



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Despliegue de motor analítico y herramienta de visualización de
datos (Big Data) para el conocimiento del entorno marítimo en
tiempo real*

Grado en Ingeniería Mecánica

ALUMNO: Juan Cuquerella Lorenzo

DIRECTORES: Belén Barragáns Martínez
Pablo Sendín Raña

CURSO ACADÉMICO: 2020-2021

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Despliegue de motor analítico y herramienta de visualización de
datos (Big Data) para el conocimiento del entorno marítimo en
tiempo real*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

UniversidadeVigo

RESUMEN

Con la llegada de la Cuarta Revolución Industrial, la Armada Española se ha visto necesitada de una renovación y actualización de sus unidades y sistemas que le permita mantenerse a la vanguardia de las potencias navales en el mundo. Con este propósito, la Armada inició un proyecto de investigación liderado por el CUD, en el que participa el COVAM, con el objeto de promover la aplicación de Inteligencia Artificial y *Big Data* para el conocimiento del entorno marítimo, dentro del cual se encuadra el presente trabajo.

En este TFG se diseña la arquitectura de un sistema que permite almacenar en una plataforma *Big Data* un flujo de datos AIS en tiempo real facilitando su procesamiento posterior. En la arquitectura desplegada se emplea, tras una detallada revisión y selección, Apache Kylin como motor analítico y Apache Superset como herramienta de visualización de datos.

Este sistema permite realizar consultas con muy bajas latencias cuyos resultados son visualizados en tres paneles de mando y control, diseñados atendiendo a las necesidades del COVAM.

PALABRAS CLAVE

Dashboard, Apache Kylin, Apache Superset, *Big Data*, COVAM

AGRADECIMIENTOS

Me gustaría que mis primeras palabras de agradecimiento fueran hacia mi familia que con sus rezos y sacrificios ha demostrado durante estos cinco años de carrera, y más aún durante el desarrollo de este TFG, ser el apoyo firme e inamovible sobre el que todo marino debe sostenerse.

También quiero, y debo, agradecer a mis tutores Belén y Pablo por su inmensurable paciencia hacia mi innata torpeza con la informática. Es gracias a ellos y a su atenta mentorización que este trabajo ha resultado tan gratificante. Tenía especial interés en aprender de este tema ya que, sin duda alguna, conformará parte del futuro de la Armada y sin ellos no habría sido posible.

Por último, dado que este trabajo pone fin a mi estancia en la Escuela Naval Militar, quiero agradecer a todos mis compañeros de la promoción 421-151 junto a los cuales he pasado estos maravillosos cinco años de mi vida tan emblemáticos y definitivos para mi formación moral y profesional. Es gracias a ellos que recordaré esta etapa de mi carrera con el cariño y nostalgia que se merece.

CONTENIDO

Contenido	1
Índice de Figuras	3
Índice de Tablas.....	7
1 Introducción.....	9
1.1 Contexto y motivación	9
1.2 Objetivos	11
1.3 Estructura de la memoria	11
2 Estado del arte	13
2.1 Inteligencia Artificial	13
2.1.1 Definición	13
2.1.2 Importancia en la actualidad.....	14
2.2 <i>Big Data</i> y <i>Data Analytics</i>	16
2.2.1 Definición de <i>Big Data</i>	16
2.2.2 Importancia de <i>Big Data</i> en la actualidad.....	17
2.2.3 <i>Data Analytics</i>	19
2.3 Interés de la IA, <i>Big Data</i> y <i>Data Analytics</i> en el ámbito de Defensa y Armada.....	20
2.4 Trabajos Fin de Grado relacionados	21
2.4.1 Selección e implantación de una herramienta de diseño de <i>dashboards</i> de aplicación al COVAM	22
2.4.2 Plataforma <i>Big Data</i> para el análisis de flujos de información marítima.....	23
2.5 Revisión de motores analíticos	24
2.5.1 Descripción de los motores analíticos	27
2.5.2 Conclusión	34
2.6 Revisión de herramientas de visualización	35
2.6.1 Descripción de las herramientas de visualización	35
2.6.2 Conclusión	43
3 Desarrollo del TFG.....	45
3.1 Datos AIS	45
3.1.1 Cola <i>ship_info</i>	46
3.1.2 Cola <i>routes</i>	48
3.2 Arquitectura del sistema.....	49
3.3 Instalación y configuración.....	51
3.3.1 Requisitos previos.....	52

3.3.2 Instalación y configuración de Apache Kylin.....	58
3.3.3 Instalación y configuración de Superset	59
3.4 Creación del panel de mando y control.....	62
3.4.1 Población de las colas Kafka	62
3.4.2 Generación del cubo OLAP	62
3.4.3 Establecimiento de conexión entre Apache Kylin y Apache Superset.....	69
3.4.4 Análisis de la información	70
4 Presentación de resultados.....	89
4.1 <i>Dashboard</i> de enfoque global	89
4.2 <i>Dahboard</i> de posibles actividades ilícitas o de interés	90
4.3 <i>Dashboard</i> del Estrecho de Gibraltar.....	91
5 Conclusiones y líneas futuras	93
5.1 Revisión de los objetivos	93
5.2 Líneas Futuras	94
6 Bibliografía.....	95

ÍNDICE DE FIGURAS

Figura 1-1 Captura de pantalla AIS del 25/1/21 a las 20:15 UTC [3].....	10
Figura 2-1 Capas de las redes neuronales artificiales [11]	14
Figura 2-2 Tecnologías constituyentes de la Industria 4.0 [15]	15
Figura 2-3 Las 7V de <i>Big Data</i> [22].....	17
Figura 2-4 Dashboards de la empresa Zeus [24]	20
Figura 2-5 Dashboard con representación de datos globales para el COVAM [28].....	23
Figura 2-6 Dashboard histórico de tráfico en el Mediterráneo [28]	23
Figura 2-7 Logotipo de Apache Druid [36].....	27
Figura 2-8 Arquitectura de Apache Druid [38]	27
Figura 2-9 Modelo de almacenamiento orientado a filas [39]	28
Figura 2-10 Modelo de almacenamiento orientado a columnas [39]	28
Figura 2-11 Logotipo Apache Kylin [42].....	29
Figura 2-12 Ejemplo de metodología de un Cubo OLAP [45]	30
Figura 2-13 Esquema del proceso <i>map-shuffle-reduce</i> [47].....	31
Figura 2-14 Esquema en estrella [51].....	31
Figura 2-15 Arquitectura de Kylin [38]	32
Figura 2-16 Logotipo de Pinot [52].....	32
Figura 2-17 Proceso de consulta en Apache Pinot [52]	33
Figura 2-18 Logo de ClickHouse [55]	33
Figura 2-19 Página principal del sitio web de Cyfe	36
Figura 2-20 Página principal del sitio web de Klipfolio	37
Figura 2-21 Página principal del sitio web de Geckoboard	37
Figura 2-22 Página principal del sitio web de Superset	38
Figura 2-23 Página principal del sitio web de Tableau	39
Figura 2-24 Página principal del sitio web de Grafana	40
Figura 2-25 Página principal del sitio web de InetSoft	40
Figura 2-26 Página principal del sitio web de Zeppelin.....	41
Figura 2-27 Página principal del sitio web de Clicdata.....	42
Figura 2-28 Página principal del sitio web de Chartio	43
Figura 3-1 Registro de datos de la cola Kafka <i>routes</i>	46
Figura 3-2 Registro de datos de la cola Kafka <i>ship_info</i>	48
Figura 3-3 Zonas definidas para la cola <i>routes</i> [elaboración propia]	48
Figura 3-4 Flujo de datos tradicional [76].....	50

Figura 3-5 Flujo de datos del sistema <i>Real-Time OLAP</i> [elaboración propia]	51
Figura 3-6 Arquitectura del sistema Apache Kylin-Apache Superset [elaboración propia]	51
Figura 3-7 Creación de usuario hadoop	53
Figura 3-8 Fichero <i>hadoop_java.sh</i>	53
Figura 3-9 Ficheros de configuración de Hadoop	54
Figura 3-10 Fichero <i>hadoop-env.sh</i> con el <i>path</i> de Java añadido	54
Figura 3-11 Fichero <i>core-site.xml</i> con las propiedades añadidas.....	54
Figura 3-12 Fichero <i>hdfs-site.xml</i> con las propiedades añadidas	55
Figura 3-13 Fichero <i>mapred-site.xml</i> con las propiedades añadidas.....	55
Figura 3-14 Fichero <i>yarn-site.xml</i> con las propiedades añadidas.....	55
Figura 3-15 Inicio de Apache Hadoop y confirmación con el comando <i>jps</i>	56
Figura 3-16 <i>Path</i> de Hadoop en <i>hive-config.sh</i>	56
Figura 3-17 Hive <i>shell</i>	56
Figura 3-18 Configuración del fichero <i>hbase-site.xml</i>	57
Figura 3-19 Configuración del fichero <i>hbase-env.sh</i>	57
Figura 3-20 HBase <i>shell</i>	57
Figura 3-21 variable <i>\$KYLIN_HOME</i>	58
Figura 3-22 Error en inicio de Kylin	58
Figura 3-23 Descarga de Apache Spark	58
Figura 3-24 Iniciación correcta de Kylin	59
Figura 3-25 Página principal de la interfaz gráfica de Kylin	59
Figura 3-26 Dependencias con Python.....	59
Figura 3-27 Creación y activación de un entorno virtual	60
Figura 3-28 Instalación de Apache Superset	60
Figura 3-29 Inicialización de la base de datos	60
Figura 3-30 Creación de usuario administrador	60
Figura 3-31 Configuración por defecto de los Roles y permisos	61
Figura 3-32 Inicio de Apache Superset	61
Figura 3-33 Página de acceso a la interfaz web de Apache Superset.....	61
Figura 3-34 Comandos empleados para la creación de la cola <i>ship_type</i>	62
Figura 3-35 Lanzamiento del programa productor de registros para las colas Kafka.....	62
Figura 3-36 Comprobación de la existencia de registros en la cola <i>ship_info</i>	62
Figura 3-37 Elección de proyecto y tipo de <i>Data Source</i>	63
Figura 3-38 Elección del <i>topic</i> y del <i>Bootstrap Server</i>	63
Figura 3-39 Creación de la tabla <i>ship_info</i>	63
Figura 3-40 Selección de la tabla para el modelo	64

Figura 3-41 Definición de las dimensiones del modelo	64
Figura 3-42 Definición de la columna de partición del modelo	64
Figura 3-43 Definición del modelo del cubo.....	65
Figura 3-44 Definición de las dimensiones del cubo para <i>ship_info</i>	65
Figura 3-45 Definición de los umbrales de compresión para el cubo <i>ship_info</i>	66
Figura 3-46 Definición de la jerarquía y unión entre dimensiones para el cubo <i>ship_info</i>	66
Figura 3-47 Definición del <i>rowkey order</i> para el cubo <i>ship_info</i>	67
Figura 3-48 Definición del <i>cube engine</i> para el cubo <i>ship_info</i>	67
Figura 3-49 Definición de las propiedades de Kylin <i>streaming</i> para el cubo <i>ship_info</i>	67
Figura 3-50 Definición de los receptores accesibles para el cubo <i>ship_info</i>	68
Figura 3-51 Definición de los <i>replica set</i> establecidos para el cubo <i>ship_info</i>	68
Figura 3-52 Cubo <i>ship_info</i> activado y recibiendo mensajes de la cola Kafka	68
Figura 3-53 Consulta a tabla de datos <i>ship_info</i>	69
Figura 3-54 Instalación de <i>kylinpy</i>	69
Figura 3-55 Conexión Apache Superset-Apache Kylin	70
Figura 3-56 <i>Datasets</i> de la interfaz gráfica de Superset.....	70
Figura 3-57 Adición de <i>Datasets</i>	71
Figura 3-58 Ejemplo de consulta con <i>SQL Lab</i>	71
Figura 3-59 Ejemplo de consulta con el método de desplegables de Superset	72
Figura 3-60 Conjunto de <i>charts</i> disponibles en Apache Superset.....	73
Figura 3-61 Adición del URL público de un mapa <i>Map Box</i> en <i>config.py</i>	74
Figura 3-62 Página principal de <i>Map Box</i>	74
Figura 3-63 Consulta para el <i>chart</i> "Mapa Estrecho de Gibraltar"	76
Figura 3-64 Mapa Estrecho de Gibraltar	77
Figura 3-65 Mapa Estrecho de Gibraltar + 5 minutos.....	77
Figura 3-66 Mapa Estrecho de Gibraltar + 10 minutos.....	77
Figura 3-67 Diseño del <i>Filter Box</i>	78
Figura 3-68 Diseño del <i>chart</i> "Número de embarcaciones a la deriva"	79
Figura 3-69 Diseño del <i>chart</i> "Embarcaciones a la deriva"	79
Figura 3-70 Información de una embarcación a la deriva en un panel de control	80
Figura 3-71 Diseño del <i>chart</i> "Gráfico de tráfico de buques de gran tamaño"	80
Figura 3-72 Consulta para el <i>chart</i> "Posible transbordo de pesca en la mar"	81
Figura 3-73 Posible transbordo de pesca en la mar.....	82
Figura 3-74 Consulta para el <i>chart</i> "Posible <i>bunkering</i> en la mar"	82
Figura 3-75 Posible <i>bunkering</i> en la mar	83

