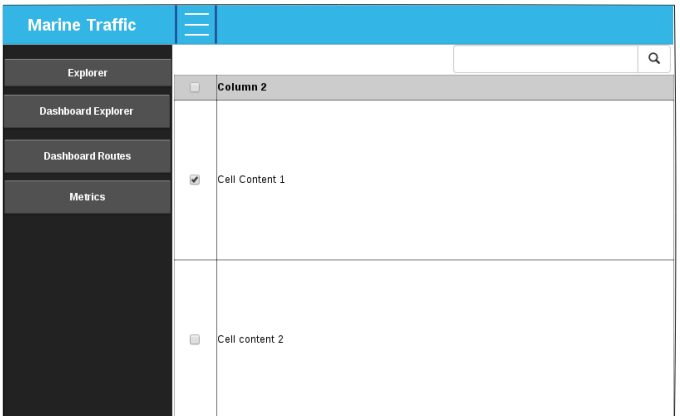
Nombre	Explorer
Descripción	Pestaña en la que es posible visualizar un listado con los navíos disponibles en la aplicación
Activación	El usuario accede a la aplicación o a la opción Explorer del menú lateral
Boceto	
Eventos	Buscar navío, ver navios, ir a "Dashboard Explorer", ir a "Dashboard Routes" o ir a "Metrics"

Tabla 8.1: DIU-01 Explorer

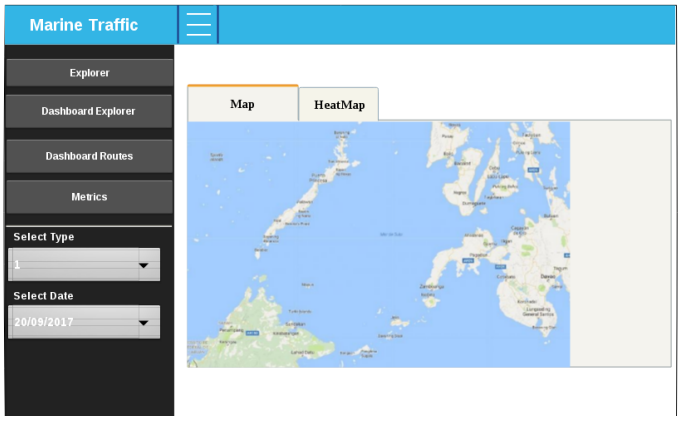
Nombre	Dashboard Explorer
Descripción	Pestaña encargada de visualizar un mapa que muestre el tráfico marítimo mundial de los navíos de un tipo y en una fecha determinados mediante filtrados previos
Activación	El usuario accede a la opción Dashboard Explorer del menú lateral
Boceto	
Eventos	Seleccionar tipo, seleccionar fecha, ver mapa, ver heatmap, ir a "Explorer", ir a "Dashboard Routes" o ir a "Metrics"

Tabla 8.2: DIU-02 Dashboard Explorer Map

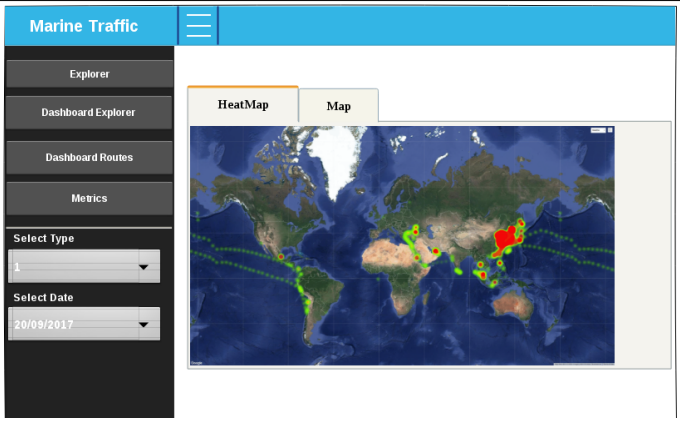
Nombre	Dashboard Explorer
Descripción	Pestaña encargada de visualizar un mapa que muestre el tráfico marítimo mundial de los navíos de un tipo y en una fecha determinados mediante filtrados previos
Activación	El usuario accede a la opción Dashboard Explorer del menú lateral
Boceto	
Eventos	Seleccionar tipo, seleccionar fecha, ver mapa, ver heatmap, ir a “Explorer”, ir a “Dashboard Routes” o ir a “Metrics”

Tabla 8.3: DIU-03 Dashboard Explorer Heatmap

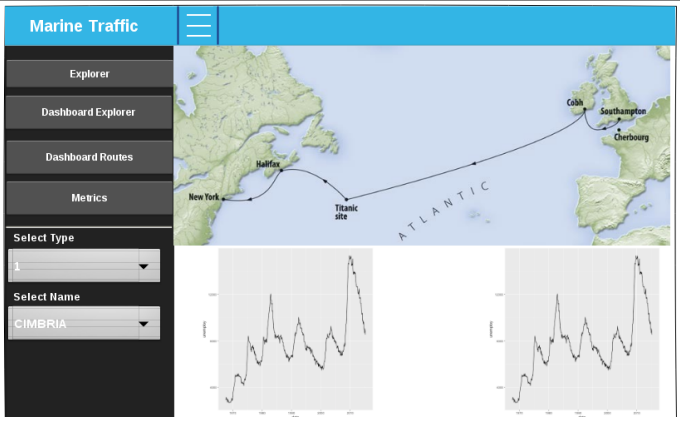
Nombre	Dashboard Routes
Descripción	Pestaña encargada de visualizar un mapa que muestre la trayectoria de un navío indicado
Activación	El usuario accede a la opción Dashboard Routes del menú lateral
Boceto	
Eventos	Seleccionar tipo, seleccionar nombre, ir a “Explorer”, ir a “Dashboard Explorer” o ir a “Metrics”

Tabla 8.4: DIU-04 Dashboard Routes

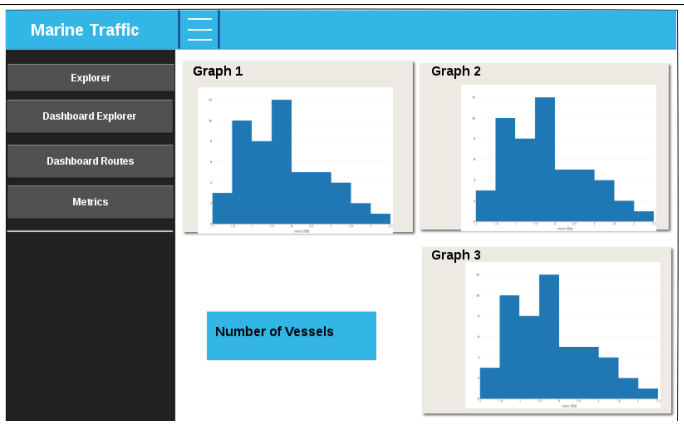
Nombre	Metrics
Descripción	Pestaña encargada de visualizar asociadas al conjunto global de datos
Activación	El usuario accede a la opción Metrics del menú lateral
Boceto	
Eventos	ir a “Explorer”, ir a “Dashboard Explorer” o ir a “Dashboard Routes”, seleccionar puerto, descargar gráfico

Tabla 8.5: DIU-05 Metrics

Capítulo 9

Implementación

En este capítulo se explicará cómo se ha implementado el sistema desarrollado a lo largo de este proyecto. Para ello se realizará una breve explicación sobre el desarrollo de la aplicación web y la arquitectura Big Data asociada a esta, así como de las diferentes herramientas y tecnologías que han sido utilizadas para el desarrollo del mismo.

9.1. Descripción del sistema

La herramienta desarrollada gira en torno a la captura de datos de navegación marítima a través de AISHub, explicada con anterioridad. En un inicio no se contemplaba el desarrollo de ninguna aplicación de visualización, *dashboard*, sin embargo con el fin de tratar de comprender mejor los datos marítimos captados se ha decidido implementar una aplicación web que permita su visualización. Esta decisión fue tomada una vez visto el gran volumen de información obtenido durante el mes en que se tuvo licencia. El hecho de contar con grandes volúmenes de datos, junto con el interés principal del proyecto de diseñar y desarrollar, aunque fuera en parte, un Data Lake para gestionar el tráfico marítimo, ha llevado al uso de novedosas herramientas y tecnologías Big Data. Por tanto la implementación de este proyecto cuenta con 3 partes totalmente diferenciadas, que se unen para componer el sistema final desarrollado.

En el inicio del proyecto, con el fin de poder captar la información ofertada por AISHub, se desarrolló una pequeña aplicación Java encargada de realizar conexiones cada 90 segundos a la API de AISHub, a través de una url, y descargar los datos disponibles en la misma en el momento de la conexión. Los datos obtenidos a través de esta aplicación se almacenan a diario en un servidor de la Universidad de Valladolid, siendo separados en función del día.

Una vez se dispone de los datos en crudo, se ha desarrollado una arquitectura Big Data en la que se trata de almacenar y procesar los datos asociados a la navegación marítima, así como la información asociada a los puertos, previamente explicada. Esta arquitectura Big Data es conocida como Data Lake y montada en un entorno Cloudera, utilizando Hive y Map Reduce con el fin de procesar la información disponible, además de HDFS, para almacenar los datos captados.

Paralelamente al desarrollo del Data Lake, anteriormente mencionado, se ha desarrollado una aplicación web, diseñada a modo de dashboard, con el objetivo de tratar de dotar de mayor valor a los datos almacenados. Esta aplicación se desarrolla de manera íntegra utilizando R, lenguaje de programación altamente orientado a la estadística, pero muy utilizado en la actualidad, junto con Python, para trabajar interactuando con sistemas Big Data. Esta aplicación

se conecta con el Data Lake, anteriormente mencionado, utilizando la librería *sparklyr*, la cual permite conectar aplicaciones desarrolladas en R con entornos Big Data. Las diversas librerías utilizadas para el desarrollo de esta aplicación serán descritas en la sección "Descripción del entorno".

El desarrollo de este sistema ha supuesto un gran reto, ya que, además de la utilización de gran cantidad de tecnologías y herramientas no utilizadas con anterioridad, se ha tenido que modificar el entorno Cloudera sobre el que se desarrolla, con el fin de poder desarrollar la aplicación web anteriormente descrita. A lo largo de este capítulo se explicará en mayor detalle el desarrollo de estas herramientas.

9.2. Descripción del entorno

Como se ha comentado en la sección anterior el proyecto se ha desarrollado utilizando un entorno Cloudera, más concretamente utilizando CDH 5.13 standalone. Este entorno utiliza como sistema operativo base un sistema CentOS 6.9, bifurcación de la distribución linux Red Hat Enterprise Linux RHEL. En este entorno nos encontramos con gran cantidad de herramientas Big Data, como Hive, Hue o Impala, ya instaladas, así como con la opción de acceder a Cloudera Manager, herramienta que permite gestionar los servicios ejecutados en el clúster con el que se trabaja. Sin embargo no se dispone de un entorno de programación R ya instalado, por ello para el desarrollo de la aplicación de visualización es necesario instalar gran cantidad de paquetes y herramientas, con el fin de poder desarrollar esta. La instalación de estas herramientas ha dado lugar a diferentes problemas derivados del sistema operativo utilizado y su versión, al estar anticuada.

La modificación de este entorno, mediante la instalación de paquetes y librerías, se divide en 3 fases, descritas a continuación:

1. Instalación del entorno R. Para llevar a cabo esto es necesario instalar R en el servidor Cloudera. Esto se realiza sin ningún problema ya que el paquete R se encuentra entre las dependencias del sistema, por lo que bastaría con acceder como superusuario por terminal y utilizar el siguiente comando: *sudo yum install R.*

Una vez lanzado el anterior comando es posible comenzar a desarrollar programas utilizando el lenguaje R. Sin embargo únicamente nos ofrece una interfaz por línea de comandos para la creación de los mismos. Por ello se decide instalar RStudio, IDE más utilizado para el desarrollo de aplicaciones en R. La instalación de este IDE da lugar a problemas, derivados de la no utilización de QT en el sistema operativo sobre el que se desea instalar. Este hecho imposibilita la instalación de dicho IDE en su versión de aplicación de escritorio, sin embargo es posible la instalación de su versión servidor, versión web. Para llevar a cabo dicha instalación se debe abrir el terminal y con permisos de superusuario seguir los siguientes 3 pasos:

- Descarga del programa a instalar: Para ello se debe utilizar el siguiente comando:

  wget https://download2.rstudio.org/rstudio-server-rhel-1.0.44-x86_64.rpm

- Instalación del paquete descargado: Para ello se debe utilizar el siguiente comando:

  sudo yum install –nogpgcheck rstudio-server-rhel-1.0.44-x86_64.rpm

- Comprobar si R-server se encuentra instalado (opcional): Para ello se debe ejecutar el siguiente comando:

  systemctl status rstudio-server.service

Una vez se ha realizado la instalación anteriormente descrita es posible desarrollar aplicaciones R utilizando el IDE RStudio-server, el cual facilita una interfaz amigable para el desarrollo de aplicaciones. Este IDE se divide en 4 pantallas principales, editor de texto, consola, entorno y gráficas, como se puede ver en la figura 9.1.

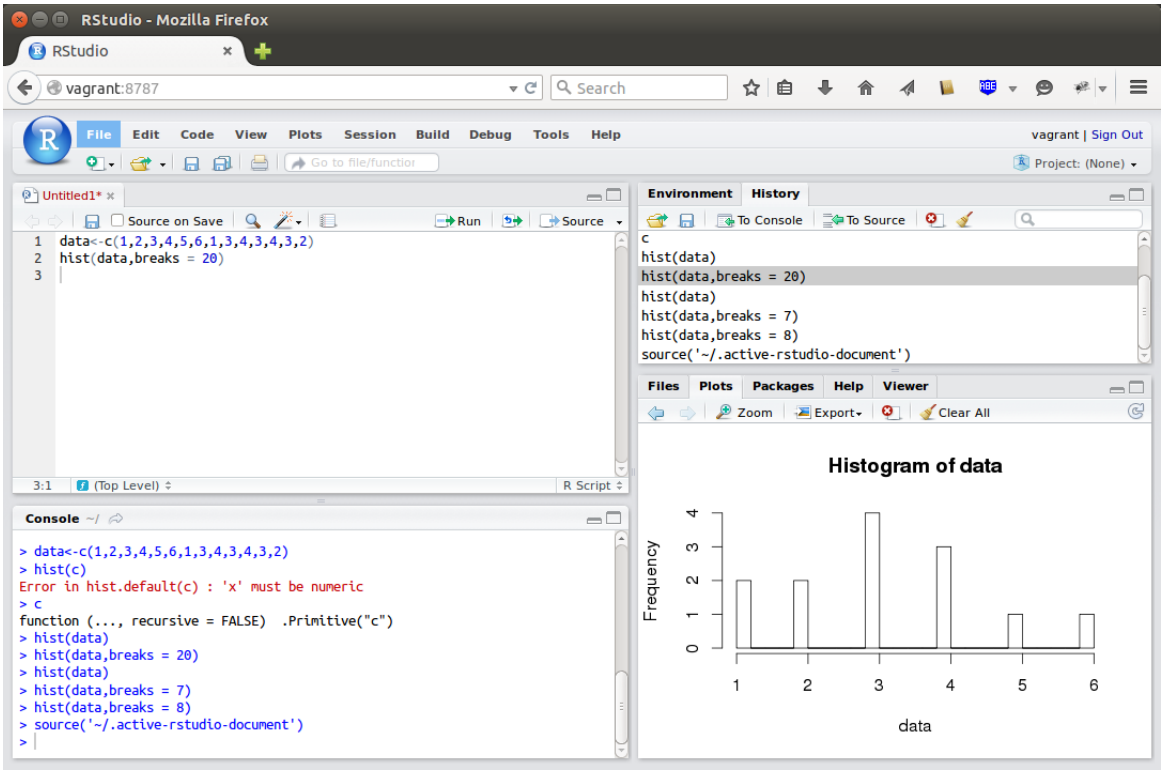


Figura 9.1: RStudio-Server

Una vez instalado el entorno de programación R es necesario instalar las diferentes librerías que se utilizarán para el desarrollo de este proyecto.