Figura 3-76 Diseño del <i>chart</i> "Derrota del J.S. Elcano"	83
Figura 3-77 Diseño del <i>chart</i> "Tipos de buques según zonas de interés"	84
Figura 3-78 Consulta para el <i>chart</i> "Mapa global"	85
Figura 3-79 Mapa Global	85
Figura 3-80 Diseño del <i>chart</i> "Mensajes AIS totales"	86
Figura 3-81 Consulta para el <i>chart</i> "Embarcaciones en zonas de interés"	86
Figura 3-82 Embarcaciones en zonas de interés.....	87
Figura 3-83 Diseño del <i>chart</i> "Tabla mensajes AIS totales"	87
Figura 4-1 <i>Dashboard</i> de enfoque global.....	90
Figura 4-2 <i>Dashboard</i> de posibles actividades ilícitas o de interés	91
Figura 4-3 <i>Dashboard</i> del Estrecho de Gibraltar	92

ÍNDICE DE TABLAS

Tabla 2-1 Motores analíticos de interés.....	26
Tabla 2-2 Comparativa de motores analíticos	26
Tabla 2-3 Comparativa final de motores analíticos.....	34
Tabla 2-4 Comparativa de herramientas de visualización.....	43
Tabla 3-1 Descripción del valor <i>postype</i> [73]	47
Tabla 3-2 Descripción del valor de <i>shiptype</i> [73]	47
Tabla 3-3 Descripción de los valores de <i>navstatus</i> [75].....	49

1 INTRODUCCIÓN

1.1 Contexto y motivación

Desde la llegada de Internet el mundo en el que vivimos se ha visto envuelto en un entorno turbulento y de constante cambio. Internet ha llegado a todos los rincones del planeta revolucionando a su paso prácticamente todos los ámbitos de la sociedad.

Con el objetivo de mantenerse a la vanguardia de los cambios que esta tecnología traía, las grandes potencias mundiales se vieron involucradas en el proceso de digitalización de sus empresas, dando lugar a la denominada Cuarta Revolución Industrial.

En esta gran revolución, ha destacado el empleo omnipresente de las herramientas y tecnologías de información y comunicación (TIC), que han exigido un proceso de evolución repentino y profundo dando como fruto la denominada Sociedad de la Información. Sin embargo, la información se generaba a una velocidad superior a la que las herramientas eran capaces de procesarla además de la añadida complejidad que suponía la heterogeneidad del inmenso número de fuentes que la generaba.

En este marco se desarrollaron por primera vez los conceptos como *Big Data* o *Data Analytics*, que se comentarán en el apartado 2.2, con el objetivo de procear ingentes flujos de datos y desarrollar herramientas capaces de su gestión (bases de almacenamiento, software de procesamiento de datos, motores analíticos, herramientas de visualización, etc.).

El entorno marítimo no ha sido ajeno a esta revolución industrial. Siendo precisamente un entorno en el que todo depende de variables (altura de las mareas, oleaje, viento, tráfico en las rutas mercantes, atracadas y desatrachadas en los millones de puertos que pueblan el mundo, etc.) la inmensa generación de grandes volúmenes de datos de los diferentes sensores, boyas y sonoboyas, faros, estaciones costeras, estaciones radio de embarcaciones y sistemas GPS (*Global Positioning System*) entre otros, ha proporcionado grandes beneficios para el desarrollo de este sector.

Quizás la fuente de generación de datos de mayor importancia dentro del entorno del tráfico marítimo es el sistema AIS (*Automatic Identification System*). La Organización Marítima Internacional (OMI) [1] define el sistema AIS como un sistema que proporciona el posicionamiento, identificación e información de buques a otros buques o autoridades costeras de manera automática. La OMI, por medio del convenio internacional SOLAS (*Safety of Life at Sea*), exige el uso de AIS a todos aquellos buques de más de 300 toneladas brutas que se encuentren realizando viajes internacionales, buques cargueros con más de 500 toneladas brutas y buques de pasajeros sin importar el tamaño. Según [2], se estima que al año más de 300,000 dispositivos AIS transmiten su ubicación e información referente a su buque

(rumbo, velocidad, puerto origen, puerto destino, eslora, etc.). También estima que un 80% de los barcos pesqueros en alta mar usan este dispositivo, a pesar de no estar obligados.

En la Figura 1-1 se puede observar una captura de pantalla de la aplicación web VesselFinder [3], aplicación de seguimiento y rastro de buques que emplea como fuente de datos el sistema AIS. En esta imagen se muestran los buques detectados en el Mediterráneo el día 25 de enero de 2021 a las 20:15 UTC. Con esta imagen nos podemos hacer una idea de la cantidad de información que se transmite en cada instante de tiempo en todas partes del mundo.

En la Armada, el órgano gestor encargado de fusionar y analizar la información que se recibe de un elevado número de fuentes, entre las que destaca el AIS, es el COVAM (Centro de Operaciones y Vigilancia de Acción Marítima) [4]. El COVAM tiene la responsabilidad de realizar el arduo trabajo que supone generar una imagen precisa de todo lo que sucede en los espacios marítimos de interés nacional. Esta imagen, que ha de procesarse lo más aproximadamente a tiempo real posible, se denomina *Recognized Maritime Picture (RMP)* y es el producto final que el COVAM pone a disposición de todos los buques de la Armada que se encuentren navegando, así como de las Agencias Civiles que lo soliciten y que tengan competencias en nuestras aguas.

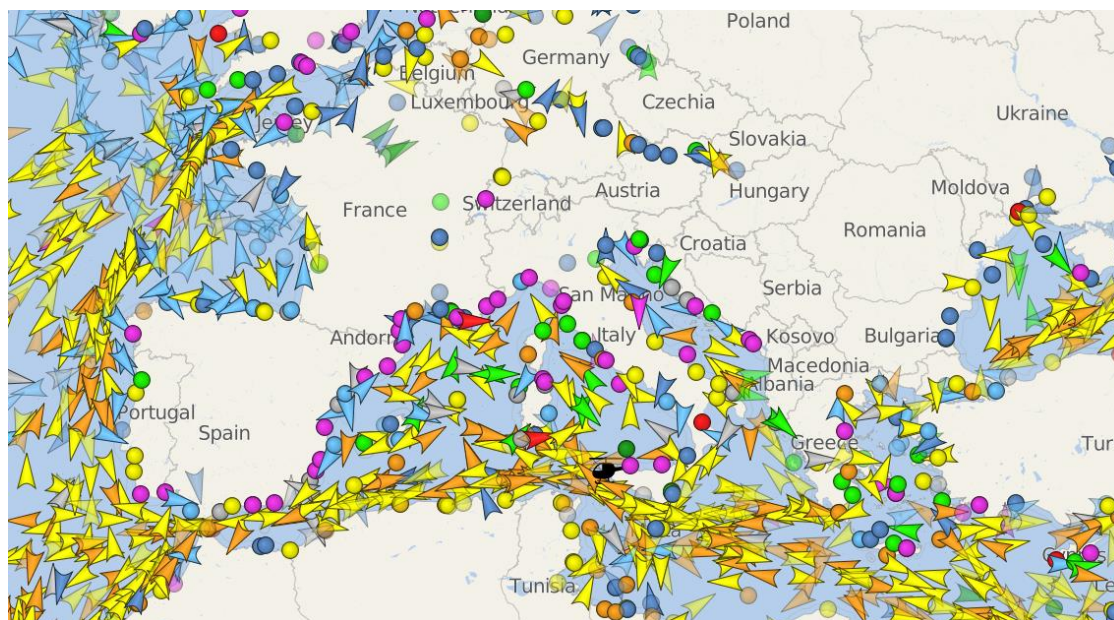


Figura 1-1 Captura de pantalla AIS del 25/1/21 a las 20:15 UTC [3]

Viendo la inmensa cantidad de datos con las que debe trabajar, es fácil entender que el COVAM, con el objetivo de mantener a la Armada al frente de esta revolución industrial junto a las fuerzas navales más potentes y punteras del mundo, haya sugerido una actualización en sus sistemas software de gestión de datos. De esta manera, se puso en marcha un proyecto, que a su vez engloba un conjunto de trabajos de los que se hablará en el apartado 2.4 y entre los cuales se encuentra el presente trabajo, con la intención de desarrollar un sistema capaz de identificar anomalías, patrones y comportamientos específicos en los grandes volúmenes de datos AIS y presentarlos en un panel de control que sirva de cuadro de mando y control para facilitar la toma de decisiones.

Los beneficios que se pueden obtener de la aplicación de un cuadro de mando y control en la seguridad marítima son incontables. Según [5], España se sitúa entre los diez países del mundo con mejor conectividad marítima. A pesar de la gran ventaja económica que esto supone para el país, también le hace susceptible de albergar ciertas actividades delictivas en sus rutas de comercio, como pueden ser el terrorismo, el tráfico de armas y personas, la explotación ilegal de recursos marinos, el tráfico de drogas y la inmigración ilegal. Todo ello viene a demostrar la gran necesidad de una vigilancia, seguridad y control de nuestro entorno marítimo que requiere de una atención indiscutible.

1.2 Objetivos

El objetivo de este Trabajo de Fin de Grado (TFG) es diseñar la arquitectura de un sistema que permita almacenar en una plataforma *Big Data* un flujo de datos AIS en tiempo real para facilitar su procesamiento y análisis posterior, así como la realización de consultas con muy bajas latencias. Los resultados de dichas consultas deben poder ser visualizados en uno o varios paneles de control que faciliten una rápida toma de decisiones. Los paneles, o *dashboards*, serán diseñados considerando los indicadores de interés para el COVAM que le permitan obtener una imagen clara del entorno.

Se pretende desplegar, instalar y configurar dicho sistema mediante el empleo de un motor analítico y de una herramienta de visualización de datos. Para ello, es necesario realizar una revisión del estado del arte de las diferentes herramientas existentes aspirando a escoger las que mejor se adapten a las necesidades del trabajo.

1.3 Estructura de la memoria

A continuación, se procede a exponer la estructura adoptada para la organización de la memoria. Este documento consta de cinco capítulos y la bibliografía.

El Capítulo 1 contiene una introducción al tema del trabajo. Se presenta el contexto y motivación de este y se definen los objetivos principales perseguidos.

En el Capítulo 2 se describen los conceptos teóricos básicos utilizados durante el desarrollo de este trabajo y se realiza una revisión del estado del arte:

- En primer lugar, se introducen los conceptos de Inteligencia Artificial y *Big Data*, dentro del cual se asocia el concepto de *Data Analytics*, exponiendo también la importancia de estos en la actualidad.
- En segundo lugar, se procede a explicar el interés y la importancia de estos conceptos en el ámbito de Defensa y la Armada.
- En tercer lugar, se presentan brevemente las ideas principales de los trabajos que preceden a este TFG, mostrando también los resultados obtenidos.
- En cuarto lugar, se realizó una revisión comparativa de los motores analíticos escogidos, concluyendo en la selección de uno de ellos.
- En quinto lugar, se realiza una revisión comparativa de herramientas de generación de *dashboards* para acabar escogiendo la que mejor se adapte a la consecución de los objetivos establecidos.

En el Capítulo 3 se expone el desarrollo del TFG. Primero, se describe el proceso de instalación y configuración de las herramientas seleccionadas, incluyendo los requisitos previos que exige cada una. Tras esto, se describe detalladamente cómo conectar ambas herramientas entre sí y cómo emplearlas. Se describe, paso por paso, el procedimiento de generación de cubos *Real Time OLAP* con Apache Kylin y se definen los diferentes métodos de visualización que pueden emplearse para la presentación de datos en Apache Superset, así como los paneles generados para su posterior agregación a los *dashboards*.

En el Capítulo 4 se muestran los tres *dashboards* generados, producto final del trabajo. Se explica el por qué del diseño de cada uno de estos *dashboards*, la información que aportan y su utilidad.

En el Capítulo 5 se comentan las conclusiones alcanzadas tras el desarrollo del TFG, proponiendo algunas mejoras y líneas futuras que propicien el avance de este proyecto.

Por último, se incluyen las referencias bibliográficas y sitios web consultados durante el desarrollo del trabajo.

2 ESTADO DEL ARTE

En este capítulo se introducirán conceptos de especial relevancia para el desarrollo de este trabajo como pueden ser los conceptos de Inteligencia Artificial (IA), *Data Analytics* y *Big Data*, y se expondrá su importancia tanto en el mundo actual como en el ámbito de Defensa y la Armada Española. Se realizará además una revisión de los distintos motores analíticos y herramientas de visualización con la intención de seleccionar los más adecuados para la consecución del objetivo final del trabajo, presentado en el apartado 1.2. Este capítulo finalizará con la selección de una herramienta de *Data Analytics* y una herramienta de generación de paneles de control. De esta manera se pretende realizar una revisión del estado del arte yendo de lo más genérico a lo más concreto y específico.

2.1 Inteligencia Artificial

2.1.1 Definición

Según B. Jack Copeland, director del *Turing Archive for the History of Computing* y autor del libro *Inteligencia Artificial*, la Inteligencia Artificial se puede definir como la capacidad de una máquina computadora u ordenador para realizar actividades comúnmente asociadas a seres inteligentes [6].

Según un estudio realizado por Luis Amador Hidalgo, catedrático de la Universidad de Córdoba y autor del libro *Inteligencia Artificial y Sistemas Expertos* [7], entre estas actividades en el mundo de la inteligencia artificial se suele hacer especial hincapié en los procesos cognitivos empleados por el ser humano como pueden ser la memorización (almacenamiento y recuperación de datos), el aprendizaje, la solución de problemas, la inferencia y deducción lógica, la percepción y reconocimiento de formas, la toma de decisiones y la comprensión del lenguaje natural.

De esta manera, según [7], la IA es una disciplina de la ciencia que se puede definir de dos formas distintas dependiendo del objetivo que con esta se pretenda alcanzar.

Por un lado, se puede hablar de la IA que se centra en el estudio de los procesos cognitivos del ser humano mencionados anteriormente. Se intenta obtener un desarrollo sistemático teórico que simule o defina las actividades realizadas por el intelecto humano simplificando así el entendimiento de este y facilitando su posterior implementación en un ordenador. De esta forma, los ordenadores se ven desplazados por esta definición a un segundo plano como herramientas para expresar teorías en forma de programas.

Por otro lado, se define la IA que se basa en el objetivo de la creación de sistemas automáticos, sistemas capaces de realizar ciertas funciones desempeñadas hasta hace no mucho por los seres humanos. Esta definición proporciona un enfoque más tecnológico a la IA, que se centra más bien en la obtención

de máquinas y ordenadores con aptitudes incluso superiores a las del ser humano, haciéndola más acertada para este trabajo.

2.1.2 Importancia en la actualidad

Desde su nacimiento, aceptado por muchos expertos en la materia que tuvo lugar en 1937 con la publicación de la máquina universal de Turing por parte del famoso matemático Alan Turing, máquina hipotética capaz de desarrollar cualquier procedimiento matemático [7]; la IA ha ido asumiendo un papel cada vez más importante en el mundo moderno, extendiendo su ámbito de aplicación a prácticamente todas las facetas de la sociedad actual: ciencia y desarrollo de nuevas tecnologías, informática, comercio, transporte, habitabilidad, etc. Esto se debe al creciente interés en esta disciplina por parte del mundo académico y empresarial.

Hace tiempo que los seres humanos dejamos de realizar ciertas funciones básicas debido a la mayor eficiencia y exactitud del resultado obtenido por los ordenadores. Estas funciones básicas que han sido delegadas a las máquinas suelen agruparse en tres categorías [8]: cálculo numérico, almacenamiento de información y ejecución de operaciones repetitivas.

Sin embargo, actualmente nos encontramos en una etapa de la tecnología en la que se pretende que los ordenadores aprendan a razonar.

En la actualidad la IA se ha ido introduciendo paulatinamente en nuestras vidas mediante su agregación a productos ya existentes (como se puede observar en los coches inteligentes de última generación y en los teléfonos móviles con asistentes virtuales). Es decir, la IA se ha hecho hueco en nuestra sociedad aportando automatización y una enorme cantidad de datos a plataformas y productos convencionales.

Se automatiza el aprendizaje de las máquinas a través de la realización de tareas computacionales de gran volumen y frecuencia prácticamente sin fallos y sin fatiga. Estas tareas se adaptan a través de algoritmos de aprendizaje mediante el estudio de cambios en la estructura o regularidad de los datos [9]. Además, estos procesos presentan una gran capacidad de escalabilidad, como se muestra en la Figura 2-1, gracias a la posibilidad de adición de capas ocultas en redes neuronales que emulan las del cerebro humano [10].

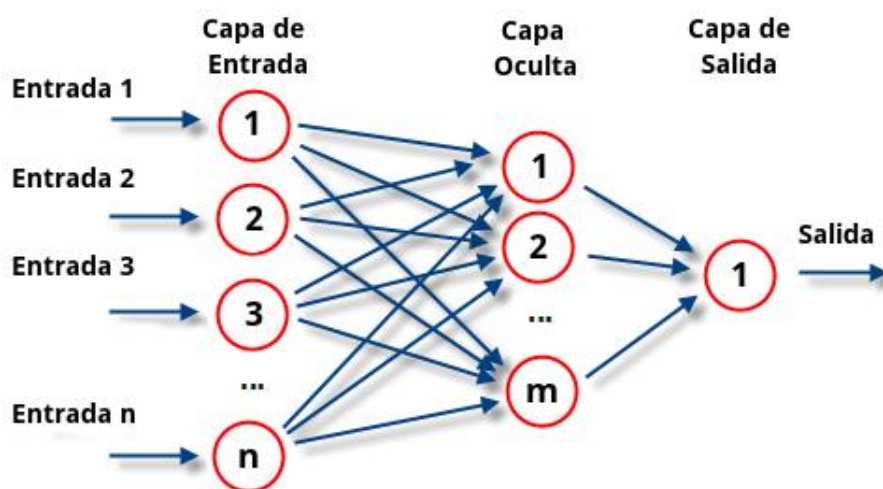


Figura 2-1 Capas de las redes neuronales artificiales [11]

En el ámbito industrial, este almacenamiento, recopilación y empleo de datos se traduce en una gran ventaja competitiva. Estos datos son generados, procesados y analizados continuamente por motores analíticos permitiendo a las máquinas y procesos industriales recopilar información de estos por sí mismos, incluso en tiempo real, mejorando notablemente la planificación, el diseño de productos y las operaciones de producción.

Ejemplo de este avance en la industria es el sistema operativo abierto IoT (Internet de las Cosas) de Siemens: MindSphere [12]. MindSphere es capaz de interconectar plantas y sistemas enteros además de vincular equipos y productos gracias a la tecnología IoT debido a su baja frecuencia (generalmente por debajo de 1Ghz) y gran poder de penetración. El IoT genera una gran cantidad de datos los cuales son analizados exhaustivamente por este sistema operativo permitiendo una optimización y simulación de la producción, así como la toma de decisiones.

Estas aplicaciones de la IA están siendo utilizadas junto a conceptos como el de Gemelo Digital (representación digital de sistemas y plantas completas sobre las cuales se puede realizar un estudio en profundidad del objeto real y su reacción a diversas condiciones y cambios) [13], realidad aumentada, impresión 3D o ciberseguridad para propulsar la ya activa Industria 4.0 (Figura 2-1) [14].



Figura 2-2 Tecnologías constituyentes de la Industria 4.0 [15]

El comercio también se ha visto revolucionado por completo debido a la IA y en especial, el almacenamiento de información a gran escala. Los datos se convierten en conocimiento y hasta hace unos años este era limitado. Esto cambia con el auge del comercio electrónico, el cual basa su éxito en la gran cantidad de datos proporcionados por los motores de búsqueda. El simple hecho de navegar por Internet pone en funcionamiento esta aplicación de la IA permitiendo a las diferentes empresas detectar qué productos y tendencias son los más solicitados además de generar sugerencias específicamente elaboradas para un individuo en concreto sobre productos que le pueden ser de interés gracias al estudio de nuestros hábitos y comportamientos. Este mismo método es también usado en servicios de ocio como Netflix, Spotify o Youtube [16].

Este estudio y análisis de datos característico de la IA se hace también visible en las oficinas y casas, en las cuales los asistentes personales se están viendo suplantados por asistentes virtuales. Estos asistentes, entre los cuales destacan Siri de Apple, Google Home de Google y Alexa de Amazon; son ya capaces de aprender nuestras rutinas, comportamientos y voz, entre otras cosas.

Los ejemplos expuestos anteriormente son solo una pequeña demostración de todo lo que ha afectado la inteligencia artificial en la sociedad y pueden ser extrapolados a una visión más amplia y global. Es así como ha nacido la idea de las *Smart Cities* (ciudades inteligentes) [17].

Las *Smart Cities* supondrían la interconexión de todos los ámbitos constituyentes de una sociedad: medioambiente, movilidad y transporte, alumbrado, sanidad, economía, etc. Todo ello basado en una tecnología en la que se complementarían sistemas integrados de hardware y software de última generación tales como: IoT, Cloud, *Big Data*, Inteligencia Artificial y sensores entre otros.

Estas *Smart Cities* darían respuesta a las necesidades cambiantes de una ciudad altamente desarrollada gracias a la mejora de la eficiencia y calidad de los servicios públicos. Sin embargo, todavía quedan muchos obstáculos que salvar para alcanzar este objetivo, como el dilema entre el empleo de la máxima cantidad de información que se pueda adquirir y las políticas de privacidad de datos, o la consecución de mayores velocidades de transmisión manteniendo las coberturas y los bajos consumos.

Es observable que la mayoría de las ventajas obtenidas de las nuevas tecnologías fruto de la IA se basan en gran medida en los algoritmos y técnicas de aprendizaje autónomo a través de los cuales las máquinas desarrollan la capacidad de realizar ciertas actividades y tomar decisiones específicas gracias a la enorme base de datos que las sustentan y alimentan mediante un flujo de datos continuo y en tiempo real, es decir, gracias al término *Big Data*.

2.2 *Big Data* y *Data Analytics*

2.2.1 Definición de *Big Data*

Hoy en día una de las definiciones más aceptadas del término *Big Data* es la dada por Kenneth Cukier, autor del libro *Big Data. La Revolución de los Datos Masivos* [18], en el cual lo define del siguiente modo: “se trata de hacer cosas a partir del análisis de inmensas cantidades de información, que simplemente no son posibles con volúmenes más pequeños”.

Se puede por tanto aceptar el concepto de *Big Data* como el estudio y análisis masivo de datos para conseguir de esta manera extraer información, tomar decisiones y alcanzar conclusiones gracias a la adquisición de un enorme conjunto de datos. Es necesario añadir que cuando se hace referencia a un conjunto de datos como *Big Data* este posee un tamaño (volumen), complejidad (variedad) y velocidad de crecimiento suficiente como para que su gestión, procesamiento y análisis no pueda ser realizado por herramientas de gestión de datos convencionales.