2. Instalación de librerías de R para el desarrollo de la web. R es un lenguaje que cuenta con una gran cantidad de librerías, gracias a las cuales es posible desarrollar gran cantidad de aplicaciones. Para el desarrollo de este proyecto es necesario desarrollar una aplicación web, algo posible de realizar utilizando la librería Shiny. Esta librería es muy utilizada al facilitar la creación de aplicaciones web interactivas sin necesidad de tener conocimientos de los lenguajes de programación web tradicionales. La instalación de esta librería es fundamental para este proyecto. En R es posible instalar de 2 maneras distintas nuevas librerías:

- Mediante la consola: Utilizando el siguiente comando *install.packages("shiny")*, en el caso de desear instalar el paquete anteriormente mencionado.

- Otra forma de instalar paquetes es buscarlos en el panel de control, en el que se ve una gráfica en la figura anterior. En ese panel de control es posible seleccionar paquetes para su uso así como instalar o actualizar paquetes.

Para el desarrollo de la aplicación web se han necesitado instalar gran cantidad de librerías, la mayoría debido a dependencias, destacando las siguientes:

- Leaflet: Gracias a este paquete es posible la gestión de mapas, algo de vital importancia en el proyecto. Para su instalación es necesario haber instalado con anterioridad el paquete "png". Leaflet es una de las librerías JavaScript más populares para gestionar mapas interactivos.
- DT: Librería con origen en JavaScript que permite la creación de tablas personalizables con gran facilidad.
- Leaflet.extras: Este paquete es una extensión del paquete original de Leaflet, el cual permite, entre otras cosas, la creación de mapas de calor.
- Shinydashboard: Paquete utilizado para crear una interfaz amigable para las aplicaciones desarrolladas utilizando shiny.
- shinycssloaders: Paquete utilizado para insertar spinners a la hora de cargar los datos necesarios para pintar las gráficas y los mapas de la aplicación.
- shinyjs: Paquete utilizado para poder insertar código CSS y JavaScript en la aplicación web.
- png: Paquete necesario para la instalación de Leaflet, al ser una dependencia de este. Este paquete da problemas en su instalación derivados de la no instalación del paquete libpng-devel, para solucionar este problema basta con abrir una terminal con permisos de superusuario y ejecutar el comando *yum install libpng-devel*.
- ggplot2: Librería especializada para la creación de gráficos. Será usada para crear métricas sobre los datos del proyecto.
- plotly: Librería encargada de mejorar el aspecto de las gráficas obtenidas utilizando ggplot2.

Uno de los problemas más complejos a los que nos hemos enfrentado en el desarrollo de la aplicación web es su conexión con Cloudera. A continuación se explicará la instalación de la librería utilizada para realizar dicha conexión, así como el código de conexión.

3. Instalación de librerías de R para la conexión e interacción con Cloudera. Para realizar la conexión entre R y Cloudera existen 2 paquetes, sparklyr e implyr. Una vez investigadas estas librerías se determinó que la instalación de sparklyr era más asequible, aún dando problemas para llevar a cabo su instalación, por lo que se descarta la instalación de implyr. Esta librería se instalaría siguiendo los métodos anteriormente vistos, sin embargo en su instalación hay problemas de dependencias. Para solventar estos problemas de dependencias se deben instalar los siguientes paquetes:

- curl: Este paquete da lugar a problemas en su instalación ya que para poder instalarlo en R es necesario tener instalado el paquete libcurl-devel en el sistema sobre el que corre, en este caso CentOS. Para instalar ese paquete es necesario abrir una terminal como superusuario y utilizar el comando *sudo yum install libcurl-devel*.

- xml2: La instalación de este paquete da lugar a problemas derivados de la no instalación del paquete libxml2-devel en el sistema operativo sobre el que corre R. Para solucionar este problema basta con instalar este paquete abriendo una terminal y ejecutando como superusuario el comando *sudo yum install libxml2-devel*.
- httr: La instalación de este paquete no da lugar a problemas, por lo que se podrá instalar de manera normal desde la consola de R.

Sin embargo con la instalación de las anteriores librerías no acaban los problemas de instalación del paquete sparklyr, ya que aparece el problema cuya solución más ha costado encontrar, la versión del paquete de gcc instalada, la cual necesita una actualización. La solución para este problema nos la ofrece Ken Takagiwa[26] a través un post en su cuenta de GitHub Gist. Esta solución consiste en abrir una terminal y seguir los siguientes pasos:

- Descargar el repositorio "devtools": Esto se puede conseguir ejecutando el siguiente comando:

  wget http://people.centos.org/tru/devtools-2/devtools-2.repo -O /etc/yum.repos.d/devtools-2.repo
- Instalación de los paquetes gcc y binutils: Esto se lleva a cabo ejecutando los siguientes comandos:

  sudo yum install devtoolset-2-gcc devtoolset-2-binutils
- Instalación de los paquetes gcc-c++ y gcc-fortran: Esto se lleva a cabo utilizando los siguientes comandos:

  sudo yum install devtoolset-2-gcc-c++ devtoolset-2-gcc-gfortran
- Comprobar la versión de gcc: Para ello se ejecutará el siguiente comando:

  /opt/rh/devtoolset-2/root/usr/bin/gcc –version
- Habilitar devtoolset-2 bash: Esto se consigue utilizando el siguiente comando:

  sudo scl enable devtoolset-2 bash
- Cambiar la fuente de la que se coge la versión de gcc:

  source /opt/rh/devtoolset-2/enable

Una vez completados todos los pasos anteriores ya sería posible realizar la instalación del paquete sparklyr. Una vez instalado existen varias formas de conexión, en función del clúster al que nos conectemos. En este paso hay que tener especial cuidado, ya que lo más común sería intentar realizar una conexión local, lo que derivaría en tener que lanzar el comando spark_install() en la consola de R. En caso de seguir esta opción la conexión con el entorno Cloudera se torna compleja, al dar problemas de versiones de Java, siendo este un error habitual. Por ello para el desarrollo de este proyecto se ha decidido utilizar una conexión de tipo cliente. A continuación se puede ver el código necesario para realizar la conexión.

```
sc <- spark_connect(master = "yarn-client",
                    spark_home = "/usr/lib/spark/",version = "1.6.0")
```

Una vez realizada la conexión es necesario instalar alguna librería que permita realizar consultas a este tipo de entornos. Con este objetivo destacan dos librerías principales, DBI y dplyr. Para el desarrollo de este proyecto se decidió utilizar la librería DBI, ya que ofrece la posibilidad de realizar consultas utilizando una sintaxis similar a SQL. Se puede instalar esta librería sin ningún problema desde la consola de R tal y como se ha descrito en el apartado anterior.

Una vez seguidos todos los pasos anteriormente descritos es posible realizar la implementación completa de la aplicación web desarrollada, ya que se encontraría completa la configuración del entorno.

9.3. Herramientas utilizadas

En esta sección se describirán las diferentes herramientas que se han utilizado en el desarrollo del proyecto.

Cloudera CDH 5.13. Plataforma de código abierto de Cloudera sobre la que se despliegan soluciones Hadoop, al incorporar su núcleo y diversos proyectos de la fundación Apache, como Hive, HBase, Mahout o Pig. Una de las características más destacadas de Cloudera es que cuenta con una interfaz gráfica propietaria, llamada Cloudera Manager,para la administración y gestión de los nodos y servicios del clúster. La versión de Cloudera utilizada en este proyecto cuenta con un clúster de un único nodo y está desplegada sobre un sistema operativo CentOS 6.9.

En definitiva Cloudera puede definirse como una solución ya desarrollada de Hadoop en la que se pueden desarrollar proyectos que necesiten el uso de tecnologías Big Data.

NetBeans. NetBeans es uno de los IDE de programación más utilizado para programar en lenguaje Java. Este IDE es multiplataforma, de código abierto y gratuito fundado por Sun Microsystems en el año 2000.

RStudio-Server. RStudio Server es un IDE de programación gratuito para R que incluye un editor de texto, una consola y herramientas para visualizar gráficas, instalar paquetes o debuguear aplicaciones, entre otras muchas opciones. Este IDE se ejecuta vía web y tiene capacidad para gestionar el consumo de memoria RAM de manera centralizada.

TeXLive. Distribución LaTeXmultiplataforma que cuenta con una serie amplia de librerías instaladas para el desarrollo de documentos LaTeX. Se ha utilizado para la realización de la memoria, junto con el IDE TeXstudio.

TeXstudio. Uno de los editores, IDE's, de mayor popularidad para el desarrollo de documentos LaTeX, al ser multiplataforma.

VersionOne. Herramienta online para la gestión de proyectos que sigan metodologías ágiles.

9.4. Tecnologías utilizadas

En esta sección se persigue describir las diferentes tecnologías y lenguajes de programación utilizados en el desarrollo de este proyecto.

R. Lenguaje de programación multiplataforma enfocado principalmente al análisis estadístico, muy utilizado en minería de datos. Este lenguaje se caracteriza por formar parte de un proyecto libre y colaborativo, lo que facilita el desarrollo de librerías y mejoras. De este hecho deriva la gran cantidad de librerías que existen para desarrollar una enorme variedad de tipos de programa. Es posible ver un ejemplo de la gran cantidad de librerías disponibles para desarrollar aplicaciones en la Sección 9.2. Se trata de un lenguaje interpretado muy utilizado en la actualidad en proyectos que necesiten el manejo de grandes volúmenes de datos, Big Data.

En definitiva R es un lenguaje en auge en la actualidad, tanto por su utilización en proyectos Big Data como por la gran actividad de la comunidad de R, permitiendo el crecimiento continuo de este lenguaje.

CSS. Lenguaje de "hojas de estilos" creado para controlar el aspecto de los documentos electrónicos definidos con HTML y XHTML, permitiendo separar así los contenidos y su presentación. Se considera que CSS es un lenguaje de marcado. CSS es uno de los 3 pilares del desarrollo Web cliente, junto con HTML y JavaScript.

JavaScript. Lenguaje interpretado que destaca por ser débilmente tipado y dinámico. Pese a ser utilizado principalmente en el lado cliente en la actualidad han surgido frameworks que permiten su uso para programar la parte servidor de las aplicaciones web, como Nodejs. En la actualidad JavaScript es uno de los lenguajes más utilizados de manera profesional, junto con la gran variedad de frameworks que derivan de él, como Angular, React, Vue.js o el ya mencionado Nodejs. JavaScript, como antes se ha mencionado, se considera uno de los 3 pilares del desarrollo web.

Java SE. Plataforma Java estándar utilizada para el desarrollo de pequeñas aplicaciones, al contar con un menor conjunto de API's, entre otros componentes.

A continuación se pasa a describir las tecnologías Big Data utilizadas en el desarrollo de este proyecto.

Apache HDFS. HDFS (Hadoop Distributed File System) es el software encargado del almacenamiento distribuido de datos en un clúster de servidores, especialmente creado para trabajar con grandes volúmenes de datos.

En este tipo de sistema de archivos los datos son divididos en pequeñas partes, denominadas bloques. El tamaño de los bloques es superior al habitual, lo que permite reducir el tiempo en los accesos para lectura de datos. Estos bloques se almacenan de forma distribuida a través de un clúster siguiendo el patrón "Write once read many", muy útil para almacenar grandes ficheros de logs que serán consultados en un gran número de veces. Esta forma de almacenamiento ayuda a que las funciones MapReduce,de las que hablaremos posteriormente, puedan ser ejecutadas sobre pequeños subconjuntos de bloques, alcanzando así una mayor escalabilidad.

Por tanto un fichero es almacenado en un conjunto de pequeños bloques, que a su vez son replicados en los distintos servidores de todo el clúster de Hadoop. Por defecto cada bloque

citado se ha replicado en 3 ocasiones, sin embargo se puede realizar un mayor número de réplicas.

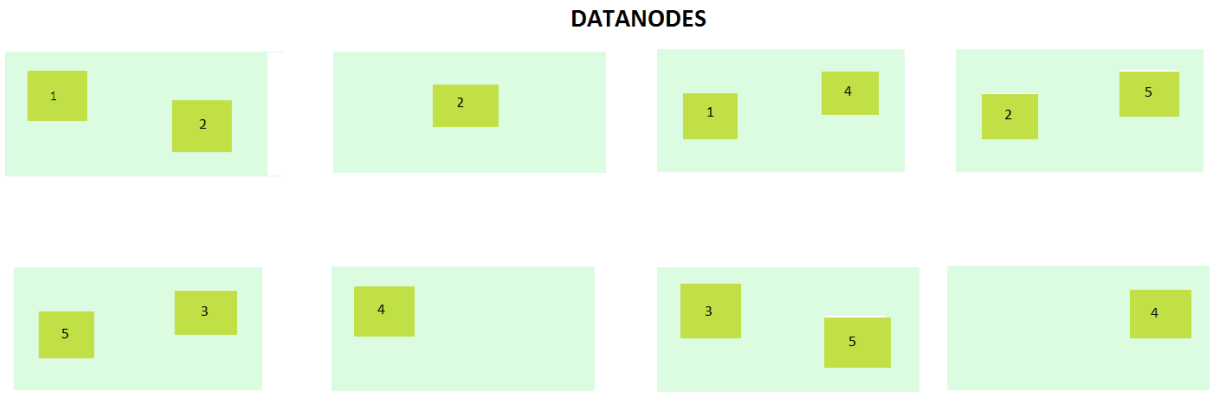


Figura 9.2: Replicación en clústers utilizando data nodes

HDFS sigue una arquitectura maestro/esclavo, en la que el nodo maestro es denominado NameNode y el resto de nodos se denominan DataNodes.

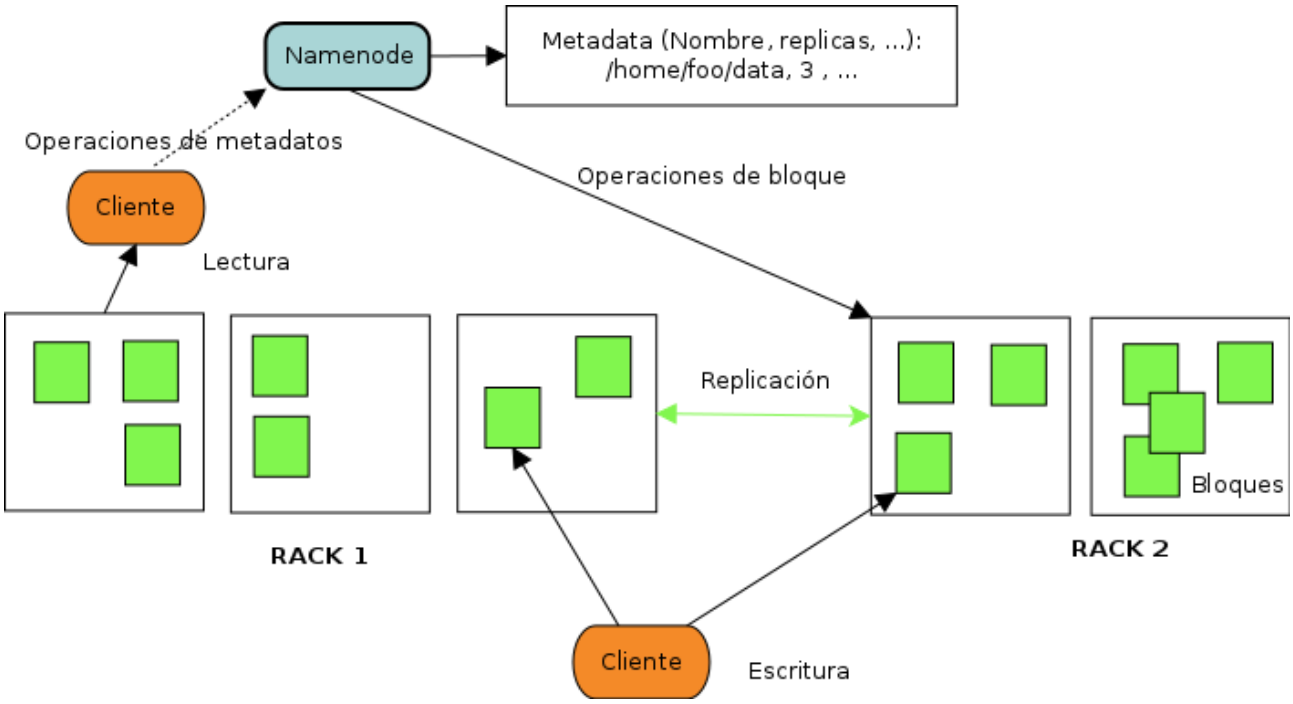


Figura 9.3: Gestión de acceso de los Name Nodes a los Data Nodes

Los NameNode son servidores que gestionan el acceso a los archivos almacenados en los DataNodes, para ello almacenan la información de qué bloques forman un archivo y donde se localizan (los metadatos de los archivos). Además de esto los NameNode ejecutan operaciones del sistema de fichero, tales como la apertura, el cierre o el renombrado de ficheros y directorios, así como el mapeado de los bloques en DataNodes. En cambio los DataNodes son los responsables de servir los requerimientos de lectura y escritura de los clientes del sistema de ficheros, además realiza la creación, eliminado y replicación de los bloques siguiendo las instrucciones ofertadas por el NameNode.