De esta manera quedan definidas las 3V (Volumen, Variedad y Velocidad) con las que comenzó la tecnología basada en *Big Data* a las cuales, a lo largo de estos últimos años marcados en el mundo de la informática por el interés en el desarrollo de esta disciplina de la tecnología, se le han añadido otras cuatro características más constituyendo las conocidas 7V del *Big Data*, mostradas en la Figura 2-3: Volumen, Variedad, Velocidad, Veracidad, Viabilidad, Visualización y Valor. El centro de investigación español del Instituto de Ingeniería del Conocimiento (IIC) [19] da con una muy acertada explicación de cada una de ellas:

- **Volumen:** se refiere a la cantidad masiva de datos generados, almacenados, procesados y analizados. No hay una medida clara a partir de la cual se puede considerar un conjunto de datos como *Big Data*; sin embargo, la gran mayoría de analistas y expertos de la materia hablan de volúmenes de entre 30 Terabytes hasta varios Petabytes. Según un estudio realizado por Dell Technologies, las grandes organizaciones pasaron de gestionar 1,45 PB en 2016 a 9,7 PB en 2018 de media general [20].
- **Velocidad:** se refiere a la rapidez con la que los datos son generados, almacenados y gestionados. Esta cualidad de los datos cobra especial importancia cuando se trata de procesos en los que se requiere llevar a cabo un estudio en tiempo real para que estos no pierdan su utilidad (la detección de un fraude en una transacción bancaria, por ejemplo).
- **Variedad:** hace referencia a la naturaleza compleja del *Big Data* debido a la gran diversidad de formas, tipos y fuentes en las que se registran los datos heterogéneos. Pueden ser datos estructurados y de gestión sencilla, como lo es una base de datos, o no estructurados que

requieren de herramientas específicas para su manipulación (los datos de sensores, vídeos, imágenes, correos electrónicos, publicaciones en redes sociales o el número de veces que abrimos una página web determinada, por ejemplo). Normalmente, para poder tratar con este tipo de datos, se suelen combinar con una base de datos estructurados de alguna aplicación más convencional como pueden ser las ERP (*Enterprise Resource Planning*) o CRM (*Customer Relationship Management*) [21].

- Veracidad: se trata de la incertidumbre de los datos, del grado de fiabilidad de la información que nos proporcionan los datos. Debido a la gran variedad y cantidad de los datos que se generan constantemente es necesario disponer de métodos y soluciones que permitan desechar datos erróneos y diferenciarlos de los verdaderamente útiles.
- Viabilidad: hace referencia al uso eficaz que se le da a la gran cantidad de datos que maneja una empresa o compañía. Esta cualidad está, por tanto, relacionada con la innovación de los equipos de trabajo y el uso de tecnologías.
- Visualización de los datos: hace referencia a la presentación de los datos, cómo estos son mostrados. Tras un filtrado y procesamiento de los datos es necesario presentar los resultados obtenidos de manera legible y comprensible para facilitar alcanzar conclusiones de manera visual. Existen herramientas de visualización y creación de dashboards (paneles de control) que se encargan de esta función y las cuales serán estudiadas más adelante en este capítulo.
- Valor de los datos: el valor de los datos manejados lo proporciona únicamente la información extraída de estos. Los datos de por sí no tienen valor. Es por tanto fundamental un buen proceso de traducción de datos a conocimiento ya que este es el objetivo final de *Big Data*.



Figura 2-3 Las 7V de *Big Data* [22]

2.2.2 Importancia de *Big Data* en la actualidad

Big Data supone un punto de referencia para las organizaciones, empresas y compañías. Como expresa PowerData (organización multinacional de servicios y tecnologías de la información) [21], *Big Data* muestra a estas empresas respuestas a muchas preguntas que ni siquiera sabían que tenían.

Permite extraer conocimiento por parte de empresas de un gran volumen de datos que antes no eran capaces de gestionar. Esta información puede tener utilidad en muchos ámbitos como en movimientos de negocio más inteligentes, operaciones más eficientes, mayores ganancias y mayor satisfacción del cliente.

Según [21], este valor de la información se puede agrupar en tres subapartados dependiendo del objetivo:

- Reducción de costes: el conocimiento que resulta del almacenamiento masivo de datos permite a las grandes organizaciones identificar las maneras más eficientes de hacer negocios, así como la capacidad de visualizar y cambiar las áreas en las que se estén produciendo pérdidas.
- Mayor velocidad en la toma de decisiones: las grandes velocidades en los análisis de datos proporcionan a las empresas la gran ventaja de actuar con celeridad sobre la información recién adquirida.
- Nuevos productos y servicios: disponer de un seguimiento y predicción del mercado tan acertado además del conocimiento de la satisfacción de los clientes o usuarios ha supuesto un avance importante para todas aquellas organizaciones implementando estas tecnologías. Con la analítica de *Big Data*, más empresas están creando nuevos productos para satisfacer las necesidades de los clientes.

A continuación, se expondrán ejemplos de algunos de los ámbitos en los que se observa la importancia que tiene el concepto de *Big Data* en la sociedad actual:

- En el turismo: *Big Data* permite, como ya se ha explicado anteriormente, medir la satisfacción de los clientes mediante el análisis de sus datos. De esta forma los turistas son los clientes en las ciudades y estos datos son registrados y utilizados por resorts, casinos, supermercados y centros comerciales, entre otros, permitiendo corregir los problemas identificados.
- En las empresas manufactureras: mediante sensores en productos, equipos, plantas y transporte, se puede llevar a cabo un estudio de los datos generados para identificar patrones de uso, tasas de fracaso, horas de trabajo, saturación de departamentos, etc.
- En la salud: Los planes de salud, registros de pacientes, información de los pacientes, capacidad restante de habitabilidad en los centros médicos, generación de diagnósticos... son solo algunos ejemplos de la gran ayuda y avance que ha proporcionado el *Big Data* en la industria sanitaria.
- En la publicidad: la proliferación del comercio electrónico y teléfonos inteligentes proporciona a las empresas una exagerada cantidad de datos sobre sus clientes y productos permitiéndoles hacer publicidad selectiva y dirigida. Además, el uso de dispositivos con GPS ofrece a los anunciantes la posibilidad de actuar cuando los portadores de estos dispositivos se encuentren cerca de tiendas, restaurantes o centros comerciales.
- En finanzas: posibilita la detección y prevención de fraudes en industrias que realicen transacciones financieras online como pueden ser compras, actividades bancarias, inversiones y seguros. Además, también se puede hacer uso de *Big Data* en la transacción de mercados financieros para una evaluación más rápida del riesgo y la toma de medidas de prevención y corrección.

Sin embargo, vista la gran cantidad de ventajas que trae consigo el *Big Data* también es conveniente analizar los desafíos que plantea, especialmente relacionados con la calidad de los datos.

En el pasado era habitual extraer la información de los sistemas CRM o ERM, por ejemplo, y a partir de ahí comenzar el proceso denominado ETL (*Extract, Transform and Load*). Sin embargo, ahora las organizaciones cuentan con fuentes de datos fuera del dominio de esta (como pueden ser los comentarios en redes sociales), lo que provoca una serie de problemas, expuestos en [21], mencionados a continuación.

Actualmente las fuentes de datos son muy diversas y heterogéneas, existiendo un amplio rango de orígenes de estos (datos de Internet y móviles, datos del IoT, datos experimentales, datos recopilados por empresas especializadas, etc.), y también los son los tipos de datos (estructurados, semiestructurados y no estructurados); generando grandes complejidades a la hora de su integración en las distintas máquinas y sistemas.

Como ya se ha visto anteriormente, la cantidad de datos generados es inmensa, dificultando la adopción de un proceso adecuado de control de calidad de estos. Es complicado filtrar, limpiar, procesar y analizar el extenso volumen de datos para la obtención de información de alta calidad a una velocidad adecuada. Sumado a esto, la gran mayoría de los datos son de tipo no estructurado y, por tanto, requieren ser transformados antes en datos estructurados.

Por último, es importante tener en cuenta la volatilidad de los datos, que cambian rápidamente adquiriendo una validez muy corta y que, para solucionarlo, se requiere un poder de procesamiento muy alto.

2.2.3 Data Analytics

Tras haber estudiado el concepto de *Big Data*, se puede entender la importancia de analizar la información que proporcionan los datos, es decir, la importancia del *Data Analytics*. *Data Analytics* se puede definir como el conjunto de herramientas y tecnologías empleadas para obtener resultados y conclusiones tras un análisis de los datos recopilados.

Este estudio analítico tiene como objetivo mejorar la eficiencia de la empresa u organización en cuestión y puede dividirse en cuatro pasos según [23], cada uno de ellos cobra mayor o menor importancia dependiendo del objetivo del estudio y del tipo de datos:

- *Extract, Transform and Load* (ETL): trata de emplear el proceso ETL para convertir la enorme cantidad de datos no estructurados recibidos en un formato más sencillo para su posterior almacenamiento, gestión y análisis. Es el paso en el que más tiempo se suele invertir.
- *Data Management*: implica diseñar e implementar bases de datos que permitan un acceso sencillo y rápido al resultado obtenido en el paso anterior. Solía implicar trabajar con bases de datos SQL, sin embargo, las bases de datos no relacionales están ganando popularidad.
- *Statistic Analysis*: consiste en la obtención de información como tal. Se utilizan técnicas y tecnologías de aprendizaje automático para el análisis de los datos y se implementan modelos estadísticos que revelan tendencias y patrones en los datos.
- *Data Presentation*: la presentación de los datos permite, como bien dice su nombre, presentar y compartir la información y los datos con las partes interesadas. La visualización de datos, parte importante de este paso, permite una rápida toma de decisiones y extracción de conclusiones con tan solo observar la información ordenadamente presentada en dashboards (paneles de control).

Como afirma la empresa de creación de dashboards Zeus [24] (Figura 2-4), la visualización de información inteligente (*Smart Visual Data*) de forma amena y sencilla les otorga a los datos orden y sentido. Esta técnica permite a los equipos de trabajo disponer de manera visual de los indicadores KPI (*Key Performance Indicators*) más relevantes en un proceso para la empresa.



Figura 2-4 Dashboards de la empresa Zeus [24]

2.3 Interés de la IA, *Big Data* y *Data Analytics* en el ámbito de Defensa y Armada

Al igual que en todos los demás ámbitos de la sociedad actual, los conceptos de IA, *Big Data* y *Data Analytics* han revolucionado las Fuerzas Armadas proporcionando nuevas tecnologías y metodologías para la realización de ciertas funciones de manera más precisa y eficaz.

Concretamente, en lo referente a la IA, la Armada colabora conjuntamente con Indra [25], una de las mayores empresas globales de tecnología y consultoría, en el desarrollo de dichas tecnologías para garantizar la actualización y adaptación de la Armada Española manteniéndola a la altura de las flotas más punteras mundialmente.

La colaboración con Indra se ha traducido en la creación de uno de los primeros sistemas de inteligencia artificial existentes que posee la capacidad de aprender y tomar decisiones por sí solo. Este sistema ha sido diseñado para su implementación en los buques de la Armada con el fin de predecir fallos y mejorar su mantenimiento y disponibilidad. El sistema es referido como proyecto Soprene [26] y utiliza IA no supervisada.

En la IA no supervisada la máquina o sistema no se limita únicamente al aprendizaje de un procedimiento para alcanzar la solución de un problema, función de la IA supervisada, sino que es también capaz de detectar el problema y, aplicando algoritmos de aprendizaje autónomo y operaciones lógicas conocidas por la gran mayoría de ordenadores, llegar a la solución por su cuenta, sin ayuda humana ni indicadores de cómo hacerlo.

Los ingenieros de Indra enseñaron al sistema milimétricamente y al detalle el funcionamiento de los motores de las fragatas F100 y los BAM (Buque de Acción Marítima). Tras este estudio, el sistema fue capaz de predecir las averías que estos buques habían sufrido en los últimos cinco años y detectar alteraciones en el funcionamiento normal en tiempo real [26].

El Centro de Supervisión y Análisis de Datos Monitorizados de la Armada (CESADAR), ubicado en Cartagena, ha sido la entidad directora técnicamente del proyecto dentro de la Armada y ha proporcionado el histórico de datos empleado para la comprobación de resultados.

Tras poner en marcha el sistema y comprobar su funcionamiento se concluyó que las ventajas que dicho sistema aporta son las siguientes:

- El aprendizaje automático de sus aciertos y errores mejora exponencialmente su eficacia y rapidez, y permite que este sistema llegue a resultados no predefinidos a priori.
- Es un sistema universal capaz de adaptarse a diferentes buques, no solo a las F100 y los BAM, e incluso de ser exportado a otros ámbitos de defensa o civiles.
- Como ya se ha mencionado, el sistema emplea algoritmos de aprendizaje automático no supervisado para la toma de decisiones y, por tanto, es capaz de predecir fallos y averías que el

buque todavía no ha sufrido. Estos fallos, al no poder ser reproducidos en un laboratorio para su estudio, son los más impredecibles y de mayor riesgo.