Map Reduce. MapReduce es un paradigma de programación utilizado para dar soporte a la computación paralela sobre grandes colecciones de datos en clústers de servicores. Este paradigma es usado para la resolución práctica de algoritmos susceptibles de ser paralelizados. Normalmente son utilizados para abordar problemas con grandes volúmenes de datos suelen ejecutarse en HDFS.

Gracias a MapReduce es posible ofrecer una escalabilidad enorme a través de cientos o miles de servidores de Apache Hadoop, estando formado por la operativa de las funciones Map () y Reduce (). Ambas funciones se ejecutan de forma distribuida en diversas máquinas.

La función Map() tiene como objetivo la transformación de una entrada de datos, por filas de clave valor, en una salida clave/valor, como se puede ver en el ejemplo 9.4.1.

Ejemplo 9.4.1. Tenemos un fichero con varias filas, con datos de ciudades y su temperatura registrada. El resultado de realizar Map() sobre los datos de este fichero de ejemplo sería: (Toronto, 20) (Whitby, 25) (New York, 22) (Rome, 32), (toronto,4) (Rome, 33) (New York, 18)

Figura 9.4: Datos sobre los que trabajaremos

La función Reduce() es aplicada en paralelo para cada grupo de valores, produciendo una colección de valores para cada dominio, es decir, toma todos los valores de una clave específica y genera una nueva lista reducida como salida (Reduce (k2, list(v2)) ->list(v3). Esto se puede ver en el ejemplo 9.4.2.

Ejemplo 9.4.2. Sobre la lista anterior, realizamos una reducción para quedarnos solo con las temperaturas máximas de cada ciudad. El resultado de realizar Reduce() sobre las tuplas de datos obtenidas al realizar la operación Map() es: (Toronto, 20) (Whitby, 25) (New York, 22) (Rome, 33).

Apache Hive. Apache Hive surge como una tecnología de tipo data warehousing que permite gestionar grandes volúmenes de datos almacenados en HDFS, así como en otros tipos de almacenamiento como HBase. Hive dispone de un lenguaje de consulta declarativo, llamado HiveQL, muy similar a SQL. Se suele considerar que Hive es una abstracción de alto nivel de Map Reduce, ya que su motor de ejecución se encarga de traducir las consultas HQL en configuraciones de jobs Map Reduce, ejecutados directamente sobre la infraestructura Hadoop.

El hecho de utilizar una sintaxis similar a SQL y su forma de trabajo hace que se asemeje a los sistemas de bases de datos tradicionales. Sin embargo Hive se encuentra pensado para entornos de data warehouse (OLAP), en los que los datos suelen ser estáticos y se demanda llevar a a cabo tareas analíticas de manera exhaustiva, siendo lentas las consultas en caso de compararlo con los sistemas tradicionales.

En definitiva Hive es altamente utilizado para implementar ETL's, construir informes u obtener análisis específicos. Es considerada una de las principales tecnologías Big Data para el procesamiento y análisis de grandes volúmenes de información. En la siguiente figura es posible ver la posición que ocupa Hive en el ecosistema Hadoop.

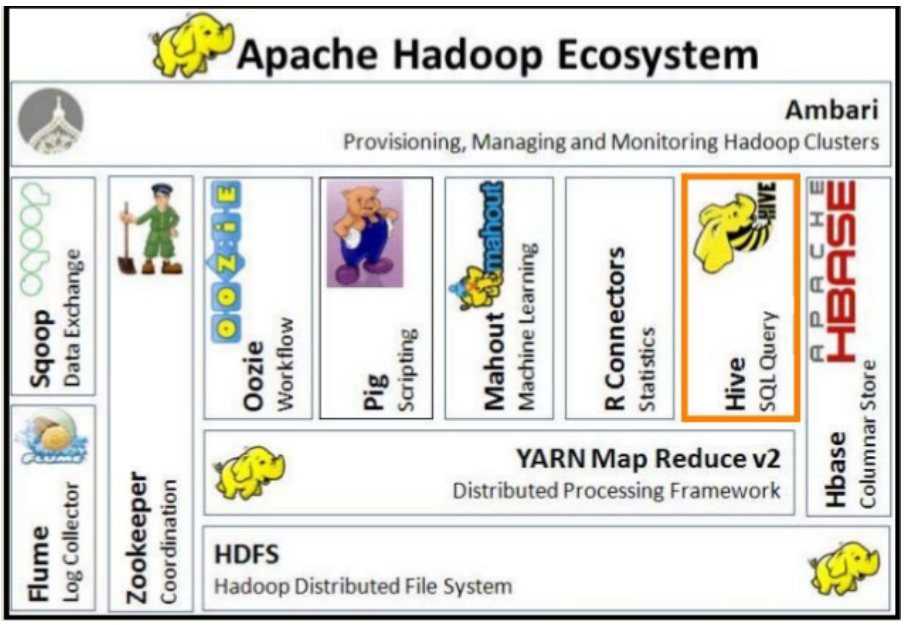


Figura 9.5: Hive en el ecosistema hadoop

Apache Oozie. Tecnología encargada de gestionar y planificar la ejecución de pipelines complejos a la hora de construir ETL's. Oozie por tanto permite diseñar y ejecutar workflows de procesamiento de datos utilizando tecnologías Hadoop principalmente. Algunas de las características principales de esta tecnología son:

- Está desarrollado para operar sobre el modelo de computación que propone Hadoop, por lo que simplifica todas aquellas cuestiones necesarias para la ejecución de jobs, facilitando su adopción.
- Es capaz de resolver fácilmente los problemas que puedan suceder durante la ejecución de los jobs, recuperándose de forma efectiva en caso de errores.
- Es extensible, dando soporte a un número creciente de tipos de acciones requeridas en los jobs.
- Es escalable y fiable, permitiendo la ejecución concurrente de múltiples jobs.
- Plantea un servicio multi-tenant que ayuda a reducir el coste de operación.

9.5. Implementación del programa de captación de datos

Con el fin de poder descargar toda la información que ofrece AISHub, en el plazo en el que se tuvo acceso a una licencia, se decidió desarrollar una aplicación Java. La elección del lenguaje Java para el desarrollo de esta radica en la mayor experiencia del desarrollador en dicho lenguaje, unido a la necesidad de comenzar a captar datos en el menor tiempo posible. Esta necesidad descartó otras opciones, como Flume, debido a la curva de aprendizaje que se necesitaría y la falta de tiempo.

Para la descarga de datos se accedió a una licencia gratuita, gracias al hecho de ser estudiante. Para la obtención de la licencia era necesario comunicarles la IP de la máquina a través de la cual se realizarán las conexiones. La conexión a la API de AISHub se realiza a través de una url personalizada, en función de los datos de la licencia. A continuación se muestra la url genérica y se explicarán las diferentes opciones existentes para montar la url personalizada.

Ejemplo URL:

http://data.aishub.net/ws.php?username=A&format=B&output=C&compress=D&
latmin=E&latmax=F&lonmin=G&lonmax=H&mmsi=I&imo=J

Los elementos que componen la url son:

- username: Nombre de usuario indicado en la licencia obtenida
- format: Formato de los datos, puede ser de dos tipos, en formato de lectura para humanos (1), o con codificación AIS (0). Para este proyecto se eligió el formato para humanos.
- output: Formato de salida de los datos, se puede elegir entre csv, xml o json. Para este proyecto se eligió el formato de salida csv.
- compress: Campo que permite elegir la compresión, o no compresión, de los datos al ser descargados. En caso de desear que los ficheros se descarguen comprimidos se permiten 3 formatos, ZIP (1), GZIP (2) y BZIP (3). En nuestro caso se elige la opción de descargar sin comprimir, por lo que a la url en ese atributo se le dará el valor 0.

Existe otra serie de parámetros adicionales que permiten limitar los datos a descargar, como las latitudes y longitudes máximas y mínimas. Sin embargo se ha decidido no utilizar estas opciones. A continuación se podrá ver la url definitiva utilizada para la conexión.

http://data.aishub.net/ws.php?username=AH_TRIAL_Q3AZ&format=1&output=csv&compress=0

Una vez se encuentra disponible la licencia y es posible conectar con la API, a través de la url anteriormente mencionada, se puede empezar a descargar información. AISHub informa en su web de una gran limitación a la hora de descargar datos a través de su API. Esta consiste en que en caso de conectarse a descargar datos de la API más de una vez por minuto es posible que no se descargue ninguna información. Debido a esta limitación se ha decidido crear una aplicación multihilo, en la que se crea un hilo que cada 90 segundos se conecta a la API de AISHub y descarga la información. La aplicación creada se desarrolla en 2 clases diferentes, a continuación es posible ver el código de creación de dicha aplicación.

Clase Main. Clase encargada de instanciar el objeto hilo, el cual se encarga de realizar las conexiones a la API y de descargar la información que esta contenga.

Clase Hilo. Clase encargada de realizar las conexiones a la API, a través de la url previamente explicada, así como de descargar los datos disponibles en el momento de cada conexión, bit a bit, en formato csv. Los ficheros se almacenarán con la fecha, en milisegundos, del momento en que se realiza la descarga. En esta clase cabe destacar el uso de un nuevo hilo de ejecución, el cual se ejecutará cada 90 segundos para realizar la conexión. A continuación se mostrarán algunos de los aspectos más destacados de dicha clase.

- Esta clase implementa la interfaz Runnable, con el fin de permitir crear un hilo para la descarga de datos. A continuación se muestra como se implementa esta interfaz y los métodos iniciales asociados a la creación de un hilo.

```
public class Hilo implements Runnable {

        Thread hilo;
        public static Hilo instancia;

        public static Hilo getInstance() {
                if (instancia == null) {
                        instancia = new Hilo();
                }else{
                        System.out.println("El programa se esta
                        ejecutando ya");
                }
                return instancia;
        }

        public void start() {
                if (hilo == null) {
                        hilo = new Thread(this);
                        hilo.start();
                }
        }
```

- Metodo *run*: En este método se realiza la conexión y descarga de datos, durante 90 segundos. A continuación veremos el codigo referente a la conexión con la API.

```
        //Conexion con Internet
        URL prueba =
        new URL("http://data.aishub.net/ws.php?
        username=AH_TRIAL_Q3AZ&      format=1&output=csv&compress=0");

        prueba.openConnection().connect();
```

  Una vez el programa se conecta con la API de AISHub es posible descargar la información disponible en la misma. A continuación se puede ver el código de descarga desarrollado, perteneciente al mismo método

```
String folder = "../Descargas/";
//Carpeta de destino de la descarga
//Para poner nombre al archivo
Date fecha = new Date();
long fecha1 = fecha.getTime();
String nombre;
nombre = Objects.toString(fecha1, null);
//Creo el archivo. La extension seria .csv
File archivo = new File(folder +
nombre + ".csv");

InputStream entrada = prueba.openStream();
OutputStream salida =
new FileOutputStream(archivo);
int b = 0;

while (b != -1) {
       b = entrada.read();
       if (b != -1) {
              salida.write(b);
       }
}
salida.close();
entrada.close();
```

Es posible ver el código completo del programa en el CD del proyecto.

Con el fin de mantener el programa en ejecución en segundo plano en el servidor donde se captan los datos es necesario utilizar el comando "nohup". Este comando, disponible en entornos Unix, permite mantener un servicio ejecutándose pese a cambiar de sesión en la máquina. Para lanzar el programa por tanto se creó un fichero con extensión .jar del mismo y un script en bash en el que se lanza el programa utilizando el comando anteriormente mencionado. A continuación es posible ver el script mencionado.

```
#!/bin/bash
nohup java -jar Recolector.jar
```

Una vez se ha implementado lo anteriormente descrito ya es posible descargar los datos ofertados por AISHub. A continuación se explicarán el procesamiento de los datos, así como la creación de una base de datos con estos y el desarrollo de una aplicación web que utilice dichos datos.

9.6. Implementación del Data Lake

En esta sección se persigue explicar la implementación de los diferentes programas y scripts utilizados para la implementación del Data Lake. La implementación del Data Lake se hace utilizando 4 tecnologías, previamente explicadas, HDFS, Hive, Map Reduce y Oozie. Con el objetivo de no hacer demasiado tediosa la explicación de esta sección se explicará lo que se ha implementado con cada tecnología, el motivo de su implementación y un breve ejemplo,

en código, de cada una de ellas. A continuación se explicarán los diferentes procesamientos realizados sobre los datos, en función de la tecnología utilizada.

Map Reduce. La fase inicial de procesamiento de los datos se ha realizado utilizando Map Reduce. Con este motivo se ha creado un programa utilizando este paradigma de programación, previamente explicado, con el fin de conseguir obtener los identificadores de trayectoria y mensaje. A continuación se detallará el funcionamiento de las diversas clases que componen el programa, así como los métodos más interesantes de estas.

- *Clase Identification*: Esta clase se encuentra compuesta por los métodos estáticos map y reduce, así como por el método main encargado de la ejecución del programa. En el método "map" se eliminan las comillas y los espacios de los campos que los tienen en los ficheros y tras esto se generan pares clave valor y se envían los valores de los ficheros al método "reduce". A continuación se puede ver el código asociado a dicho método.

```
public static class MessageMapper extends Mapper<Object, Text,
ETLCompositeKey, Text>
{
        private ETLCompositeKey mid = new ETLCompositeKey();

                @Override
                public void map(Object key, Text value, Context context)
                 throws IOException, InterruptedException {
                        String line =
                         value.toString().replaceAll("\"", "");
                        String[] fields = line.split(",");

                        for(int i=0;i<fields.length;i++){
                                fields[i] =
                                        fields[i].replaceAll("\"", "");
                                fields[i] = fields[i].trim();
                        }

                        String mmsi = fields[0];
                        String fecha = fields[1];
                        try {
                          fecha =
                          L0Message.parseTime(fecha, false).toString();
                        } catch (ParseException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        };

                   mid.setMmsi(new Text(mmsi));
                   mid.setTime(new LongWritable(Long.valueOf(fecha)));
                   context.write(mid, new Text(line));
                }
```

```
        }
```

Tras esto se envían estos objetos al método reduce, en el cual se procesan todos los mensajes con el fin de generar el identificador de trayectoria y de objeto, a continuación se puede ver el código de dicho método.

```
public static class MessageReducer extends Reducer<ETLCompositeKey,
Text,NullWritable, Text>
{
        @Override
        public void reduce(ETLCompositeKey key, Iterable<Text> values,
         Context context) throws IOException, InterruptedException {

                ArrayList<LOMessage> arrMsg = new ArrayList<LOMessage>();

                String idMsg;
                // This loop iterates over all messages sent by the
                 current vessel.
                for (Text value : values) {
                        String line = value.toString();
                        try {
                                LOMessage current =
                                        new LOMessage(line.split
                                        (LOMessage.COMMA_I, -1));
                                idMsg = current.getMmsi()+
                                        "-"+current.getTime();
                                current.setId(new Text(idMsg));
                                arrMsg.add(current);
                        } catch (ParseException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                        }
                }

                Collections.sort(arrMsg, LOMessage.MsgComparator);
                for(int i=0;i<arrMsg.size();i++){
                        LOMessage tmp = arrMsg.get(i);
                        String idTrj = arrMsg.get(0).getMmsi()+
                        "-"+arrMsg.get(0).getAuxFecha().toString();
                        tmp.setIdTrj(new Text(idTrj));
                        context.write(NullWritable.get(), tmp.dump());
                }
        }
}
```

- Clase ETLCompositeKey: Clase encargada de generar claves compuestas por el MMSI y el timestamp de cada mensaje.

- Clase LOMessage: Clase encargada de instanciar los diferentes atributos que componen los mensajes que se reciben en los ficheros, además de los atributos nuevos que se quieren generar. Es utilizada esta clase en el método "reduce" anteriormente visto con el fin de crear una lista de mensajes a partir de la cual se pueda acceder a la fecha de la primera vez que un navío emitió un mensaje. El método más destacado de esta clase es el siguiente.

```
public static Comparator<LOMessage> MsgComparator = new
 Comparator<LOMessage>() {
       public int compare(LOMessage msg1, LOMessage msg2) {

              Text t1 = msg1.getTime();
              Text t2 = msg2.getTime();
              return t1.compareTo(t2);

       }
};
```

- Clase SecondarySorting: Esta clase se encarga de ordenar la ejecución de los programas sobre los ficheros que se encuentran a lo largo del clúster.

Oozie. Se ha utilizado Oozie con el fin de poder ejecutar el programa Map Reduce anteriormente explicado. Sin embargo debido a un problema en los ficheros, relativo a la existencia de cabeceras en los mismos y la imposibilidad de Map Reduce para procesarlas, ha llevado a realizar un procesamiento previo de los ficheros antes de ejecutar el script Oozie desarrollado. Utilizando el siguiente comando es posible eliminar la primera fila de todos los ficheros de extensión ".csv" del directorio que se quiera procesar: *sed -i '1d'*.csv*

Una vez procesados estos datos será posible cargarlos en el servidor, con ese fin se ha creado una carpeta genérica, llamada db, en la que hay dos subdirectorios, raw y ident, en los que se almacenan los datos en formato original (raw) y ya procesados, tras lanzar el programa Map Reduce, ident. Estos datos se almacenan en HDFS y el script Oozie creado es el siguiente.

```
<workflow-app name="identification" xmlns="uri:oozie:workflow:0.5">
       <start to="ident"/>
       <action name="ident">
              <map-reduce>
                     <job-tracker>${jobTracker}</job-tracker>
                     <name-node>${nameNode}</name-node>
                     <prepare>
                            <delete path="/user/cloudera/db/ident/
                            dia1/"/>
                     </prepare>
                     <configuration>
                            <property>
                                   <name>mapred.mapper.new-api</name>
                                   <value>true</value>
                            </property>
                            <property>
```

```
        <name>mapred.reducer.new-api</name>
        <value>true</value>
</property>
<property>
        <name>mapreduce.job.map.class</name>
        <value>
          barco.Identification$MessageMapper
        </value>
</property>
<property>
        <name>
          mapreduce.job.partitioner.class
        </name>
        <value>
         barco.SecondarySorting$
                MessagePartitioner
        </value>
</property>
<property>
        <name>
         mapreduce.job.output.group
         .comparator.class
        </name>
        <value>
         barco.SecondarySorting
                $MessageGroupingComparator
        </value>
</property>
<property>
        <name>
          mapreduce.job.reduce.class
        </name>
        <value>
         barco.Identification$MessageReducer
        </value>
</property>
<property>
        <name>
          mapreduce.job.output.key.class
        </name>
        <value>barco.ETLCompositeKey</value>
</property>
<property>
        <name>
         mapreduce.map.output.value.class
        </name>
```

```
                                <value>org.apache.hadoop.io.Text
                                </value>
                        </property>
                        <property>
                                <name>
                                 mapreduce.job.output.value.class
                                </name>
                                <value>org.apache.hadoop.io.Text
                                </value>
                        </property>
                        <property>
                                <name>
                                 mapreduce.input.
                                  fileinputformat.inputdir
                                </name>
                                <value>
                                 /user/cloudera/db/raw/dia1/
                                </value>
                        </property>
                        <property>
                                <name>
                                 mapreduce.output
                                  .fileoutputformat.outputdir
                                </name>
                                <value>
                                 /user/cloudera/db/ident/dia1/
                                </value>
                        </property>
                </configuration>
        </map-reduce>
        <ok to="success"/>
        <error to="fail"/>
    </action>
    <kill name="fail">
        <message>
        Action failed, error
         message[${wf:errorMessage(wf:lastErrorNode())}]</message>
    </kill>
    <end name="success"/>
</workflow-app>
```

Como se puede ver en este script se llama a las diferentes clases que componen el programa Map Reduce y se le pasa a este un directorio de entrada, así como la ruta a un directorio de salida donde almacenar la información procesada.

Hive. Mediante Hive se persigue realizar las siguientes fases de procesamiento sobre la información, desde un nivel más elevado de abstracción. A continuación se mostrará un ejemplo de

script en Hive que permite crear una tabla a través de una consulta realizada sobre otra tabla, pudiendo así insertar los datos ya procesados.

```
create table vessel1
stored as textfile
as
select mmsi,imo,
       callsign,name,
       case when type between 0 and 99 then type else null end as type,
       case when a = 0 then null else a end as a,
       case when b = 0 then null else b end as b,
       case when c = 0 then null else c end as c,
       case when d = 0 then null else d end as d,
       case when draught = 0.0 then null else draught end as draught
from vessel
```

Hive, además de para procesar y limpiar datos, ha sido utilizado también con el fin de obtener métricas y dotar de valor a la información almacenada en el Data Lake. A continuación se muestra un pequeño script en hive que permite obtener la distribución de navíos por nacionalidad en cada puerto de destino.

```
select o.dest, o.type, count(*) as c
from(
       select i.mmsi,i.type,i.dest
       from(
              select v.mmsi, v.type, t.dest
              from vesselprueba2 as v, trajectoryprueba1 as t
              where v.mmsi = t.mmsi
       )i
       group by i.mmsi,i.type,i.dest
)o
where istnotnull(o.type) and isnotnull(o.dest)
group by o.dest, o.type
order by c desc
```

9.7. Implementación de la aplicación web

Para el desarrollo de la aplicación web se ha decidido utilizar el lenguaje de programación R. Este lenguaje es muy utilizado en la actualidad con el fin de crear a gran velocidad aplicaciones de visualización que sirvan de apoyo al análisis de grandes volúmenes de información. La creación de páginas web en R es muy sencilla al contar con una librería, llamada shiny. Esta librería facilita la creación de páginas web interactivas, gracias en parte al concepto de programación reactiva ligada a esta librería, no siendo necesario tener ningún conocimiento previo de los lenguajes web tradicionales como HTML, CSS o JavaScript. Se considera programación reactiva a la programación orientada al manejo de steams de datos asíncronos y a la propagación del cambio. El desarrollo de una aplicación shiny se divide en 2 partes, la interfaz de usuario (UI) y la parte servidor (server).

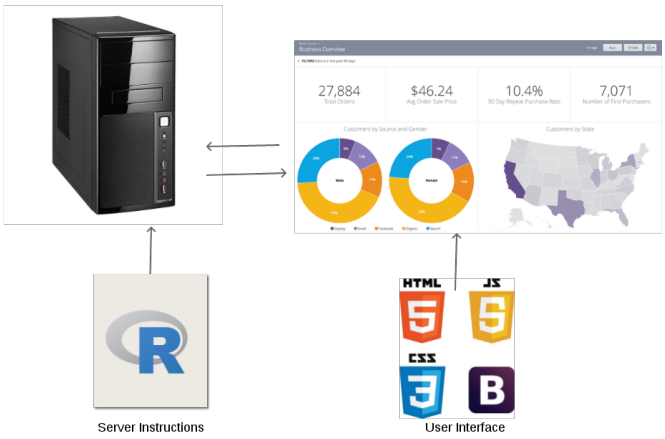


Figura 9.6: Arquitectura Shiny

Para desarrollar este tipo de aplicaciones es necesario instalar la librería shiny, algo que como hemos visto anteriormente se consigue desde la consola de R utilizando el siguiente comando:

install.packages("shiny")

Una vez instalado shiny la estructura básica de una aplicación utilizándolo, sin empezar a desarrollar sería:

```
library(shiny)

# Define UI for application that draws a histogram
ui <- fluidPage()

# Define server logic required to draw a histogram
server <- function(input, output) {}

# Run the application
shinyApp(ui = ui, server = server)
```

A continuación se explicarán las diferentes partes que componen una aplicación desarrollada utilzando shiny.

Interfaz de Usuario. Esta función/fichero permite el desarrollar la interfaz de usuario de una aplicación web. Es posible insertar código HTML, CSS y JavaScript, sin embargo es mucho más eficaz utilizar librerías de R que facilitan funciones desarrolladas en dichos lenguajes. En definitiva shiny genera la parte front de la aplicación transformando el código R generado a HTML, CSS y JavaScript, para permitir su visualización en web desde este archivo. Para el desarrollo de la interfaz de usuario es muy utilizada la librería shinydashboard, la cual ofrece gran cantidad de posibilidades de personalización, así como elementos JavaScript, CSS y HTML

prediseñados. A continuación se pueden ver los diferentes componentes asociados a la interfaz de usuario de una aplicación que utilice esta librería.

```
library(shiny)
library(shinydashboard)

ui <- dashboardPage(
       dashboardHeader(), #Cabecera de la aplicacion y nombre
       dashboardSidebar(), #Menu lateral
       dashboardBody() #Cuerpo central de la aplicacion
)
```

Una vez identificados los diferentes elementos que componen la parte visual de nuestra aplicación se realizará un pequeño ejemplo de implementación de cada uno, empezando por el header.

```
       ui <- dashboardPage(
              skin = "blue", #Cambia los colores del tema de la aplicacion

              dashboardHeader(title = "Maritime Traffic"),
```

A continuación se puede ver el código con el que se implementará el menú lateral de nuestra aplicación.

```
       #loading window
       appCSS_sidebar <- "
       #loading-content-sidebar {
              position: absolute;
              background: #000000;
              opacity: 0.9;
              z-index: 100;
              left: 0;
              right: 0;
              height: 100%;
              text-align: center;
              color: #FFFFFF;
       }
       "
       ## Sidebar content ################
              dashboardSidebar(
              useShinyjs(),
              inlineCSS(appCSS_sidebar),
              # Loading message
              div(
              id = "loading-content-sidebar"
              ),
              sidebarMenu(id = "tabs",
                     menuItem("Explorer",
                     tabName = "Explorer", icon = icon("th")),
```

```
                menuItem("Dashboard Explorer",
                tabName = "DashboardExplorer",
                icon = icon("dashboard")),
                menuItem("Dashboard Routes",
                tabName = "DashboardRoutes",
                icon = icon("dashboard")),
                menuItem("Metrics", tabName = "Metrics",
                icon = icon("bar-chart-o")),

        #Condicion para variar los input laterales
                conditionalPanel(
                condition = "input.tabs === 'DashboardExplorer'",
                        selectInput("loc",label = "Vessel type",
                        choices = types),
                        dateInput("dates", label = "Date")
                ),
                conditionalPanel(
                condition = "input.tabs == 'DashboardRoutes'",
                        selectInput("loc1",label = "Vessel type",
                                choices = types),
                        uiOutput("names")
                )))
```

Gracias a este código es posible visualizar el menú lateral de la aplicación, el cual varía en función de la opción en la que uno se encuentre.

Por último se podrá ver un ejemplo sobre el elemento dashboardBody. Este elemento se puede dividir en columnas y filas, al igual que en las aplicaciones web tradicionales, con el fin de ordenar los contenidos. A continuación se podrá ver un ejemplo de su uso en nuestra plataforma, más concretamente el asociado a la opción de visualizar rutas.

```
dashboardBody(
        useShinyjs(),
        tabItem(tabName = "DashboardRoutes",
                fluidRow(column(
                        width = 12, leafletOutput("routesmap") %>%
                        withSpinner(type= 6, color="#0dc5c1")
                )),br(),
                fluidRow(

                        box(width = 6, title = "Speed Over Ground",
                                status = "primary",
                                solidHeader = TRUE, collapsible = TRUE,
                                plotlyOutput("plotSOG")
                                %>% withSpinner(type= 1, color="#0dc5c1")),
                        box(width = 6, title = "Course Over Ground",
                                status = "primary",
                                solidHeader = TRUE, collapsible = TRUE,
                                plotlyOutput("plotCOG")
```

```
                        %>% withSpinner(type= 1, color="#0dc5c1"))
                    )
)))
```

En este ejemplo la función "leafletOutput" permite pintar un mapa con la funcionalidad descrita en el server. El identificador que se pasa a la función leafletOutput debe ser igual al nombre dado a la función en el servidor, en este caso esta sería *output$routesmap*.

Servidor. Las instrucciones desarrolladas en esta función/fichero se encuentran orientadas a dotar de funcionalidad a la página web desarrollada. En este fichero se realizan las conexiones a bases de datos, así como el trabajo con los datos que se desean visualizar en la interfaz de usuario. A continuación se puede ver el ejemplo del código a través del cual se genera el mapa cuya salida por pantalla se ha descrito con anterioridad.

```
output$routesmap <- renderLeaflet(
       {
       if(is.na(input$names)){

       }else{
              dataset1 <- dbGetQuery(sc,paste("select t.mmsi,v.name,
              m.latitude,m.longitude
              from trajectoryprueba1 as t, vesselprueba2 as v,
               messageprueba1 as m
              where t.mmsi=v.mmsi and t.id_trajectory = m.id_trajectory
               and v.type=\"",input$loc1, "\" and name=\"",
              input$names,"\"", sep = "", collapse = NULL))

              m1 <- leaflet(dataset1,
                     options = leafletOptions(minZoom = 2)) % > %
              addTiles(group = "OSM (default)") % > %
              addProviderTiles(providers$Esri, group = "Esri") % > %
              addProviderTiles(providers$CartoDB, group = "Carto") % > %
              addCircleMarkers(
              radius = 6,
              lng = dataset1$longitude,
              lat = dataset1$latitude,
              label = dataset1$name,

              popup = ~ htmlEscape(dataset1$name),
              group = "dataset"
              )%>%
              addPolylines(lng = dataset1$longitude,
              lat = dataset1$latitude)% > %
              addLayersControl(baseGroups = c("OSM (Default",
              "Esri", "Carto")
              )
       }
}
```

Esta es una de las funciones más importantes desarrolladas en este proyecto y permite visualizar las trayectorias del navío seleccionado previamente. Para llevar a cabo esta función se han utilizado 2 librerías, descritas en la Sección 9.2, llamadas Leaflet y DBI. Esta función obtiene la latitud y longitud de un navío a lo largo de la trayectoria que realiza y con ellas se añaden marcadores al mapa, permitiendo así visualizar dichas trayectorias. Además se permite cambiar el tipo de mapa utilizado de manera dinámica.

Para desarrollar la parte servidor se han utilizado gran variedad de librerías, además de las ya mencionadas, entre las que destacan sparklyr y DT, descritas también en la Sección 9.2. Es posible ver el código completo de la aplicación en el CD del proyecto entregado.

Capítulo 10

Métricas

En este capítulo se persigue realizar un análisis completo de los datos ya almacenados, tratando de dotar a estos de valor. Para ello se han definido una serie de métricas que, unidas a la cantidad de datos existentes, nos permitan conocer de primera mano algunos puntos del tráfico marítimo que nos eran desconocidos hasta el momento.

A lo largo de este capítulo se propondrán y describirán una serie de métricas, tras lo que se explicará su creación, basada principalmente en su interés, así como los resultados obtenidos de aplicar dichas métricas.

10.1. Definición de las métricas

En esta sección se pretende describir en detalle las diferentes métricas que se han planteado para el análisis de los datos previamente almacenados. A continuación se podrán ver las diferentes métricas que se han desarrollado en la fase inicial de este proyecto.

M.01. Puertos de destino más frecuentes. Mediante esta métrica se persigue conocer los destinos más frecuentes, es decir, los destinos a los que llega un mayor número de navíos. Gracias a esta métrica es posible tener una mayor información sobre los puertos, pudiendo así conocer los de mayor uso y analizar así la información relativa a estos.

M.02. Tiempo promedio de demora de llegada de un buque a puerto. Mediante esta métrica se persigue analizar los posibles retrasos existentes en la llegada de un navío a puerto. Esta métrica es de gran utilidad a futuro para este proyecto, ya que, en el caso de contar con información sobre el precio y la carga de los navíos, se podría conocer la fluctuación en el precio del producto a entregar en función del tiempo de demora del navío. Pese al planteamiento de esta métrica en esta fase inicial del proyecto es imposible de desarrollar, al no disponer de la capacidad de almacenamiento y procesamiento necesarias para llevarla a cabo.

M.03. Distribución por tipo de navío en cada puerto. Mediante esta métrica se persigue conocer el número de navíos que hay de cada tipo en cada puerto, pudiendo así conocer el tipo de comercio en el que se especializa cada puerto, ya que el tipo de buque se encuentra altamente relacionado con su carga. Aunque para ello sería mejor contar con una información detallada sobre la carga de cada navío, la cual se podría obtener a través de una licencia de pago.

M.04. Distribución de banderas de los navío por puerto. Mediante esta métrica se persigue conocer los navíos de qué países atracan en cada puerto, permitiendo así conocer en parte las relaciones comerciales internacionales existentes.