Este proyecto supone un gran avance para la Armada a la hora de planificar misiones con mayor precisión, asumiendo menos riesgos y adelantando reparaciones en caso necesario.

Entrando más en detalle, se puede hablar de la importancia del *Big Data* para las FAS (Fuerzas Armadas) como una consecuencia de la integración de la IA en este entorno.

Se ha visto en el apartado 2.2.1 que la generación masiva de datos por parte de numerosas y diversas fuentes, estructuradas o no, afecta a la sociedad actual en todos los sentidos. Sin embargo, como explica el Centro Superior de Estudios de la Defensa Nacional (CESEDEN) en el documento de investigación [27], es fácil inferir que el *Big Data* ha asumido un papel de especial importancia en el entorno de la defensa y la seguridad, ofreciendo grandes beneficios en la lucha contra los conflictos y crisis recientes que no hacen más que aumentar su grado de complejidad (gran número de actores con mayor interconectividad, nuevos escenarios con líneas divisorias entre lo civil y lo militar muy difusas, creciente mezcla de escenarios virtuales y reales, etc.).

Esta complejidad que caracteriza a las situaciones a las que se enfrenta la defensa y seguridad en la actualidad hace necesario adoptar un enfoque mayor en la comprensión de los conflictos debido a que las amenazas actuales procuran el aprovechamiento del entorno actual, haciendo uso de la congestión (de información, actores, alternativas, etc.) y del ruido (por el exceso de datos, falsas alarmas, complejidad en la diferenciación de lo civil y lo militar, etc.) que lo definen.

Según [27], las aplicaciones específicas se pueden agrupar en:

- Detección de intrusión física en grandes espacios o infraestructuras abiertas.
- Computación sobre información encriptada.
- Análisis automático de vulnerabilidades de red (máquinas-tráfico de datos).
- Criminología computacional.
- Uso fraudulento de recursos corporativos y/o sensibles.
- Análisis de vídeo en tiempo real / búsqueda y recuperación rápida en bibliotecas de vídeo.
- Inteligencia visual en máquinas.
- Identificación de anomalías, patrones y comportamiento en grandes volúmenes de datos.
- Análisis de texto (estructurado y no estructurado) como apoyo a la toma de decisión en tiempo real en entornos intensivos en datos.
- Consciencia situacional.
- Traducción automática a gran escala (en número de idiomas y en volumen).
- Predicción de eventos.

Precisamente una de estas aplicaciones (la identificación de anomalías, patrones y comportamientos en grandes volúmenes de datos) sirve para enmarcar el contexto de un conjunto de trabajos, realizados dentro de un proyecto de investigación del Centro Universitario de la Defensa (CUD), siendo la aplicación de IA para el conocimiento del entorno marítimo su idea principal y en la cual se encuadra el presente TFG.

2.4 Trabajos Fin de Grado relacionados

A la hora de alcanzar un mayor entendimiento de la motivación, justificación y estado del arte del presente trabajo resulta imprescindible el análisis de los dos trabajos que le preceden: *Selección e implantación de una herramienta de diseño de dashboards de aplicación al COVAM* (TFG del ahora AN Díaz García) [28] y *Plataforma Big Data para el análisis de flujos de información marítima* (TFG del ahora AN González Guitart) [29].

Como ya se expuso en el apartado 1.1, el COVAM fusiona y analiza información que recibe de un elevado número de fuentes heterogéneas. Una de estas fuentes es un flujo de datos AIS. Los anteriores TFG's se centraron en analizar este flujo de información utilizando herramientas *Big Data* (Apache Storm) o mediante su visualización en dashboards (paneles de control). Dado que el objetivo último de este TFG es utilizar un flujo de datos de AIS en tiempo real para presentar la información en un panel de control compatible, se puede observar que está estrechamente relacionado con [28] y [29].

2.4.1 Selección e implantación de una herramienta de diseño de dashboards de aplicación al COVAM

En [28] se realiza una comparativa de las herramientas especializadas en la creación de paneles de control capaces de trabajar con fuentes de *Big Data* para el posterior diseño de un *dashboard* que sirva de cuadro de mando y control para el COVAM.

La importancia del diseño de un panel de control se justifica en el concepto *Smart Visual Data*, visto en el apartado 2.2.3, nueva filosofía empresarial creada por Zeus (empresa de creación de *dashboards*) [24] que asegura que una presentación atractiva, equilibrada y armoniosa nos facilita el procesamiento de los datos dándoles orden y sentido. La empresa Zeus define los *dashboards* como: “la herramienta perfecta que hace posible que un equipo de trabajo tenga a su disposición la visualización de sus KPIs más importantes a la hora de tomar decisiones en la organización.”

Las herramientas estudiadas en este TFG son: Klipfolio, Cyfe, SAP Lumira, Inetsoft, Penatho, Clicdata, Datorama, Grafana, Chartio, Tableau, Geckoboard y QlikSense. La herramienta elegida es Grafana [30], una aplicación *open-source* instalable en Windows y Linux, entre otros sistemas operativos.

Grafana fue escogida por ser *open-source*, por tener acceso directo a la Demo sin requerir de ningún requisito previo (proporcionar el correo electrónico, tamaño de la empresa, intenciones, etc.) y sin necesidad de esperar a la respuesta del servicio técnico para comenzar a usarla, por tener la capacidad de crear dashboards ilimitados y por su sencillez (interfaz gráfica cómoda y fácil instalación).

Los datos proporcionados por el COVAM para este trabajo son dos ficheros zip con archivos CSV (*Comma-separated-values*) que contienen datos AIS de los días 11 y 12 de diciembre de 2019. Estos ficheros requirieron de un preprocesamiento para poder ser añadidos a una base de datos que sirve de consulta para la herramienta de creación de *dashboards*. Esta base de datos se crea de forma remota en un servicio MySQL en un ordenador con sistema operativo Linux perteneciente al CUD.

Una vez seleccionada Grafana como la herramienta más adecuada, se procede a su instalación de manera remota en una máquina Linux del CUD.

Por último, se procede al diseño del *dashboard*. Se crea una serie de paneles para representar los diferentes KPIs considerados de interés para el COVAM:

- Localización según el origen y el tipo de barco.
- Reloj.
- Porcentaje de buques según puerto de destino.
- Número de buques según tipo de barco.
- Número de buques según su país de origen.
- Tabla de datos AIS.
- Número de retransmisiones AIS a lo largo del tiempo.
- Panel de alertas.
- Localización de los buques en alerta por cercanía.

El resultado obtenido fue la creación de dos *dashboards* que muestran las distintas capacidades proporcionadas por Grafana y el gran potencial que tienen estos para mostrar información de manera

atractiva y sencilla para la posterior toma de decisiones. Los dos *dashboards* generados son: *Dashboard con representación de datos globales para el COVAM* y *Dashboard histórico de tráfico en el Mediterráneo*, mostrados en la Figura 2-5 y en la Figura 2-6 respectivamente.

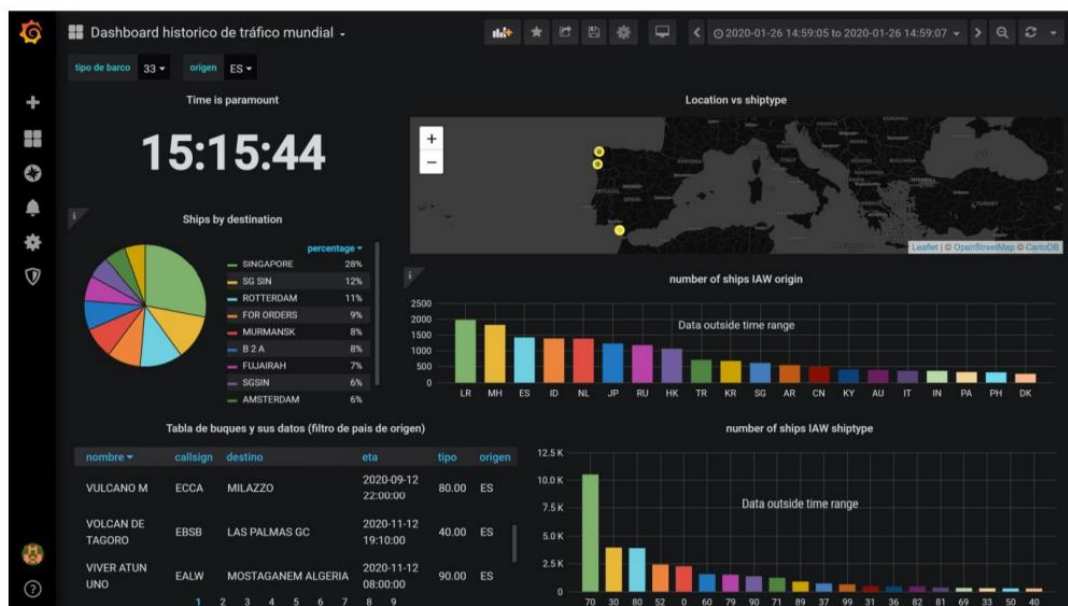


Figura 2-5 Dashboard con representación de datos globales para el COVAM [28]



Figura 2-6 Dashboard histórico de tráfico en el Mediterráneo [28]

2.4.2 Plataforma Big Data para el análisis de flujos de información marítima

El *Big Data* está adquiriendo una importancia irrefutable en el sector marítimo. Analizar el enorme flujo de datos AIS que continuamente genera la multitud de embarcaciones que lo emplean proporcionando información relevante sobre su origen, destino, tipo de buque, derrota, rumbo y velocidad, entre otros, puede suponer grandes beneficios para aplicaciones de seguridad. Sin embargo, *Big Data* también puede presentar problemas al tratarse de un volumen ingente de datos e información a considerar.

Por ello, en [29] se realiza un estudio en profundidad de las herramientas de software libre más populares de *Big Data* para el análisis de flujos de información y así escoger la más adecuada para este trabajo.

Las herramientas estudiadas son: Apache Spark, Apache Storm, Apache Flink, Apache Samza, Apache Flume y Apache Kinesis Data Analytics.

Tras este análisis se comparan las herramientas basándose en el estudio de investigación [31] que aboga un método de calidad para la elección de un sistema de procesamiento de flujo de tiempo real adecuado.

Utilizando este método se acaba por seleccionar cuatro criterios de comparación:

- Modelo de datos.
- Tolerancia a fallos.
- Semántica de mensajes.
- Latencia.

Siguiendo estos criterios se acaba escogiendo Apache Storm que presenta, entre otras, las siguientes cualidades: un procesado de flujo de datos rápido y eficiente con dos modelos diferentes (en microlotes y en tiempo real), una latencia del orden de milisegundos, y un consumo de CPU, memoria RAM, memoria en disco y ancho de banda muy aceptables en comparación con el resto de las herramientas.

Se validó la propuesta implementando Apache Storm en un servidor Linux que, utilizando datos AIS de los días 11 y 12 de diciembre de 2019, procesaba y limpiaba la fuente de datos y etiquetaba por zonas de interés cada embarcación seleccionada.

2.5 Revisión de motores analíticos

Para poder comenzar a generar el cuadro de mando y control para el COVAM con la alimentación del flujo de datos AIS en tiempo real, es necesaria la elección de un motor analítico que sirva como software de análisis y método de realización de las consultas a la base de datos.

Es importante destacar que este flujo de información en tiempo real de datos AIS es proporcionado por el COVAM y almacenado en un servidor Linux del CUD y que su procesado previo realizado por el almacén de datos analítico Apache Storm, en el cual se realiza un filtrado y eliminación de las duplicidades y errores existentes. Esto no es objeto de estudio del presente trabajo, aunque si se debe tener en cuenta ya que supone nuestra fuente de datos. La salida de datos de Apache Storm se conformará por una o varias colas Kafka. Apache Kafka [32] es un proyecto software dedicado a proporcionar a herramientas de análisis de datos acceso y disponibilidad a mensajes proporcionados por una base de datos. Este proceso se explicará con detalle en el apartado 3.4.1.

Cuando se habla de motores analíticos en el entorno de *Big Data* se hace referencia a herramientas que dan soporte al proceso analítico en línea de una base de datos. Este proceso analítico en línea se denomina OLAP (*Online Analytical Processing*). Estas herramientas OLAP permiten procesar grandes volúmenes de información, en campos bien definidos y con un acceso inmediato a los datos para realizar consultas y análisis posteriores.

Según [32] las funciones que debe ser capaz de llevar a cabo un motor analítico son las siguientes:

- Transformación de datos: antes del almacenamiento, los datos deben pasar un proceso previo de "transformación". Esto incluye operaciones de filtrado, reestructurado y combinación de datos.
- Almacenamiento: los datos deben almacenarse de manera que se maximice la capacidad de gestión y disponibilidad de estos.
- Análisis: OLAP incluye cálculos rápidos, análisis hipotéticos, pronósticos para análisis de datos eficientes, etc.

- Presentación y acceso a los datos: se emplean herramientas de visualización y debe proporcionar un acceso simple a los datos.

Por tanto, se puede resumir el concepto de motor analítico como una herramienta software de gestión de *Big Data* capaz de procesar y analizar la información obtenida de un gran volumen de datos, proporcionando una gran rapidez de respuesta a consultas realizadas por otro sistema o usuario final.

El proceso de elección de un motor analítico puede ser complicado debido a la gran cantidad de herramientas y servidores OLAP que hay disponibles en el mercado, cada uno implementando una arquitectura y una serie de funciones diferentes ya que, por lo general, suelen ser diseñadas para un propósito específico. En lo que a este trabajo respecta, se ha realizado un proceso de selección por eliminación, descartando todos aquellos que no cumplieran una serie de requisitos considerados imprescindibles. Una vez reducida la elección a un número de herramientas que permitiera una fácil comparación, se procede a la presentación en detalle de las herramientas que resultan de mayor interés para, por último, alcanzar una decisión.

Las características necesarias que debe poseer el motor analítico para este TFG son las siguientes:

- *Open-source*: es el término empleado para definir herramientas software de código abierto. La gran ventaja que supone utilizar herramientas *open-source* es la posibilidad de compartir, modificar y estudiar su código fuente. Además, en contraste con las herramientas específicas de determinadas compañías que requieren licencias de pago, se dispone de un gran número de aplicaciones compatibles con el sistema, lo cual proporcionará un mayor rango de decisión para la selección de una herramienta de visualización. Por último, supone una mayor confianza en la continuidad del programa gracias a la colaboración entre usuarios (para la corrección de fallos, por ejemplo) no dependiendo exclusivamente de una empresa específica.

- Compatible con herramientas de visualización *open-source*: por el mismo razonamiento que el punto anterior, es importante que el motor analítico escogido tenga la capacidad de conectarse a herramientas de visualización de código abierto. Es más, es importante que pueda disponer de más de una herramienta *open-source* de creación de *dashboards* para poder escoger la más adecuada y no estar limitado en opciones [33].

En este punto también se tiene en cuenta si las herramientas de visualización accesibles por los distintos motores analíticos son lo suficientemente avanzadas (el estudio en profundidad de las herramientas de visualización se realizará en el apartado 2.6).

- Capacidad MOLAP: es decir, capacidad OLAP multidimensional. Todos los datos se almacenan en estructuras multidimensionales lo que aumenta significativamente la velocidad de procesamiento de las consultas. Existen motores analíticos que no utilizan MOLAP, sin embargo, se ha establecido como un requisito para que sirva como límite de calidad de procesamiento para este estudio.
- Sintaxis de consulta simple: se requiere un lenguaje de consulta cómodo y de fácil empleo por el cliente. Por este motivo, la capacidad de empleo del lenguaje SQL (*Standard Query Language*) supone puntos a favor, al tratarse del lenguaje estándar para los sistemas de gestión de bases de datos relacionales (SGBD), según ANSI (*American National Standards Institute*) [34].
- Compatible con Apache Storm: ha de ser capaz de asimilar y utilizar la salida de Apache Storm de manera simple, que a su vez obtiene los datos del flujo AIS, para procesar y gestionar la información y presentarla en un panel de control compatible. Se acepta la compatibilidad a través del empleo de Apache Kafka.

Tras un estudio del mercado en profundidad, se reduce el número de opciones de interés, por cumplir al menos dos de las características expuestas, a las siguientes expuestas en la Tabla 2-1.

	Empresa	Ultima versión disponible	Licencia software
Kyvos	Kyvos Insight	2020.1	No Libre
Mondrian OLAP server	Pentaho	3.7	EPL
Apache Kylin	Apache	3.1.0	Apache 2.0
Apache Druid	Apache	0.19.0	Apache 2.0
Apache Pinot	Apache	0.6.0	Apache 2.0
Essbase	Oracle	21.1	No Libre
icCube	icCube	6.8.9	No Libre
ClickHouse	Yandex	20.5.2.7	Apache 2.0

Tabla 2-1 Motores analíticos de interés

En la Tabla 2-2 se puede observar la concordancia de estas herramientas con las cualidades citadas anteriormente. Se indica en rojo aquellos motores analíticos que no cumplen el requisito señalado y en verde los que sí. También se indica, en la columna correspondiente, el lenguaje de consultas empleado y ejemplos de las herramientas de visualización. Por último, se ha añadido una columna en la que se valora la sencillez de uso de la herramienta, así como la comodidad que presenta trabajar con su interfaz. Esta última columna es un valor subjetivo y se califica en tres colores según el grado de conocimiento requerido sobre el programa para su uso: verde (usuarios de conocimiento básico), amarillo (requiere un conocimiento algo más avanzado) y rojo (conocimiento avanzado).

	<i>Open-source</i>	Herramientas de visualización open-source	Fácil compatibilidad con Apache Storm (Kafka)	Lenguaje de consultas	MOLAP	Sencillez de uso e interfaz (subjetivo)
Kyvos		Power BI		SQL		
Mondrian OLAP server		Pentaho		XML		
Apache Kylin		Superset, Zeppelin		SQL		
Apache Druid		Superset, Pivot		Druid SQL		
Apache Pinot		Superset		Pinot SQL		
Essbase				MDX		
icCube				MDX		
ClickHouse		Zeppelin, Superset		SQL		

Tabla 2-2 Comparativa de motores analíticos

2.5.1 Descripción de los motores analíticos

Basándonos en la Tabla 2-2 es observable que los motores analíticos que van a resultar de mayor interés son Apache Kylin, Druid, Pinot y ClickHouse. A continuación, se realiza un estudio más en profundidad de dichas herramientas con el objetivo de obtener un mayor conocimiento de ellas y poder alcanzar una elección acertada.

1. Apache Druid

Apache Druid [36] es un sistema de análisis de bases de datos en tiempo real diseñado para realizar consultas OLAP en grandes volúmenes de datos. Es un sistema de código abierto escrito en Java. Druid está especializado en el trabajo con información orientada a eventos (cambios de estado de los datos).

Druid se inició en 2011 con el objetivo de servir de impulsor de la herramienta de análisis de Metamarkets [37]. Comenzó como un proyecto bajo la licencia libre GPL (*General Public License*) pero pronto pasó a formar parte de la licencia software Apache 2.0.

En la actualidad, algunas de las compañías que utilizan Druid son Alibaba (empresa de tecnología multinacional china), Airbnb, Cisco, eBay y Netflix.



Figura 2-7 Logotipo de Apache Druid [36]

Las características principales de Druid quedan explicadas mediante la exposición de su funcionamiento y arquitectura [38], siguiendo el esquema de la Figura 2-8.

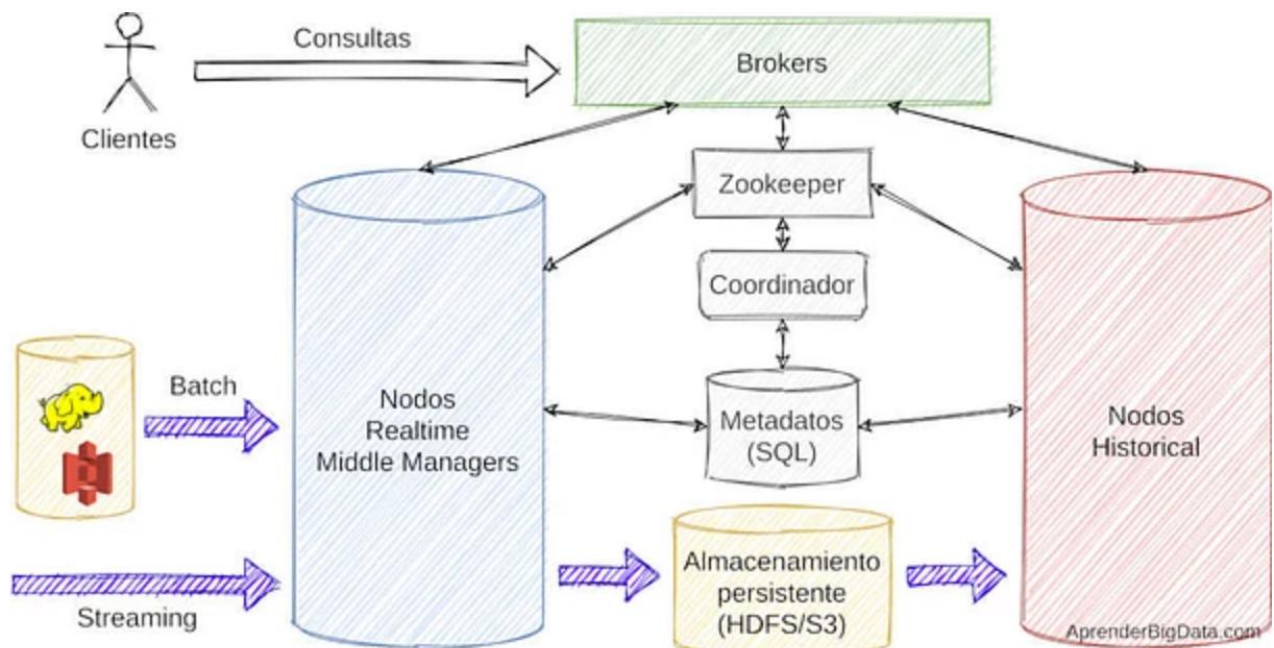


Figura 2-8 Arquitectura de Apache Druid [38]

La arquitectura de Apache Druid se basa en microservicios, agrupados en los servicios de ingesta, consulta y coordinación.

Los nodos *realtime* son los encargados de gestionar las lecturas y escrituras de *streaming* (en tiempo real) y de almacenar los datos en segmentos. Sobre ellos se calcularán los índices de clave primaria (para agilizar las búsquedas cuando las consultas entren dentro del rango de estas claves).

Este sistema está orientado a columnas [39], es decir, a diferencia del método convencional de ordenar los datos almacenándolos en filas (Figura 2-9), Druid estructura los datos en columnas almacenando los valores de diferentes columnas por separado (Figura 2-10).

El enfoque en fila suele generar problemas de desestructuración en los bloques de almacenamiento de discos duros. Si un bloque es lo suficiente grande puede almacenar varias filas conjuntas, y si este es demasiado corto se requerirá de varios bloques para una sola fila.

En el almacenamiento enfocado a columnas, el hecho de almacenar datos de un mismo tipo en cada bloque permite una mayor agilidad a la hora de realización de consultas, se reduce el número de entradas y salidas reduciendo así la latencia. Druid mantiene las latencias de consultas entre unos pocos segundos y sub-segundos es decir, en tiempo real. Además, también se hace posible la reducción del tamaño ocupado (reduciendo así el consumo de espacio de almacenamiento) mediante esquemas de compresión específicos para cada tipo de datos.