M.05. Variación en los destinos a lo largo del viaje. Mediante esta métrica se persigue conocer si existen variaciones en cuanto al destino del navío una vez este ha salido de puerto, comprobando así la existencia de escalas a lo largo de una trayectoria. La información derivada de la creación de esta métrica no será muy descriptiva en esta fase, debido a las limitaciones presentes en los datos y la tecnología con las que se trabaja en la actualidad.

M.06. Número de naves por tipo. Mediante esta métrica se puede conocer la cantidad total de navíos por tipo de los que se dispone información.

M.07. Número de naves por nacionalidad. Mediante esta métrica se puede conocer la cantidad total de navíos por nación de los que se dispone información.

10.2. Resultados

En esta sección se persigue mostrar los diferentes resultados obtenidos una vez se han desarrollado las métricas anteriormente descritas. Es importante destacar que el valor actual asociado a la realización de las anteriores métricas es menor del esperado, ya que la falta de recursos para trabajar con el conjunto global de datos imposibilita dotar de mayor valor a estas métricas. Sin embargo el desarrollo de estas métricas sobre un mayor conjunto de información, en fases posteriores del proyecto, no tendría diferencia en cuanto a su implementación. A continuación se explicarán los resultados obtenidos de dichas métricas.

M.01. Puertos de destino más frecuentes. Gracias al desarrollo de esta métrica se ha podido comprobar que el puerto de destino más conocido es el puerto de Rotterdam (Holanda), el cual es el puerto de mayor tamaño de Europa y uno de los más grandes a nivel mundial. El hecho de que este sea el puerto de destino más frecuente dentro de nuestra aplicación se puede derivar de la mayor cantidad de estaciones receptoras de AISHub en Europa. A continuación, en la figura 10.1, se podrán ver los puertos más frecuentes, así como el número de navíos que tienen estos como destino. Se puede obtener más información accediendo a la aplicación.

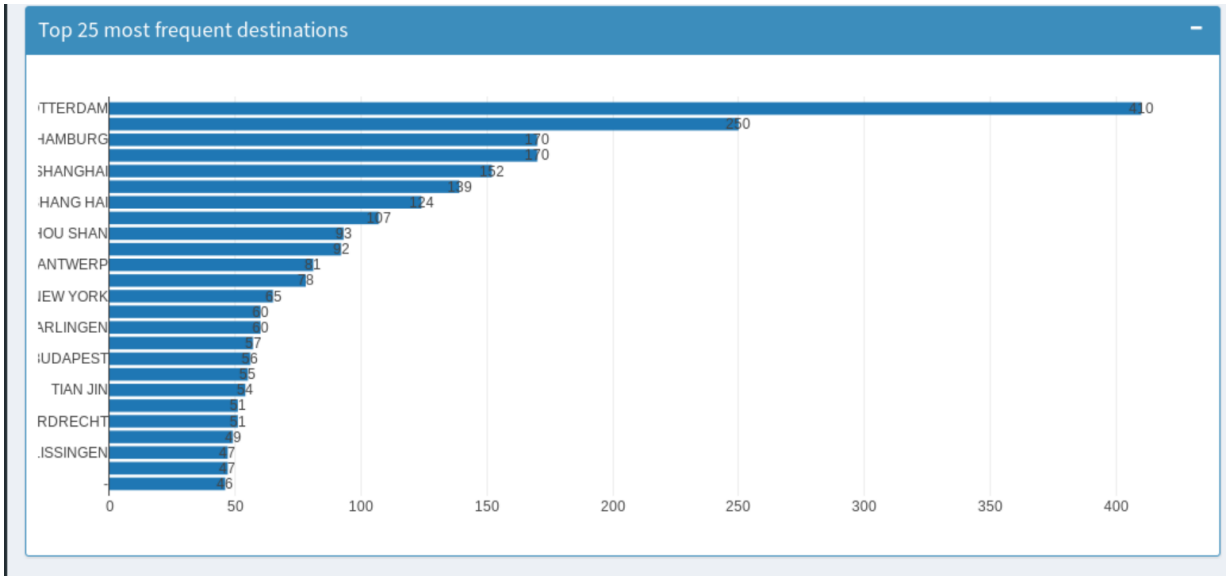


Figura 10.1: Top 25 de destinos más frecuentes

M.02. Tiempo promedio de llegada a puerto. Como se ha comentado en la explicación de dicha métrica ha sido imposible ejecutarla, debido a los reducidos recursos con los que se cuenta en la actualidad.

M.03. Distribución por tipo de navío en cada puerto. Gracias a esta métrica es posible acceder al número de navíos por tipo en cada navío, conociendo así la especialidad de cada puerto. Como ejemplo, la figura 10.2 muestra el número de navíos por tipo en el puerto de Rotterdam.

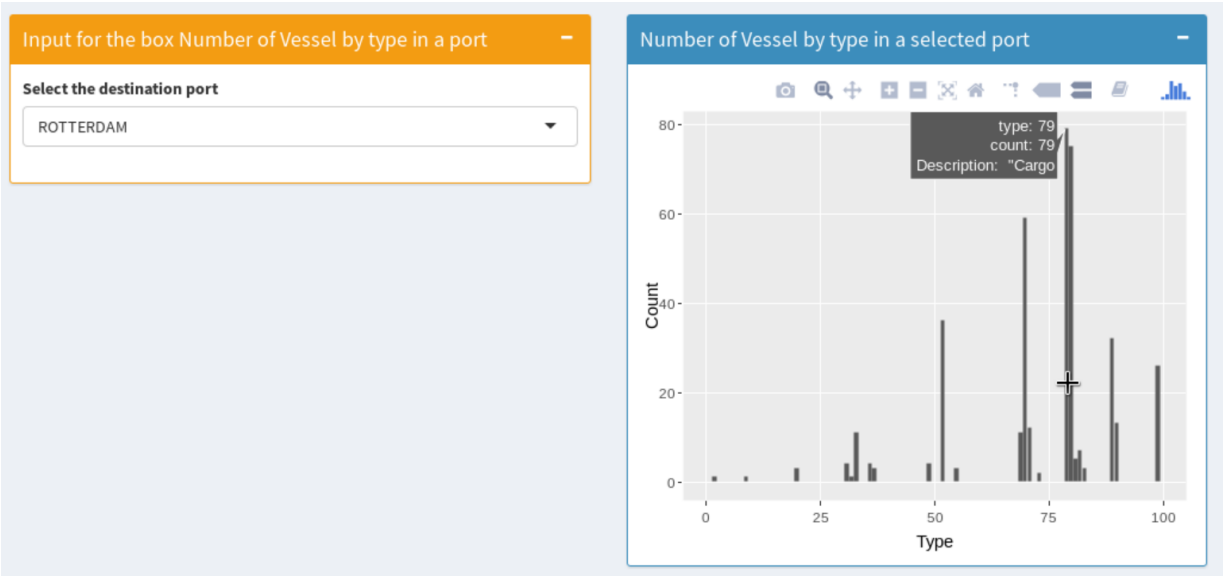


Figura 10.2: Ejemplo métrica distribución por tipo de navío en el puerto de Rotterdam

M.04. Distribución de banderas de los navío por puerto. Esta métrica es muy similar a la anterior y permite conocer las relaciones internacionales existentes.Como ejemplo, la figura 10.3 muestra la distribución de navíos por nacionalidad en el puerto de Rotterdam.

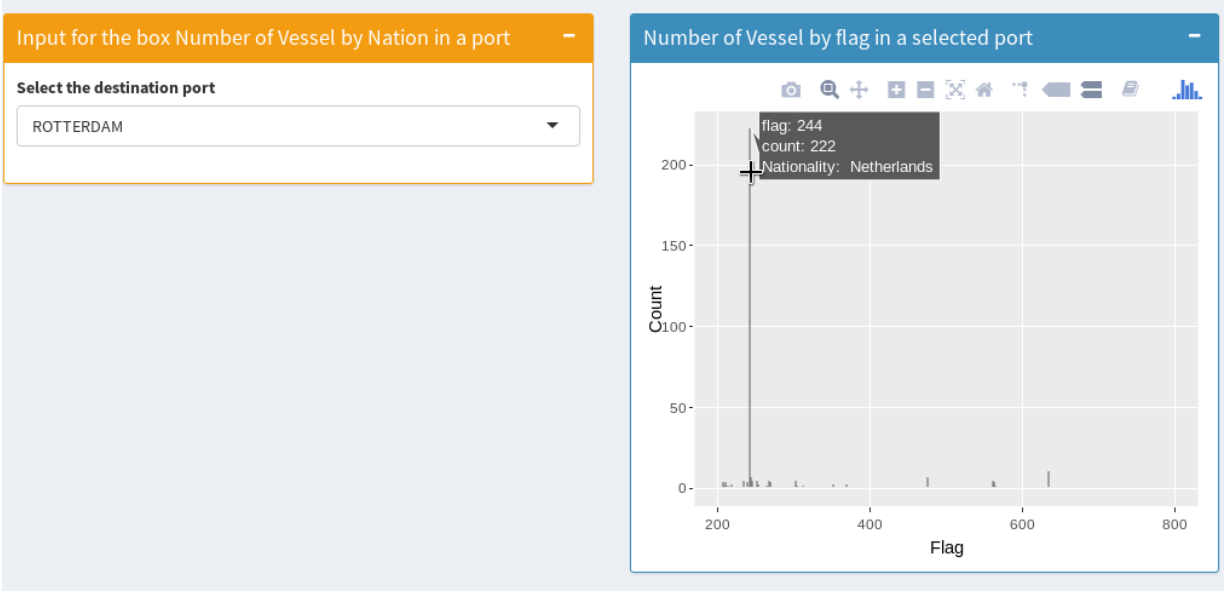


Figura 10.3: Ejemplo métrica distribución por nacionalidad de navío en el puerto de Rotterdam

M.05. Variación en los destinos a lo largo del viaje. Esta métrica sí que ha sido desarrollada pese a las limitaciones existentes en los datos. Estas limitaciones disminuyen el valor de dicha métrica, ya que al no poder cargar grandes volúmenes de información únicamente se dispone de fragmentos pequeños de la trayectoria en los que no se ve que la trayectoria varíe. Para el desarrollo de dicha métrica se ha decidido ordenar los navíos por el identificador principal y mostrar el puerto de destino asociado a este navío. Al agrupar por identificador de navío y destino es posible ver los diferentes destinos existentes para cada navío. A continuación se mostrará una pequeña tabla con el resultado de dicha métrica.

MMSI	Destino
789456123	ILHA GRRANDE
725019920	LAKE CHARLES LOUISIA
725017900	QUINTERO
725017900	NULL
725003670	PTO.CORONEL

Tabla 10.1: Métrica 05. Variación en los destinos

Pese a no contar con la trayectoria completa de un navío es posible ver que existen variaciones en los destinos de los navíos, al comprobar que tras realizar la consulta que trate de satisfacer esta métrica se obtiene un número de registros superior al número de navíos sobre los que se realiza el seguimiento. Al realizar un análisis más exhaustivo sobre los resultados obtenidos se ha podido ver que en varias ocasiones el destino de un mismo navío cambia a Null, normalmente debido a su llegada a puerto, como se puede ver en la tabla 10.1.

M.06. Número de naves por tipo. Gracias a esta métrica se ha podido conocer que los cargueros son el tipo de navío más captado. A continuación se podrá ver una gráfica asociada al número de navíos de cada tipo almacenado.

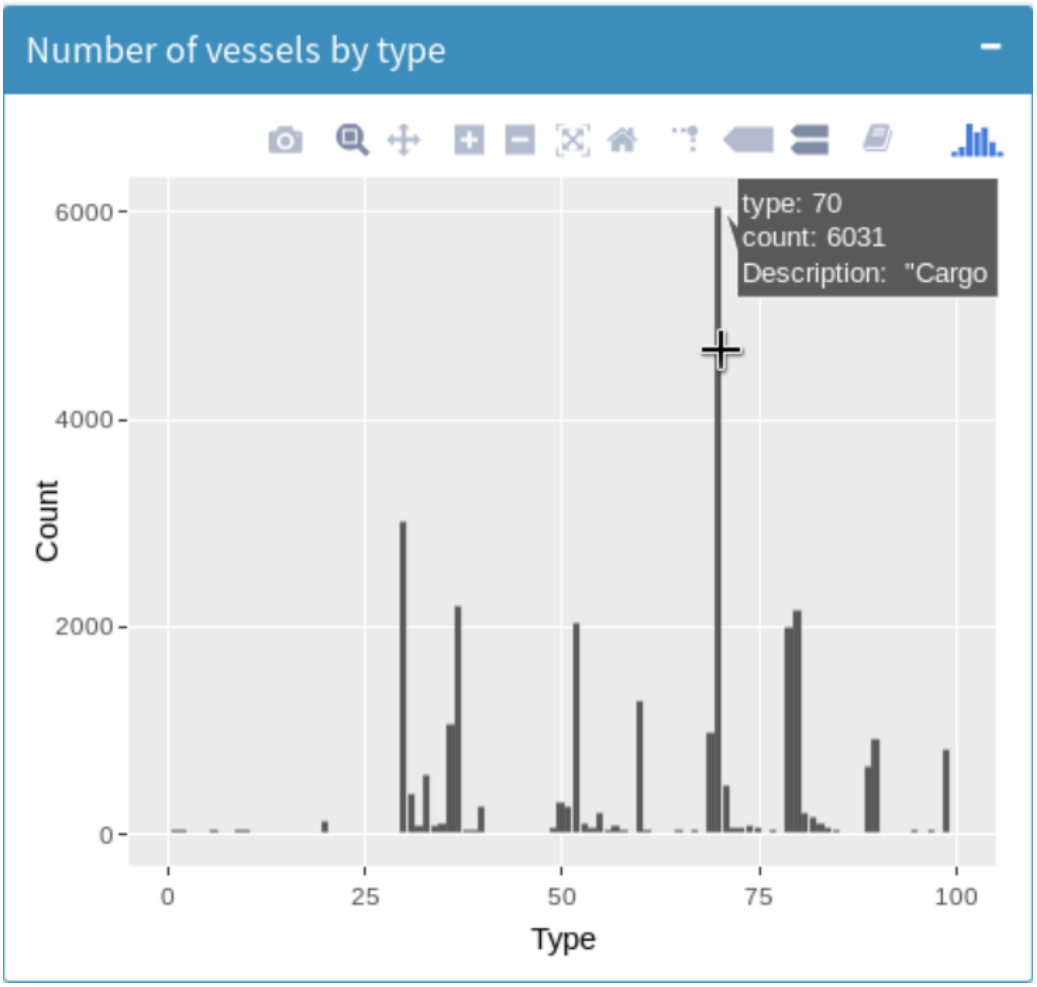


Figura 10.4: Número de navíos por tipo

M.07. Número de naves por nacionalidad. Mediante esta métrica ha podido conocer que el país con mayor número de navíos captados es Holanda, seguido de China. A continuación se podrá ver una gráfica asociada al número de navíos de cada nacionalidad almacenada.