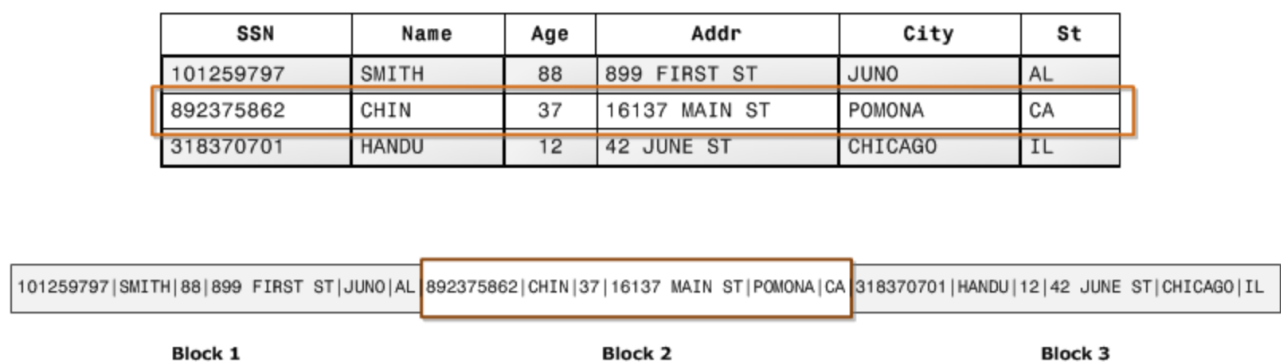


Figura 2-9 Modelo de almacenamiento orientado a filas [39]

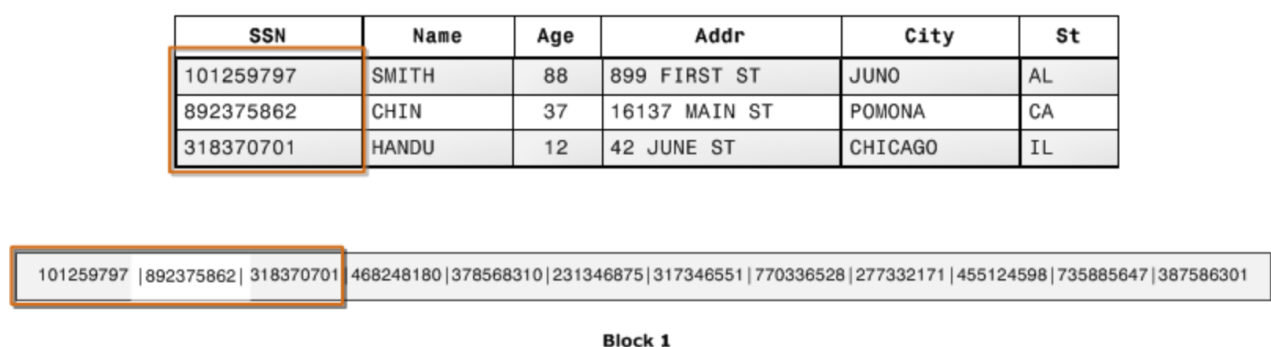


Figura 2-10 Modelo de almacenamiento orientado a columnas [39]

Posteriormente, los segmentos se almacenan en HDFS (*Hadoop Distributed File System*), que es el sistema de ficheros distribuidos de Hadoop, dentro del proyecto Apache, y sus correspondientes metadatos en una base de datos relacional para mantenerlos disponibles para otros nodos.

Los nodos históricos son los encargados de las consultas analíticas realizadas al clúster, cargando los segmentos almacenados anteriormente en HDFS. Algunos de estos segmentos son almacenados en caché para acelerar las consultas.

Los *brokers* son los nodos que gestionan las consultas de los usuarios. Redirigen estas al nodo que corresponda (*realtime* o histórico) para posteriormente agregar las respuestas.

El coordinador se encarga de coordinar los históricos e indicar cuándo desalojar, cargar, compactar o replicar los datos. Además, se encargan también de gestionar el balance de la carga entre los nodos.

Apache Druid utiliza Apache Zookeeper [40] para esta coordinación de los nodos del clúster.

Druid cuenta con la capacidad de procesar los datos de dos maneras distintas: por lotes (*batch processing*), aplicable a grandes volúmenes de datos no continuos; o en tiempo real, análisis de flujos de datos continuos de manera inmediata a la *query* (consulta).

Druid destaca por su escalabilidad gracias al procesamiento distribuido en múltiples servidores (llegando hasta centenas de servidores), haciéndolo muy tolerante a fallos y altamente fiable ya que el tiempo de inactividad de un solo nodo o de todo un centro de datos no afectará a la disponibilidad del sistema para las consultas.

También es capaz de la paralelización de procesamiento para una sola consulta mediante el empleo de múltiples núcleos.

El lenguaje nativo empleado en Druid es JSON (de JavaScript) pero empleando *Calcite Druid adapter*, un adaptador de Apache Calcite (*framework* procesador de datos) [41], Druid puede incorporar una capa denominada Druid SQL que sirve para traducir consultas SQL en consultas nativas de Druid.

2. Apache Kylin

Apache Kylin [42] es un motor de análisis distribuido de código abierto (escrito en Java) diseñado para proporcionar una interfaz SQL y análisis multidimensional OLAP sobre Hadoop [43] caracterizado por su alta velocidad de consulta aparentemente constante e independiente al volumen de datos. Kylin realiza el análisis de miles de millones de filas con latencia inferiores al segundo y, por tanto, se puede decir que trabaja en tiempo real. Kylin dispone de licencia de software Apache 2.0.

Apache Kylin comenzó en 2015 como un proyecto de la empresa eBay (empresa dedicada a la subasta y comercio electrónico) por su departamento de R&D (*Research and Development*) en Shanghai, China.

En la actualidad, entre las muchas compañías que emplean Kylin se encuentran: eBay, Samsung, XiaoMi y Yahoo! Japan.



Figura 2-11 Logotipo Apache Kylin [42]

Para entender el funcionamiento de Kylin es necesario antes comprender el concepto de Cubo OLAP.

Un Cubo OLAP es un método de almacenamiento que proporciona una presentación multidimensional (de ahí el concepto de cubo) de los datos, mejorando su capacidad respecto a las tablas tradicionales de las bases de datos relacionales. Estos cubos están formados por dimensiones y medidas. Las medidas son los valores cuantitativos que se pueden medir de los datos. Las dimensiones hacen referencia a campos determinados de un tipo de dato específico permitiendo una fácil localización de los datos y, por tanto, una latencia mínima. La información obtenida de ese campo se podrá comparar y complementar además con la de otras dimensiones, haciendo posible la evaluación en profundidad de la información.

De esta manera, este servidor OLAP ofrecen la posibilidad de crear bases de datos completas y con rápido acceso a la información [44]. En la Figura 2-12 se muestra una simplificación de la metodología de un Cubo OLAP para su mejor comprensión. En este ejemplo, los valores de Importe son la medida y la definición del tipo de producto, empleado y moneda son las dimensiones.

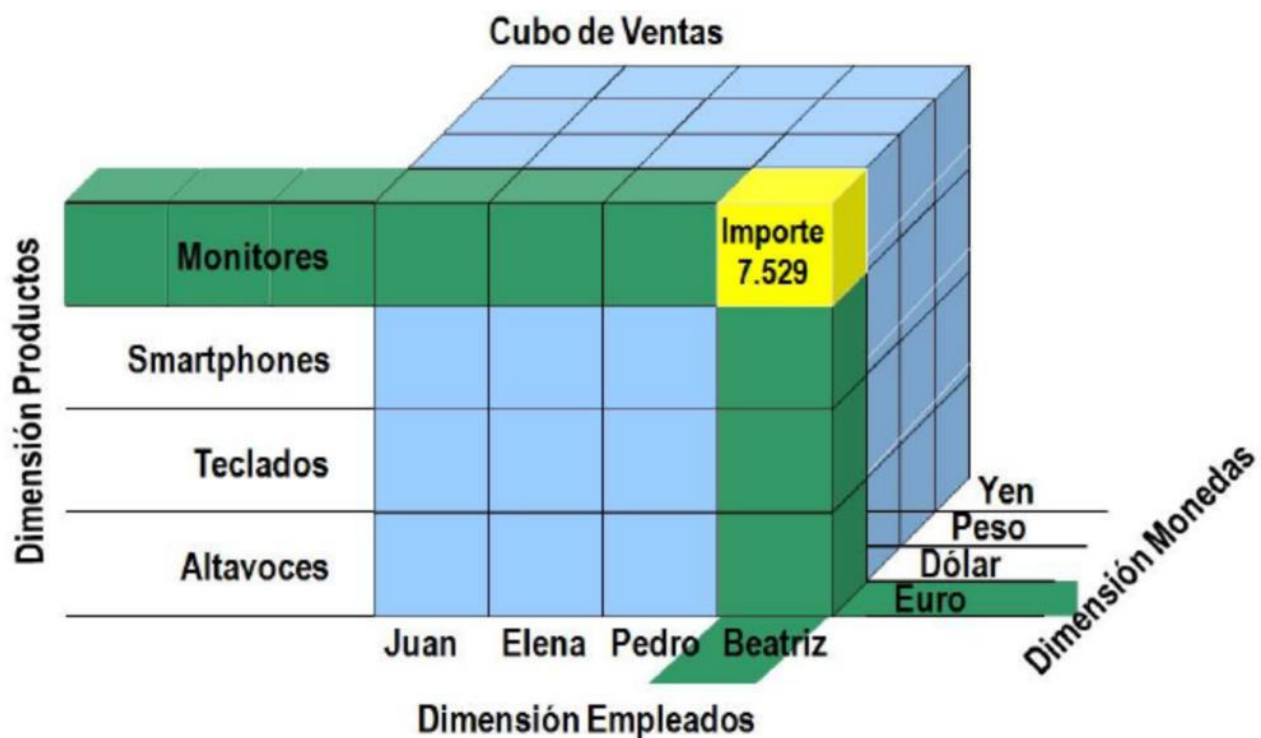


Figura 2-12 Ejemplo de metodología de un Cubo OLAP [45]

La ventaja proporcionada por Kylin es la capacidad de creación de múltiples cubos distribuidos en un clúster, frente al concepto de generar un único cubo de los métodos tradicionales, así mezclando las tecnologías de *Big Data* con el concepto OLAP. Esto permite a Kylin una gran escalabilidad horizontal sin limitaciones en su capacidad de almacenamiento.

Con este concepto en mente se procede a explicar la arquitectura [38] y características fundamentales de Kylin.

Kylin soporta varias fuentes de datos como Apache Hive [46] o Apache Kafka, de donde obtiene las tablas para la generación de los cubos. A partir de estos registros de datos, que se suelen encontrar en formato de sistema relacional basado en filas, se inician los precálculos de los Cubos OLAP mediante el proceso *map-shuffle-reduce*, mostrado en la Figura 2-13, obteniendo así un sistema no relacional basado en conjuntos clave-valor (siendo las claves identificadores únicos). El proceso *map-shuffle-reduce* consiste en identificar datos de un mismo tipo, agruparlos en un mismo bloque y realizar operaciones de compresión específicas para ese tipo de dato en concreto.

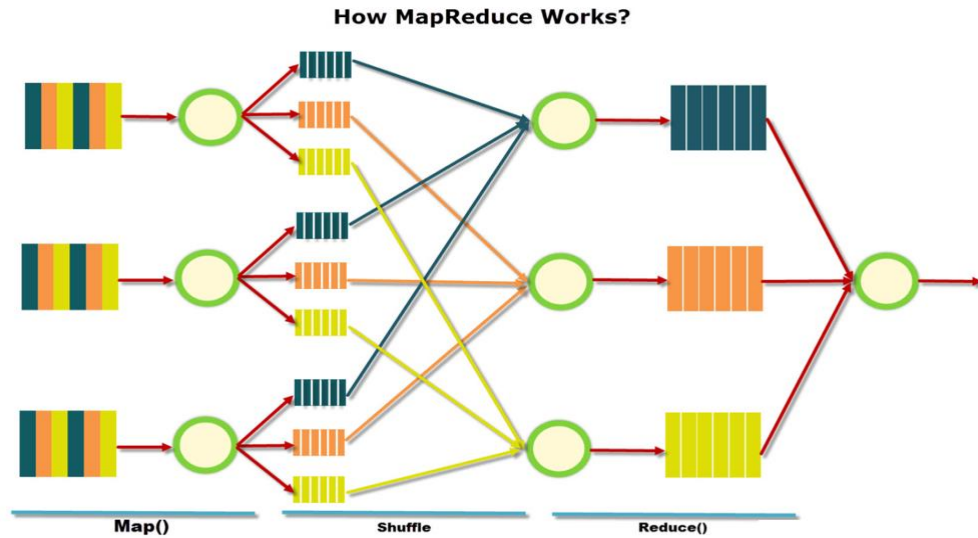


Figura 2-13 Esquema del proceso *map-shuffle-reduce* [47]

Kylin utiliza Hadoop o Spark [48] como motores de procesamiento para la agregación de los Cubos OLAP previamente calculados.

Por último, Kylin se apoya en HBase [49] para almacenar los Cubos OLAP, obteniendo así latencias de consulta de milisegundos (en entornos distribuidos).