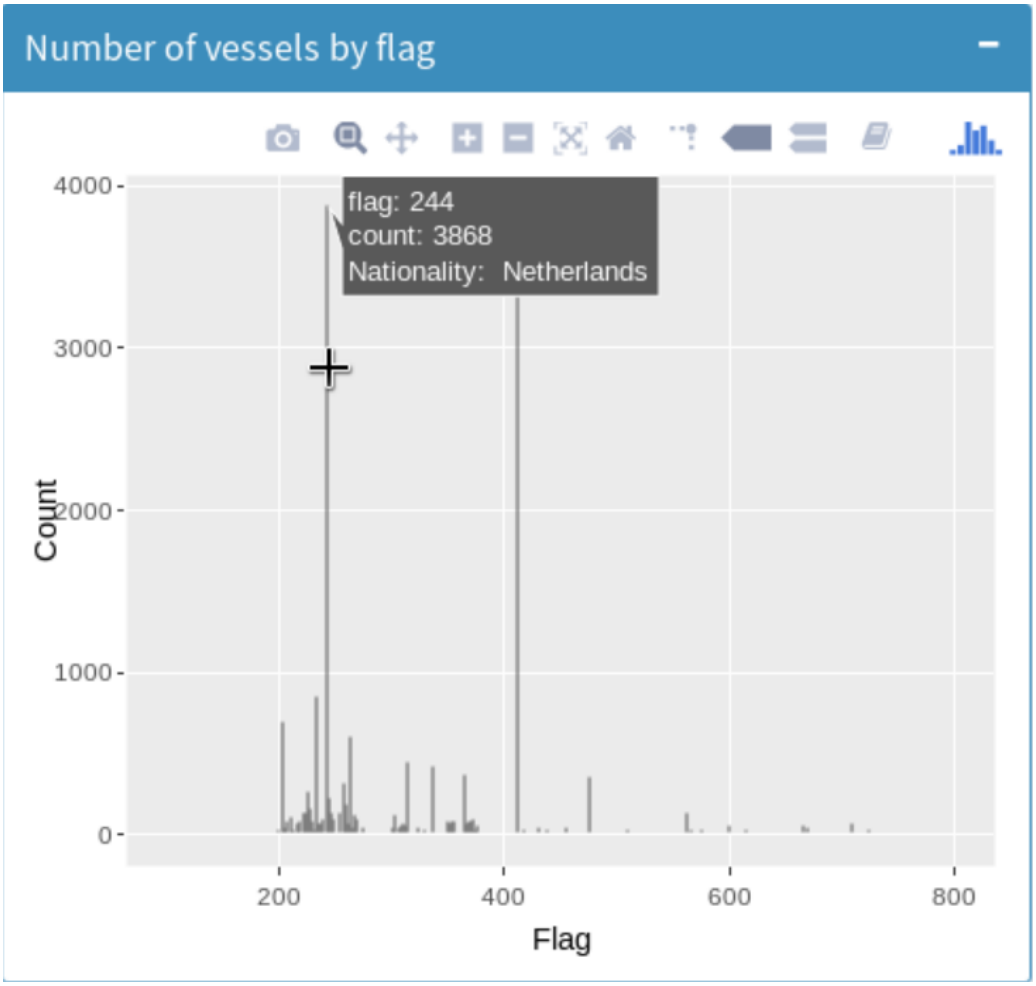


Figura 10.5: Número de navíos por nacionalidad

En caso de desear acceder a una información más detallada sobre estas métricas se puede utilizar la herramienta desarrollada, la cual ofrece diversas opciones de visualización.

Capítulo 11

Pruebas

En este capítulo se reflejarán las diferentes pruebas realizadas a lo largo del desarrollo del proyecto. En este capítulo por tanto se mostrarán los resultados de las pruebas realizadas, en forma de tabla.

Existen diferentes tipos de pruebas sobre sistemas software. Sin embargo en este proyecto únicamente se han realizado pruebas de caja negra, las cuales son pruebas realizadas sobre el funcionamiento individual de cada componente del sistema, sin tomar en cuenta la estructura interna del código o detalles de implementación.

A continuación se describirán las diferentes pruebas de caja negra realizadas a lo largo del proyecto.

11.1. Pruebas de caja negra

En esta sección se describirán las diversas pruebas de caja negra realizadas en el proyecto, así como los resultados de estas.

PCN-01	**Listado de navíos**
Propósito	Visualizar un listado de la información estática asociada a los diferentes navíos de la herramienta
Prerrequisito	Ninguno
Datos de entrada	Ninguno
Pasos	1. El usuario selecciona la opción “Explorer”
Resultado esperado	El sistema muestra una tabla con la información estática asociada a los diferentes navíos de la plataforma
Resultado obtenido	Se muestra una tabla con los datos de los diferentes navíos de la plataforma
Resultado de la prueba	Correcto

Tabla 11.1: Prueba de caja negra 01, listado de navíos

PCN-02	Buscar navíos
Propósito	Buscar los datos deseados, con independencia del atributo, en el listado de navíos anterior
Prerrequisito	Tener valores en la tabla
Datos de entrada	Cualquier atributo de la tabla
Pasos	1. El usuario selecciona la opción “Explorer”
Resultado esperado	El sistema muestra los datos obtenidos a través de la búsqueda
Resultado obtenido	El sistema muestra los datos obtenidos a través de la búsqueda
Resultado de la prueba	Correcto

Tabla 11.2: Prueba de caja negra 02, buscar navíos

PCN-03	Visualizar mapa de navíos
Propósito	Visualizar en un mapa el tráfico marítimo mundial en un momento determinado
Prerrequisito	Debe haber información sobre los navíos
Datos de entrada	Tipo de navío y fecha
Pasos	1. El usuario selecciona la opción “Dashboard Explorer” 2. El usuario indica el tipo de navío cuyo tráfico desea analizar 3. El usuario indica el momento en el que se desea analizar el tráfico marítimo
Resultado esperado	El sistema muestra un mapa con los navíos que cumplan las condiciones de tipo y fecha indicadas
Resultado obtenido	El sistema muestra un mapa con los navíos que cumplan las condiciones de tipo y fecha indicadas
Resultado de la prueba	Correcto

Tabla 11.3: Prueba de caja negra 03, visualizar mapa de navíos de un tipo en un momento determinado

PCN-04	Visualizar mapa de calor
Propósito	Visualizar un mapa de calor con los navíos de un tipo determinado y en un momento de tiempo especificado
Prerrequisito	Debe haber información sobre los navíos
Datos de entrada	Tipo de navío y fecha
Pasos	1. El usuario selecciona la opción “Dashboard Explorer”, y dentro de esta la opción “Heatmap” 2. El usuario indica el tipo de navío cuyo tráfico desea analizar 3. El usuario indica el momento en el que se desea analizar el tráfico marítimo
Resultado esperado	El sistema muestra un mapa de calor con los navíos que cumplan las condiciones de tipo y fecha indicadas
Resultado obtenido	El sistema muestra un mapa de calor con los navíos que cumplan las condiciones de tipo y fecha indicadas
Resultado de la prueba	Correcto

Tabla 11.4: Prueba de caja negra 04, visualizar un mapa de calor de los navíos de un tipo en un momento determinado

PCN-05	Visualizar trayectoria de un navío
Propósito	Visualizar en un mapa la trayectoria seguida por un navío indicado
Prerrequisito	Debe haber información sobre los navíos
Datos de entrada	Tipo de navío y nombre
Pasos	1. El usuario selecciona la opción “Dashboard Routes” 2. El usuario indica el tipo de navío sobre el que filtrar 3. El usuario indica el nombre del navío cuya trayectoria se desea analizar
Resultado esperado	El sistema muestra un mapa con la trayectoria seguida por el navío seleccionado
Resultado obtenido	El sistema muestra un mapa con la trayectoria seguida por el navío seleccionado
Resultado de la prueba	Correcto

Tabla 11.5: Prueba de caja negra 05, visualizar trayectorias

PCN-06	Visualizar estadísticas asociadas a la trayectoria de un navío
Propósito	Visualizar gráficos asociados a la trayectoria visualizada
Prerrequisito	Debe haber información sobre los navíos
Datos de entrada	Tipo de navío y nombre
Pasos	1. El usuario selecciona la opción “Dashboard Routes” 2. El usuario indica el tipo de navío sobre el que filtrar 3. El usuario indica el nombre del navío cuya trayectoria se desea analizar
Resultado esperado	El sistema muestra 2 gráficos asociados a la trayectoria seguida por el navío seleccionado
Resultado obtenido	El sistema muestra 2 gráficos asociados a la trayectoria seguida por el navío seleccionado
Resultado de la prueba	Correcto

Tabla 11.6: Prueba de caja negra 06, visualizar estadísticas asociadas a la trayectoria

PCN-07	Visualizar métricas asociadas al conjunto global de datos del navío
Propósito	Visualizar gráficos asociados conjunto total de datos disponibles
Prerrequisito	Debe haber información sobre los navíos
Datos de entrada	Tipo de navío y nombre
Pasos	1. El usuario selecciona la opción “Metrics”
Resultado esperado	El sistema muestra los diferentes gráficos asociados a las métricas explicadas en el capítulo 10
Resultado obtenido	El sistema muestra los diferentes gráficos asociados a las métricas explicadas en el capítulo 10
Resultado de la prueba	Correcto

Tabla 11.7: Prueba de caja negra 07, visualizar métricas asociadas a los datos globales del navío

Capítulo 12

Manuales

En este capítulo se persigue desarrollar el manual de usuario de la aplicación web desarrollada. Normalmente este manual suele estar acompañado por un manual de instalación, sin embargo al haber desarrollado el apartado de instalación de los diferentes componentes necesarios para desarrollar esta aplicación, en la Sección 2 del Capítulo 9, se ha decidido desarrollar únicamente el manual de usuario.

12.1. Manual de Usuario

En esta sección se detallará el funcionamiento de la aplicación web desarrollada, con el fin de poder ayudar a los usuarios que la utilicen en caso de dudas en el uso de la misma. La herramienta es accesible a través de un navegador web, por lo que los usuarios no necesitan instalar nada, siendo la configuración del servidor trabajo de los desarrolladores. Es importante destacar que esta aplicación es únicamente usada a modo de visualización por lo que por el momento se ha decidido que no es necesario disponer de ningún sistema de registro y login para visualizar esta información. Esta aplicación se encuentra dividida en cuatro apartados, los cuales se detallan a continuación.

Explorer. Pantalla inicial de la aplicación. En esta ventana se puede ver un listado de la información estática de los diferentes navíos de la plataforma, pudiendo buscar la información que se desee a través del buscador, independientemente del atributo por el que se desea buscar. Además de esto es posible desplazarse a través del listado utilizando un paginador, así como seleccionar el número de navíos que se visualicen en la tabla. A continuación se puede ver un ejemplo de esta ventana de navegación.

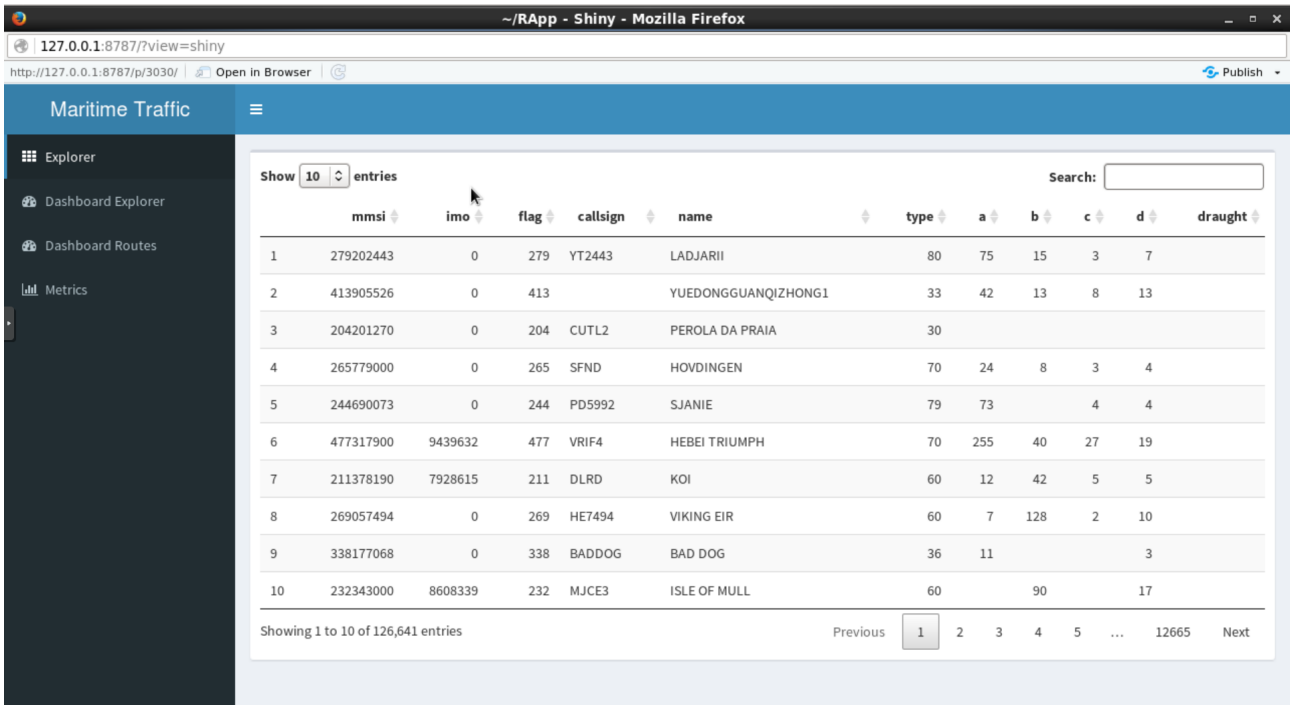


Figura 12.1: Página "Explorer" de la plataforma desarrollada

Como se ha visto en el Capítulo 8, más concretamente en la sección "Diseño de Interfaces", desde esta ventana inicial se podrá navegar al resto de pestañas de la aplicación, utilizando para ello el menú lateral, sin necesidad de registro alguno en la plataforma.

Dashboard Explorer. En esta pantalla es posible visualizar el tráfico marítimo, a nivel mundial, de todos los navíos de un tipo y en un momento determinado. Estos filtrados se realizan utilizando el menú lateral, el cual cambiará en el momento de acceso a dicha pestaña. Los filtros anteriormente mencionados se encuentran enlazados, ya que se mostraran las diferentes fechas y horas en las que se puede ver el tráfico de los diferentes navíos que pertenecen al tipo seleccionado. En esta pestaña se puede ver un ejemplo de programación reactiva, ya que la visualización de los datos cambia en función de los filtrados sin necesidad de recargar la página. Esta visualización se realizará mediante 2 tipos de mapas. En la aplicación es posible ver el tráfico marítimo en un momento determinado, teniendo la posibilidad de identificar los navíos con un label, como se puede ver en la figura 12.2. Por otro lado se puede ver un mapa de calor, en el que en función del color se puede ver la concentración de navíos en un momento determinado, como se puede ver en la figura 12.3. A continuación se podrán ver esta pestaña con los diferentes mapas de visualización. Cabe destacar la posibilidad de cambiar la capa, "layer", de los mapas a mostrar, ofertando diversas opciones.

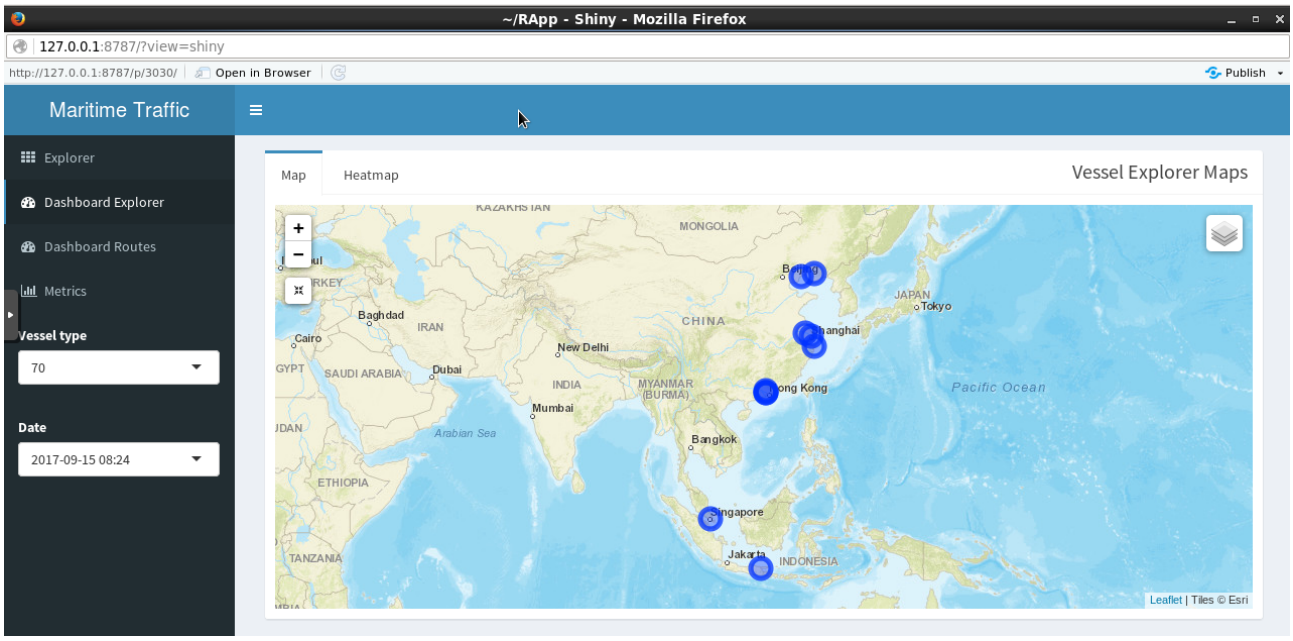


Figura 12.2: Página “Dashboard Explorer” utilizando el mapa

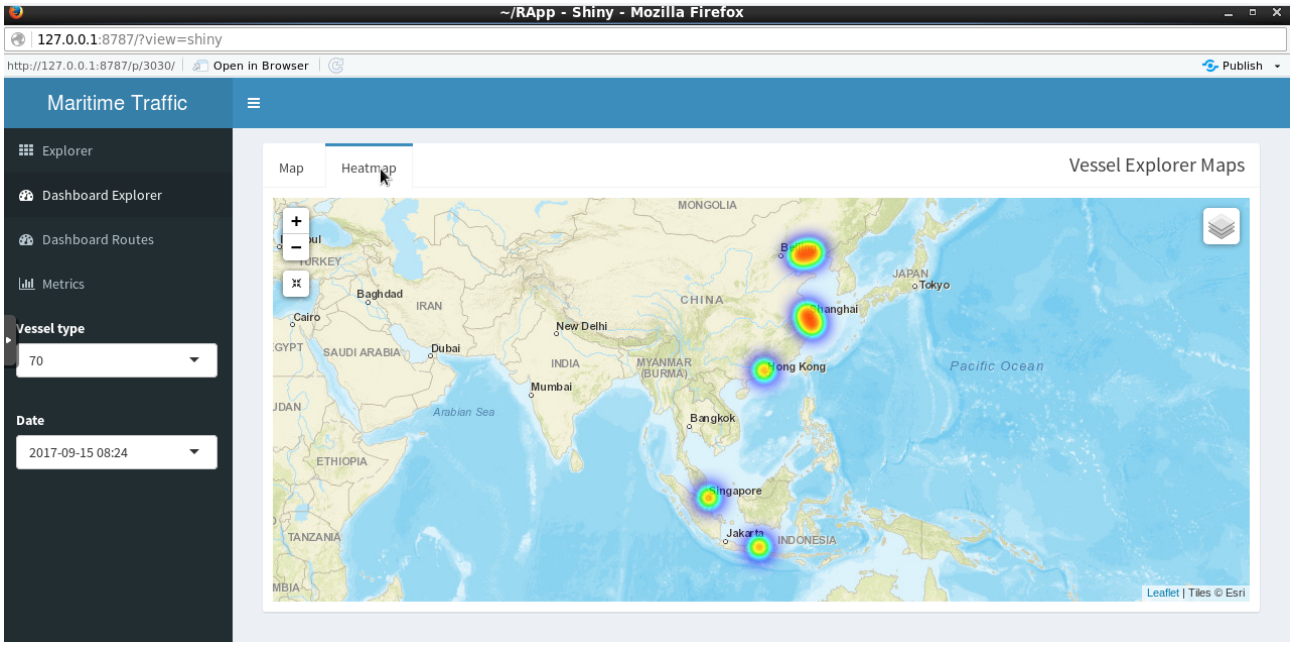


Figura 12.3: Página “Dashboard Explorer” utilizando el heatmap

Dashboard Routes. En esta pantalla se persigue visualizar la ruta realizada por un navío. Con el objetivo de visualizar los datos de este navío se realizan dos filtrados en el menú lateral, por tipo y por nombre. Cabe destacar que estos filtrados se encuentran relacionados y por tanto sólo permitirá seleccionar los nombres asociados al tipo de navío previamente seleccionado.

Una vez pintada la trayectoria del navío se podrán ver 2 métricas asociadas a dicha trayectoria, asociadas a la velocidad y el rumbo en cada punto de la trayectoria. A continuación se podrá ver el resultado de utilizar esta pantalla, divido en 2 figuras.

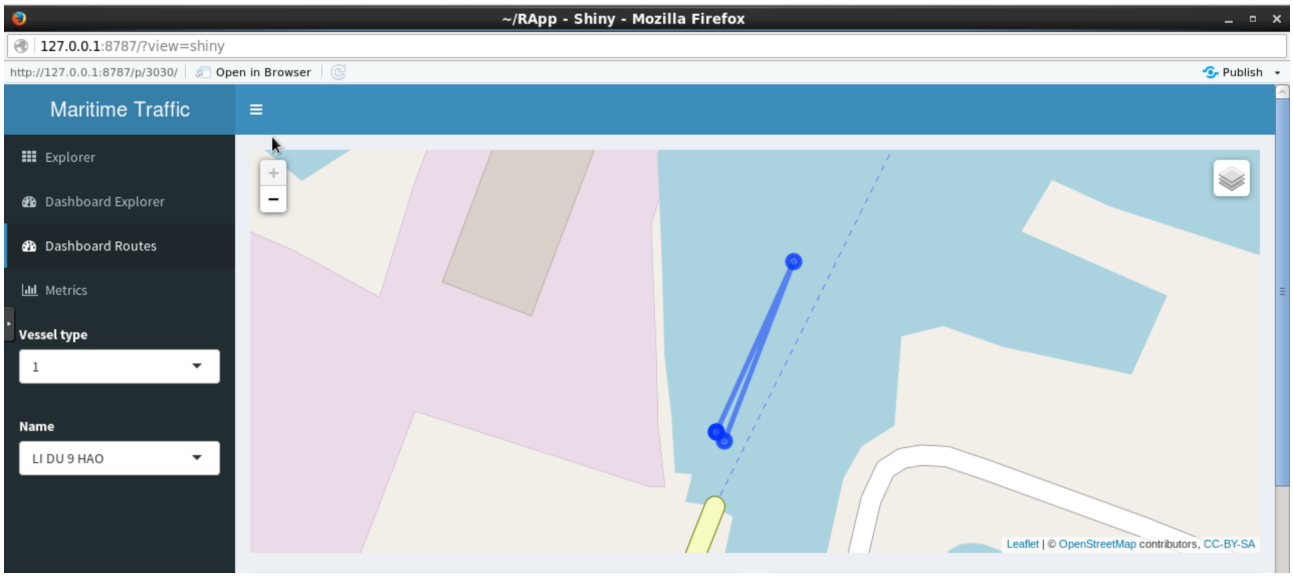


Figura 12.4: Página "Dashboard Routes" visualizando la trayectoria

Figura 12.5: Página "Dashboard Routes" visualizando las estadísticas de la trayectoria

Metrics. En esta pantalla se persigue mostrar diversas estadísticas asociadas a los datos globales de los navíos almacenados, coincidiendo estás con las definidas en el Capítulo 10. Al mostrar varias gráficas, la pantalla es de mayor tamaño, siendo imposible mostrar todas en una misma imagen, a continuación se podrán ver las diferentes capturas correspondientes a esta

pantalla.

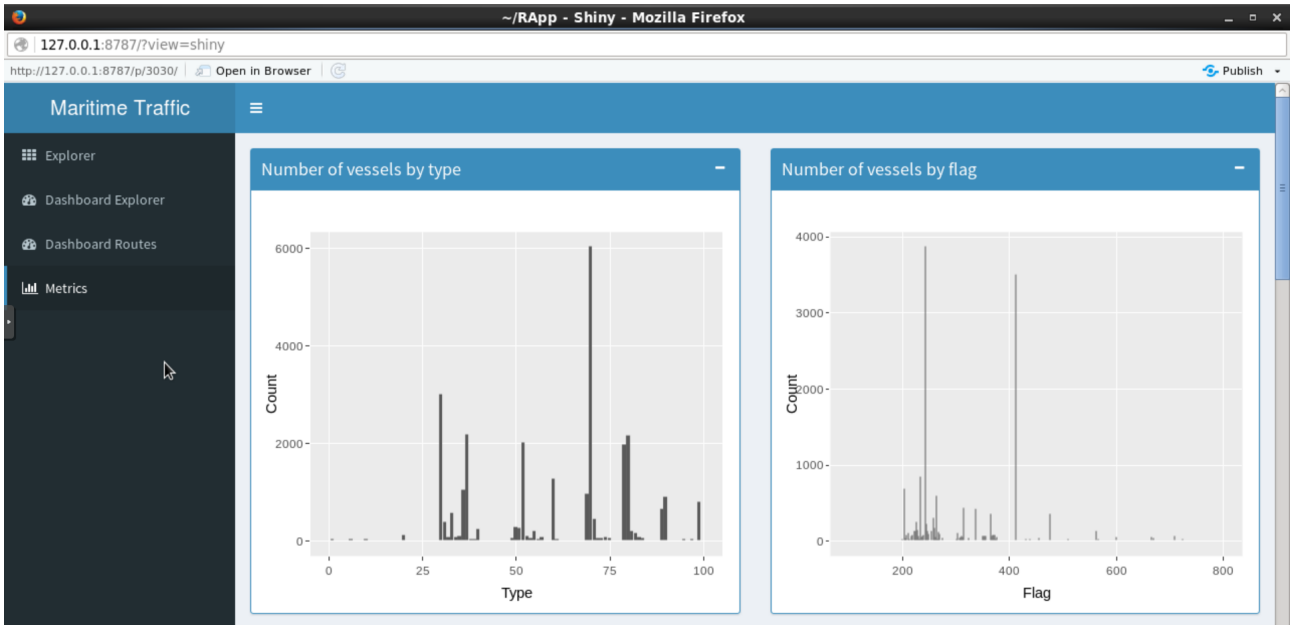


Figura 12.6: Página "Metrics", visualización gráficos iniciales

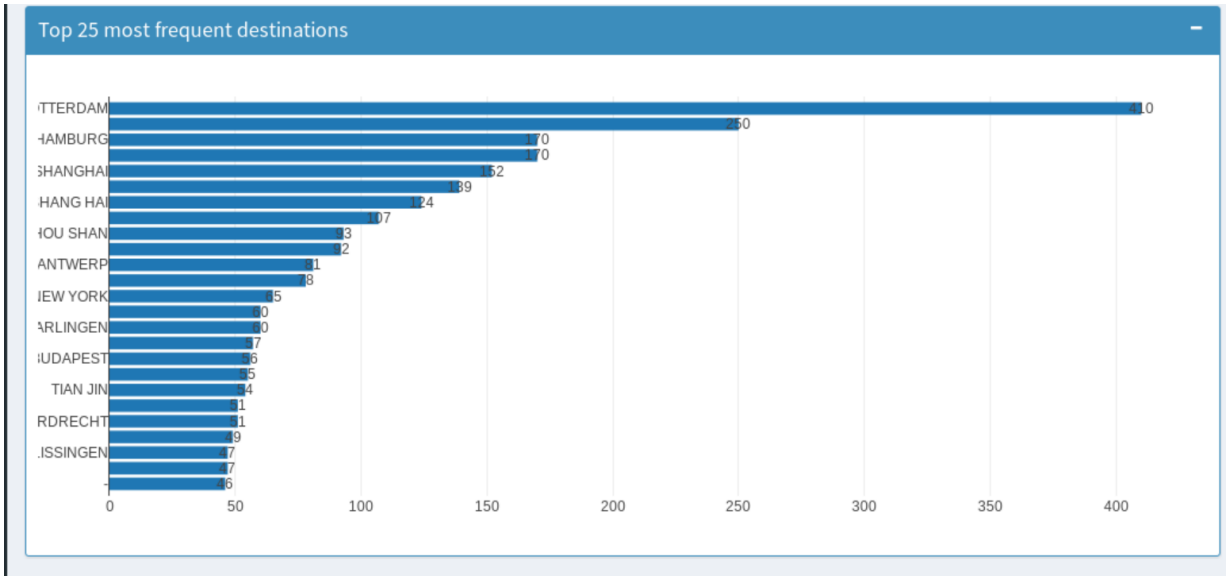


Figura 12.7: Página "Metrics", visualización gráficos iniciales

Además de estas métricas globales se podrán ver las diferentes métricas asociadas a la distribución de navíos por puerto, en este caso por nacionalidad y tipo. Con este fin existen dos desplegables que permiten seleccionar los puertos de destino disponibles por los navíos. En las figuras 12.8 y 12.9 se muestran los resultados de dichas métricas.

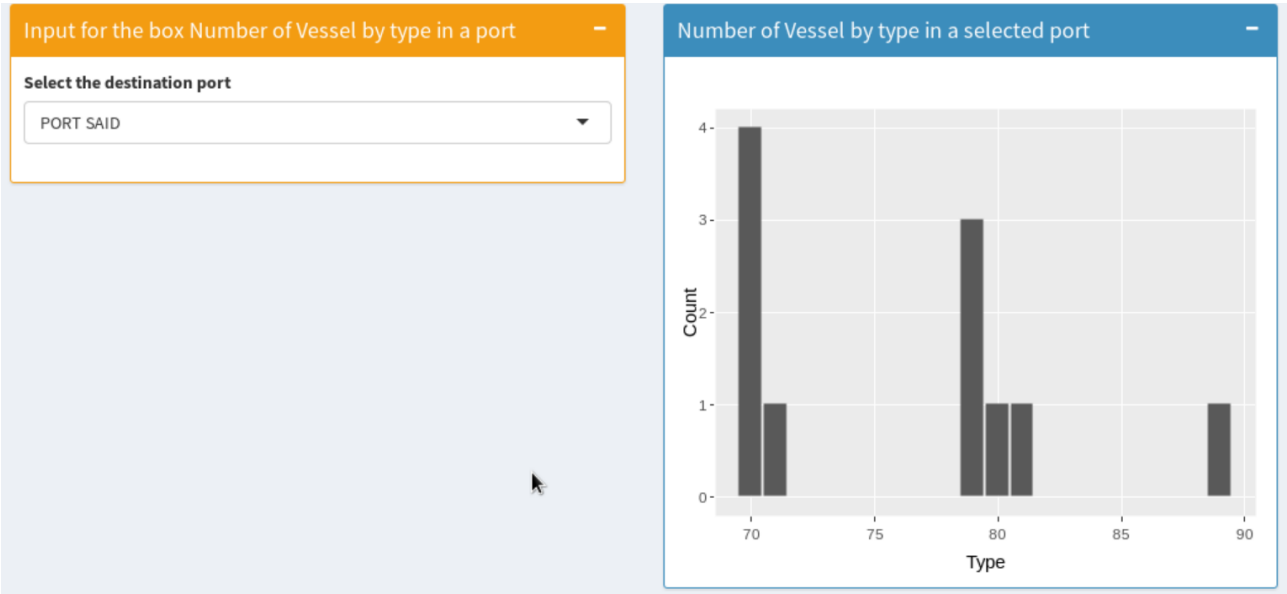


Figura 12.8: Página "Metrics", visualización gráficos distribución por puerto

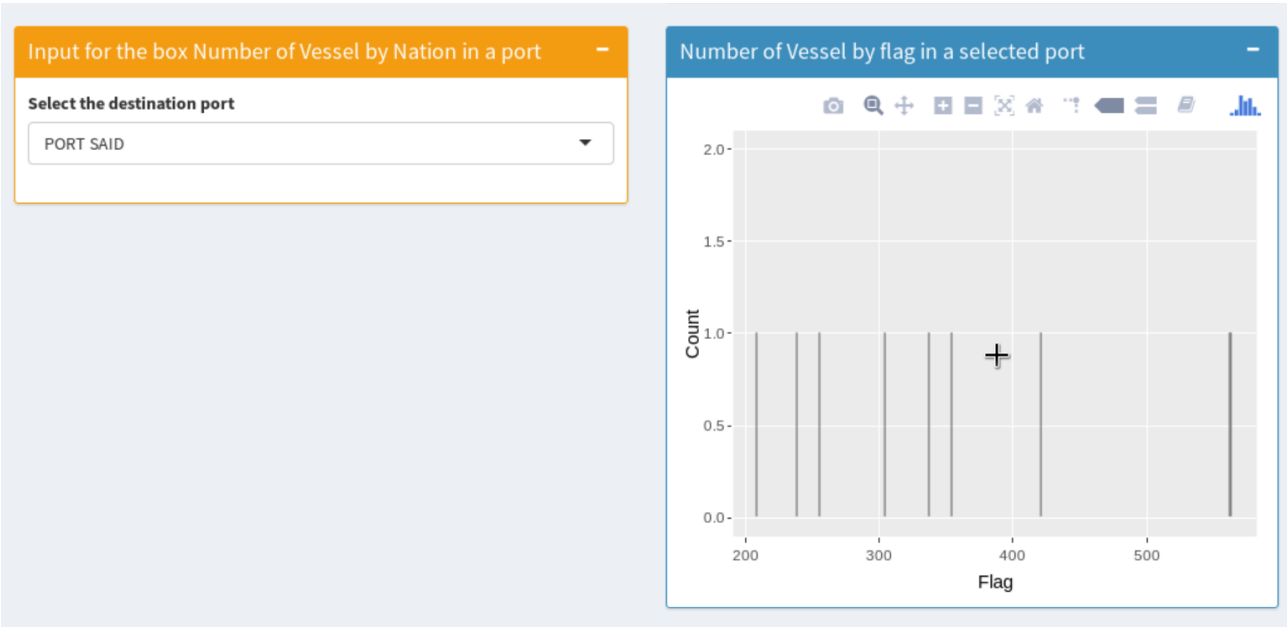


Figura 12.9: Página "Metrics", visualización gráficos distribución por puerto

Capítulo 13

Conclusiones y Trabajo Futuro

En este último capítulo del proyecto se presentarán las conclusiones derivadas de la realización de este proyecto, así como las principales metas alcanzadas. Además de esto se plantearán posibles mejoras y ampliaciones del proyecto desarrollado.

A lo largo de este proyecto me he encontrado con gran cantidad de dificultades, siendo la principal de éstas el desconocimiento total tanto de la tecnología a utilizar como del ecosistema marítimo. Esta dificultad nos obligó a comenzar a investigar el ecosistema marítimo a nivel de usuario, para luego tratar de comprender el funcionamiento de las emisiones realizadas por los navíos, los protocolos de comunicación existentes o los diferentes agentes y empresas que intervienen en la gestión de la información del ecosistema marítimo. Una vez investigado este ecosistema se pudo desarrollar el modelo de datos del proyecto, el cual es la piedra angular de este.

Otra gran dificultad a la que me enfrenté fue el desconocimiento de las diferentes arquitecturas Big Data, así como la dificultad a la hora de acceder a fuentes fiables de información con las que entender qué es un Data Lake.

Además de las dificultades iniciales asociadas a la investigación, anteriormente descritas, me he encontrado con gran cantidad de problemas derivados de la falta de recursos para tratar grandes volúmenes de información. Estos problemas se deben a la falta de memoria RAM de la máquina virtual con la que se realiza el proyecto, así las diferentes restricciones hardware que afectan a la máquina sobre la que se monta el ecosistema Big Data.

Por otro lado el trabajo con grandes volúmenes de datos y con tecnologías muy novedosas, como las tecnologías Big Data utilizadas para el almacenamiento y procesamiento de la información asociada al tráfico marítimo, me ha supuesto gran cantidad de problemas, al tener que afrontar un gran número de complicaciones no planteadas con anterioridad y derivadas del gran volumen de información al que tenemos acceso.

A lo largo de este proyecto se han dado gran cantidad de problemas asociados con el sistema operativo en el que se ha montado el ecosistema Big Data, como por ejemplo la unión entre R y Cloudera. Estos problemas me han permitido refrescar los conocimientos adquiridos en la carrera sobre sistemas, así como a dominar un nuevo lenguaje de programación, en auge en la actualidad, como es R.

13.1. Conclusiones

La principal conclusión derivada del desarrollo de este proyecto es que este tipo de tecnologías son ya el presente de la informática, ya que en la actualidad el mundo se mueve en torno a grandes volúmenes de información. El tratamiento de estos volúmenes de información supone un gran cambio que se engloba dentro de la transformación digital a la que se enfrentan gran cantidad de empresas en la actualidad. Es por eso que el desarrollo de este proyecto me ha supuesto un aprendizaje continuo y de gran valor para mi futuro laboral, ya que el aprendizaje de estas tecnologías supone la incursión en un mundo laboral falto de profesionales en este ámbito.

Otra conclusión que se extrae de este TFG es la gran necesidad de adaptación al cambio necesaria a la hora de desarrollar un proyecto real. Este hecho se asocia principalmente a la gran velocidad de evolución de la tecnología, así como a la diferente visión, sobre el producto a desarrollar,de las diferentes personas que componen el equipo de trabajo.

Sin embargo, la conclusión más interesante que he podido extraer de este TFG es el gran nicho de mercado que supone la aplicación de tecnologías Big Data al ámbito marítimo. La implantación de este tipo de sistemas se encuentra actualmente en nacimiento y por tanto aún hay pocas empresas que dedicadas a la explotación y gestión de esta información. Un buen tratamiento de esta información podría permitir reducir la contaminación asociada al tráfico marítimo, optimizando rutas, o incluso evitar colisiones entre los navíos.

13.2. Trabajo Futuro

Aunque la herramienta desarrollada en este proyecto es completamente funcional y escalable existen diversos aspectos que se pueden mejorar con el paso del tiempo.

- **Uso de bases de datos NOSQL.** Se podrían utilizar bases de datos NOSQL con el fin de agilizar la visualización de la información en la aplicación web desarrollada. Esto surgiría como complemento al ecosistema desarrollado en la actualidad, ya que sería imposible realizar el trabajo de limpieza de información sin este ecosistema.

- **Ampliar fases de limpieza.** Pese al desarrollo de diferentes fases de limpieza debido a los problemas tecnológicos a los que nos hemos enfrentado, así como a la dificultad de aprendizaje de estas tecnologías, ha hecho que no nos sea posible desarrollar todas las fases de limpieza de datos del proyecto.

- **Ampliar licencias sobre los datos.** Pese a que para esta fase inicial del proyecto los datos con los que se ha contado han sido suficientes se podrían adquirir licencias de pago que permitieran el seguimiento vía satélite de los navíos, así como el análisis de estos en tiempo real. Esto, unido a una mejora en el hardware disponible, facilitaría el desarrollo de gran cantidad de métricas de interés.

13.3. Aprendizaje

Una vez finalizado el desarrollo de este proyecto he adquirido una gran cantidad de conocimientos que serán de gran ayuda en mi futuro laboral.

En primer lugar el utilización de herramientas Big Data antes del desarrollo de este proyecto era para mi un sueño prácticamente inalcanzable, en gran parte por su complejidad y por la inexistente formación adquirida en estas tecnologías. Por ello considero que la evolución en el aprendizaje de estas tecnologías ha sido muy positiva.

En segundo lugar el desarrollo de la aplicación web de este proyecto me ha permitido conocer un lenguaje de programación totalmente desconocido para mi hasta el desarrollo de este proyecto, como es R. Además de esto me ha permitido aprender a utilizar este tipo de lenguajes junto a tecnologías Big Data, aprendiendo también a configurar su conexión a bajo nivel.

De manera externa al ámbito tecnológico este proyecto me ha permitido conocer como nace un proyecto real, gracias a mis tutores y a hAItta Sàrl. Además de esto me ha supuesto un gran desarrollo en mis capacidades de comunicación, así como un cambio en mi manera de pensar sobre cómo se desarrollan los proyectos informáticos.

Este proyecto además me ha permitido conocer una metodología de trabajo independiente. Esto se debe a la gran libertad existente a la hora de realizar un TFG, en la que los plazos te los marcas tú, a diferencia de las asignaturas desarrolladas a lo largo de la carrera, en la que te debes ajustar a unos criterios y plazos determinados con anterioridad.

Además de los anteriores aprendizajes, el desarrollo de este proyecto me ha permitido mejorar mi capacidad crítica, a la hora de investigar y descartar información. Este proyecto me ha supuesto un aprendizaje continuo, ya que además de todas las tecnologías y lenguajes no vistos con anterioridad, me ha permitido ampliar mis conocimientos sobre Data Lakes, además de conocer en profundidad el ecosistema de datos marítimo y los diferentes agentes que lo componen.

Además en gran cantidad de ocasiones me ha sido necesario aplicar los diferentes conocimientos adquiridos en asignaturas de la carrera para la elaboración de este proyecto:

- Aplicación de recolección: Para el desarrollo de la aplicación recolectora de datos ha sido de gran valor los conocimientos sobre el lenguaje de programación Java y la programación multihilo adquiridos en **Programación Orientada a Objetos** y en **Sistemas Distribuidos**.

- Aplicación web: Para el desarrollo de la aplicación web de este proyecto ha sido importante tener los conocimientos adquiridos en las asignaturas de **Tecnologías Web** y **Plataformas Software Empresariales**, así como de las asignaturas de programación vistas en la carrera y que te enseñan a utilizar estructuras de datos. Estos conocimientos me han permitido conocer lo qué deseo desarrollar y por tanto únicamente debería adaptarme al lenguaje con el que se desarrollará la aplicación.

- Arquitectura Big Data: Para el diseño y creación del Data Lake me han sido de gran utilidad los conocimientos adquiridos en la asignatura de **Administración de Bases de Datos**, facilitándome también el desarrollo del modelo conceptual.

- Fases de limpieza: Para la realización de las diferentes transformaciones realizadas en los datos han sido muy útiles los conocimientos SQL adquiridos en las asignaturas **Administración de Bases de Datos** y **Sistemas de Bases de Datos**, así como los conocimientos adquiridos en la asignatura **Programación Orientada a Objetos**, al dotarme de conocimiento sobre el lenguaje Java, en el que se basa el Programa Map Reduce creado.

- Instalación del entorno: Para la instalación de R, y de todas las librerías necesarias para desarrollar la aplicación web y su conexión con Cloudera, han sido de gran utilidad los conocimientos adquiridos en las asignaturas de **Utilización de Sistemas Operativos** y **Administración de Sistemas Operativos**, al interactuar con un sistema Unix.
- Documentación: Para llevar a cabo la documentación de este proyecto, así como la gestión del mismo, han sido de gran utilidad los conocimientos adquiridos en asignaturas como **Gestión de Proyectos Basados en las Tecnologías de la Información**, **Modelado de Software**, **Plataformas Software Empresariales** o **Proceso de Desarrollo del Software**.

Bibliografía

[1] *A Big Data Architecture for Managing Oceans of Data and Maritime Applications*. Zenodo, jun. de 2017. DOI: 10.5281/zenodo.833359. URL: https://doi.org/10.5281/zenodo.833359.

[2] AISHub. *AIS data sharing and vessel tracking by AISHub*. Mar. de 2018. URL: http://www.aishub.net/.

[3] Jose Blanco. *Que es un data lake? (Big Data y BI)*. 2015. URL: https://www.linkedin.com/pulse/que-es-un-data-lake-big-y-bi-jose-blanco.

[4] Travis CI. *Test and Deploy with Confidence*. 2017. URL: https://travis-ci.org/.

[5] CollabNet. *CollabNet VersionOne Gartner's recognized leader in Agile Management*. 2018. URL: https://www.collab.net/.

[6] Louis Columbus. "IBM Predicts Demand For Data Scientists Will Soar 28 % By 2020". En: *Forbes* (2017). URL: https://www.forbes.com/sites/louiscolumbus/2017/05/13/ibm-predicts-demand-for-data-scientists-will-soar-28-by-2020/#7cda85db7e3b.

[7] EMSA. *LRIT International Data Exchange*. 2018. URL: http://www.emsa.europa.eu/lrit-home/lrit-ide.html.

[8] *European Maritime Safety Agency*. European Maritime Safety Agency. Enero 2018. URL: http://www.emsa.europa.eu/.

[9] H. Fang. "Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem". En: *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2015, págs. 820-824. DOI: 10.1109/CYBER.2015.7288049.

[10] FleetMon. *FleetMon*. 2017. URL: https://www.fleetmon.com/.

[11] FreeNMEA. *freenmea*. 2017. URL: http://freenmea.net/.

[12] GitHub. *GitHub built for developers*. 2017. URL: https://github.com/.

[13] Jordan Golson. "Amazon is now facilitating sea freight shipments". En: *The Verge* (2017). URL: https://www.theverge.com/2017/1/25/14387154/amazon-sea-cargo-freight-logistics.

[14] Headwaters Group. "Your Unstructured Data Is Sexy – You Just Don't Know It [CIO STRATEGY]". En: *Headwaters Group Blog* (2017). URL: https://content.headwaters.group/blog/your-unstructured-data-is-sexy-see-how.

[15] IBM. *10 Key Marketing Trends for 2017 and Ideas for Exceeding Customer Expectations*. IBM. 2017.

[16] *IMDatE*. European Maritime Safety Agency. Enero 2018. URL: http://emsa.europa.eu/operations/maritime-monitoring/86-maritime-monitoring/1520-integrated-maritime-data-environment-imdate.html.

[17] MarineTraffic. *MarineTraffic*. 2017. URL: https://www.marinetraffic.com/.

[18] Bernard Marr. "3 Massive Big Data Problems Everyone Should Know About". En: *Forbes* (2017). URL: https://www.forbes.com/sites/bernardmarr/2017/06/15/3-massive-big-data-problems-everyone-should-know-about/#65278be61862.

[19] M. A. Martínez-Prieto y col. "Integrating flight-related information into a (Big) data lake". En: *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*. Sep. de 2017, págs. 1-10. DOI: 10.1109/DASC.2017.8102023.

[20] NGA. *World Port Index*. Abr. de 2018. URL: https://msi.nga.mil/NGAPortal/MSI.portal?_nfpb=true\&_pageLabel=msi_portal_page_62\&pubCode=0015.

[21] NGA. *World Port Index (PUB 150)*. 2017. URL: https://msi.nga.mil/MSISiteContent/StaticFiles/NAV_PUBS/WPI/Pub150bk.pdf.

[22] NMEA. *NMEA 0183 Standard*. 2017. URL: https://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp.

[23] D. E. O'Leary. "Embedding AI and Crowdsourcing in the Big Data Lake". En: *IEEE Intelligent Systems* 29.5 (2014), págs. 70-73. ISSN: 1541-1672. DOI: 10.1109/MIS.2014.82.

[24] Management Plaza. *Cómo funciona Scrum*. 2017. URL: http://managementplaza.es/blog/como-funciona-scrum/.

[25] Beulah Salome Purra Pradeep Pasupuleti. *Data Lake development with Big Data : explore architectural approaches to building Data Lakes that ingest, index, manage, and analyze massive amounts of data using Big Data technologies*. Birmingham, 2015.

[26] Ken Takagiwa. *Install g++/gcc 4.8.2 in CentOS 6.6*. 2017. URL: https://gist.github.com/giwa/b1fb1e44dc0a7d270881.

[27] VTExplorer. *Vessel MMSI MID Codes*. 2017. URL: http://www.vtexplorer.com/mmsi-mid-codes-en/.

[28] Robin Wigglesworth. "Maersk's tanker unit invests in quant hedge fund". En: *Financial Times* (2017). URL: https://www.ft.com/content/6fa02436-873e-11e7-8bb1-5ba57d47eff7.

Apéndice A

Glosario

A continuación se especificarán los diferentes términos propios de la navegación utilizados a lo largo del documento, así como las abreviaturas usadas para ellos (ver tabla A.1).

Término	Definición
Automatic Identification System (AIS)	Sistema, creado por la IMO, que permite a los navíos comunicar su posición y la información de interés propia del navío.
Long-range identification and tracking	Sistema creado por la IMO que permite a los navíos comunicar su posición y la información de interés propia del navío a su país de bandera
Gross Tonnage (GRT)	Forma de medir el tamaño de los navíos a partir de su volumetría
Carga Especial	Se considera carga especial a todo elemento que por su peso o dimensión no pueda ser transportado o cargado en buques convencionales
Nº de identificación del servicio móvil marítimo (MMSI)	Serie de 9 dígitos encargada de identificar de manera única a cada estación de servicio móvil
Organización Marítima Internacional (IMO)	Organismo encargado de promover la cooperación entre Estados y la industria del transporte, persiguiendo una mejora en la seguridad marítima y la prevención de la contaminación
NMEA 183	Estándar de comunicación entre los diferentes agentes marítimos. Todos los datos marítimos en bruto tienen este formato

Tabla A.1: Glosario

Apéndice B

Contenido del CD-ROM

Se entregará un CD-ROM que contenga los programas y script de limpieza de datos, así como el código fuente de la aplicación desarrollada y una copia de esta memoria.

El contenido del CD sigue la siguiente estructura de directorios:

- Aplicación web: En este directorio se encuentra el fichero con el código fuente de la aplicación web desarrollada en lenguaje R.
- Data Lake: Este directorio se subdivide a su vez en 2 directorios, así como un fichero workflow que no entraría en dichos directorios:
  - Map Reduce: Directorio en el que se encuentra el programa Map Reduce desarrollado para realizar una primera fase de procesamiento.
  - Hive: Directorio en el que se encuentran los diferentes scripts de procesamiento de datos y creación de tablas en Hive.
  - Workflow.xml: Fichero encargado de lanzar el programa Map Reduce implementado.
- Memoria TFG: En este directorio se encuentra una copia en formato PDF de este documento.
- Aplicación Recolector: En este directorio se encuentra el código fuente del programa de obtención de datos desarrollado en Java en la fase inicial de este proyecto.