Kylin puede trabajar tanto en lotes como en tiempo real. Para las consultas en tiempo real, el servidor de consultas de Kylin determina si los datos buscados se encuentran en el histórico (HBase) o si aún no han sido precalculados. En este ultimo caso, Kylin cuenta con un nodo *realtime*, similar al descrito en el apartado dedicado a Apache Druid, que realiza la búsqueda de los datos requeridos en Hive. Hive emplea un esquema de almacenamiento en estrella por su simplicidad y velocidad para el ámbito multidimensional. Un esquema en estrella está conformado por una tabla de hechos, que contiene los datos medibles y cuantitativos, rodeada de diversas tablas de dimensiones, que indican los atributos de los datos contenidos en la tabla de hechos [50]. En la Figura 2-14 se puede observar un ejemplo de un esquema en estrella. En este caso la tabla de hechos es *dw_ventasfact* y el resto son tablas de dimensiones.

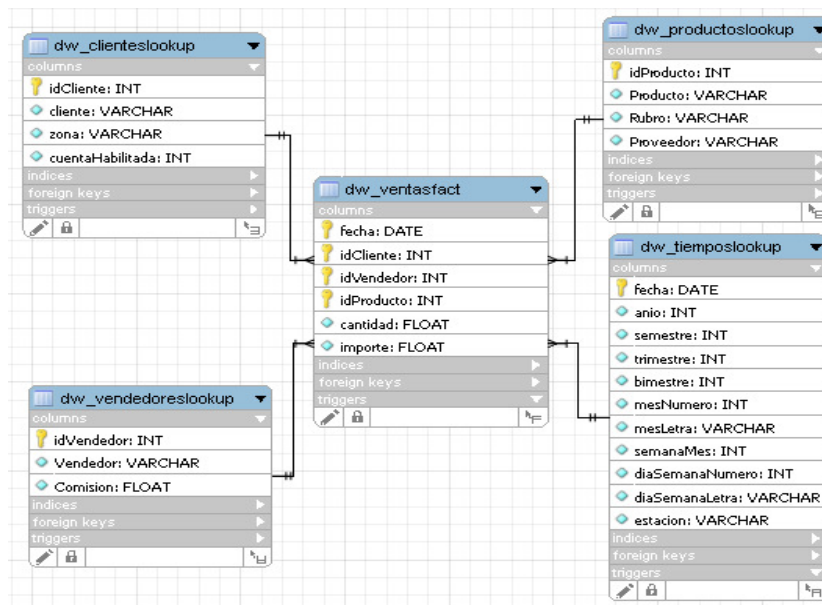


Figura 2-14 Esquema en estrella [51]

Para resumir, el proceso OLAP en tiempo real consiste en realizar los precálculos del cubo a medida que llegan los datos al sistema. En caso de entrar la consulta en el rango de datos de los cubos almacenados en Hbase, el resultado se generará en milisegundos. En caso contrario, muy poco probable, se realizará el análisis en Hive lo cual supondrá una espera de segundos.

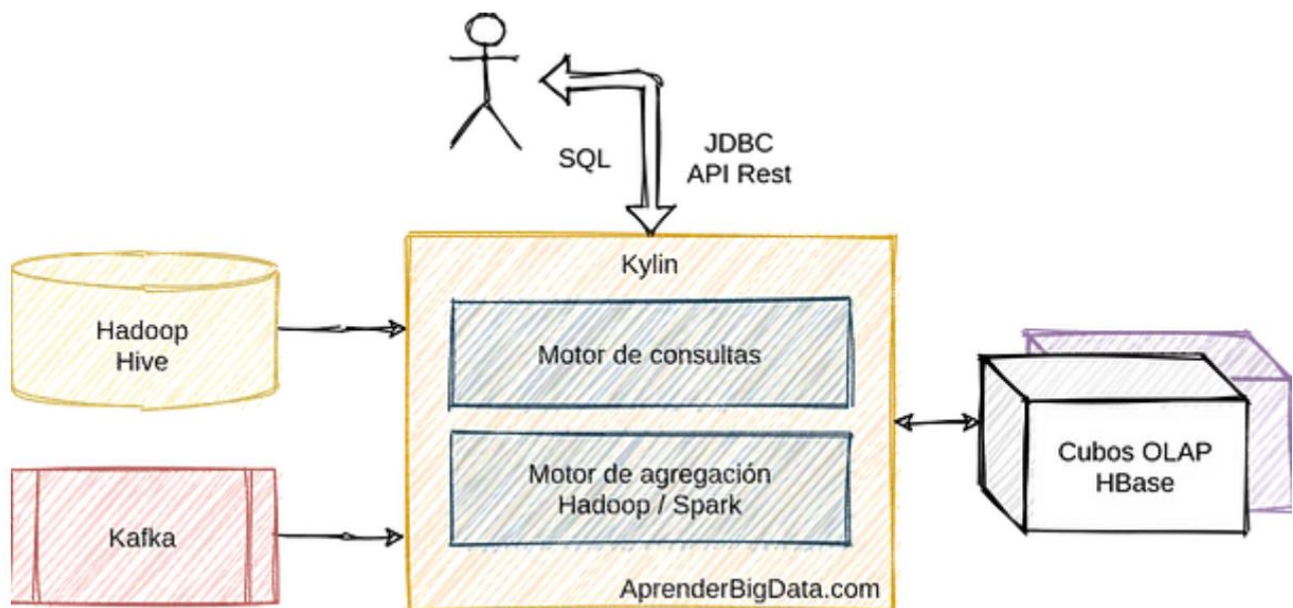


Figura 2-15 Arquitectura de Kylin [38]

En cuanto al lenguaje empleado para las consultas, como se ha mencionado en la introducción a la herramienta, Kylin ofrece una interfaz ANSI SQL de fácil uso y sencilla de administrar.

3. Apache Pinot

Apache Pinot [52] es un motor analítico distribuido OLAP de código abierto escrito en Java y con licencia software Apache 2.0.

Apache Pinot se creó para que sirviese de motor analítico de LinkedIn (red social orientada a los negocios y uso empresarial) y desde entonces muchas son las empresas que lo emplean entre las que destaca Uber.



Figura 2-16 Logotipo de Pinot [52]

En cuanto a la arquitectura de Pinot, esta es prácticamente idéntica a la de Druid (como se puede observar en la

Figura 2-17). Como describe [53], ambos son como dos implementaciones desarrolladas de forma independiente de exactamente el mismo sistema.

La principal diferencia es que Druid mantiene una copia de los segmentos de datos (metadatos) almacenados en una base relacional SQL permitiendo que se pueda recuperar la información de un segmento que haya sido desalojado del proceso de consultas (por exceder un tiempo de permanencia, en

todos los motores analíticos se eliminan los segmentos viejos). Esto además supone una oportunidad de liberar parte de su dependencia en la labor de Zookeeper del control de los nodos históricos y en líneas futuras poder llegar a trabajar de manera independiente a este.

En contraste, Pinot delega gran parte de la gestión y control de los segmentos y clústers en el *framework* Helix [54] que proporciona una potentísima capacidad de reasignación, expansión, reconfiguración y, en definitiva, gestión de los nodos. Es ahí donde se conservan los metadatos de los segmentos. Helix trabaja sobre Zookeeper y, por tanto, los metadatos se alojan realmente en este, lo cual supone una dependencia directa en Zookeeper.

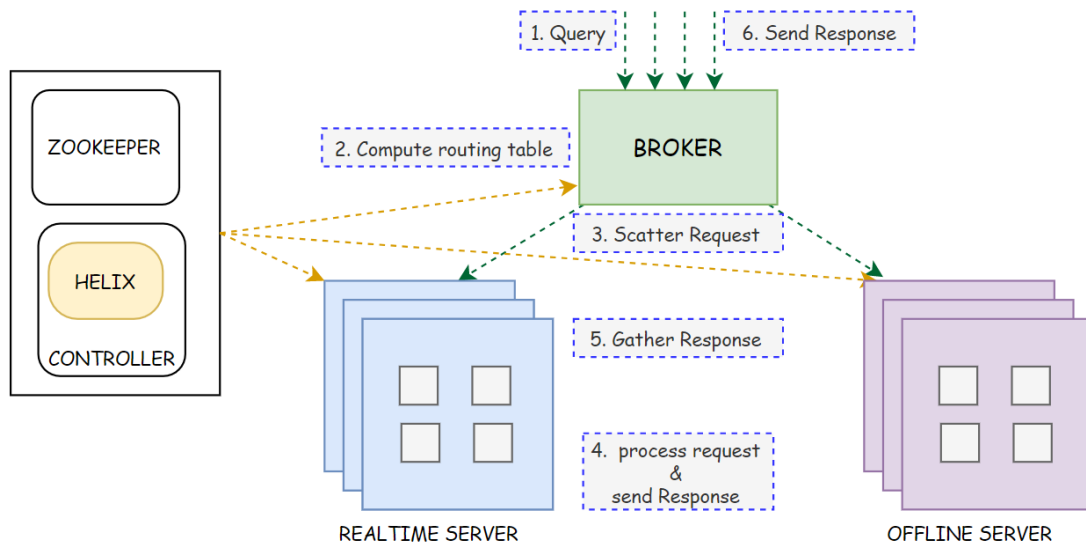


Figura 2-17 Proceso de consulta en Apache Pinot [52]

El lenguaje de consultas usado por Pinot es una variante de SQL denominada PQL (*Pinot Query Language*), con una semántica prácticamente idéntica pero con limitaciones en el conjunto de consultas disponibles, limitándolas únicamente a consultas simples que le permitan mantener una latencia baja (menos de un segundo).

4. ClickHouse

Se trata de un DBMS (sistema gestor de base de datos) OLAP de código abierto (con Apache 2.0 como licencia software) escrito en C++.

ClickHouse [55] fue desarrollado por la empresa de tecnología rusa Yandex [56] para que sirviera como base a su herramienta Yandex Métrica (una herramienta de analítica web similar a Google Analytics). ClickHouse permite a Yandex Métrica generar informes personalizados en tiempo real a partir de datos no agregados (lo que supone cálculos *online*).

En la actualidad, algunas de las compañías que usan ClickHouse son: Aloha Browser (aplicación móvil), ArenaData (plataforma de datos), Amadeus (empresa de viajes) y Cloudflare (red de distribución de contenidos).



Figura 2-18 Logo de ClickHouse [55]

Entre las características principales de ClickHouse destacan las que se mencionan a continuación.

Al igual que Druid y Pinot, ClickHouse emplea almacenamiento orientado a columnas proporcionando al sistema de una gran velocidad de consultas (sub-segundos para las consultas simples y pocos segundos para las más complejas) y procesamiento.

En cuanto a su funcionamiento, ClickHouse dispone de una arquitectura muy similar a Druid y Pinot, sin embargo, existen ciertas diferencias que la convierten en una estructura más sencilla (con sus ventajas y desventajas) que se explicarán a continuación.

ClickHouse no requiere de un nodo central coordinador o controlador (como se puede observar en el esquema de la arquitectura de Druid en la Figura 2-8), es decir, no dispone de un servicio únicamente dedicado a la gestión de datos. Esto se consigue incluyendo "pesos" a los metadatos de las tablas particionadas de los datos recibidos, que indican qué porcentajes de datos deben ir a qué nodos.

Además, no se requiere de almacenamiento específico, como HDFS utilizado en Druid o Pinot, ya que los nodos no son solo responsables del procesamiento de consultas sino también de la persistencia de los datos almacenados en ellos.

Sin embargo, un inconveniente de esta metodología es que cuando un cluster de nodos crece rápidamente, los datos no se reequilibran sin la intervención manual de un humano.

En cuanto al lenguaje utilizado para la consulta es un lenguaje declarativo basado en SQL prácticamente idéntico al estándar ANSI SQL con limitaciones en el número de consultas disponibles.

2.5.2 Conclusión

Tras este estudio en profundidad comparando las diferentes herramientas, respaldado por fuentes como [57] y [58], se muestra en la Tabla 2-3 una serie de criterios básicos sobre los cuales se sustenta la elección final.

Estos criterios base son los siguientes:

- Velocidad de consulta.
- Soporte SQL.
- Integración de herramientas de visualización.
- Consultas complejas.

	<u>Apache Druid/Pinot</u>	<u>Apache Kylin</u>	<u>ClickHouse</u>
Velocidad de consulta	Muy rápido	Rápido	Rápido
Soporte SQL	Limitado	Bueno	Limitado
Integración de herramientas de visualización	Limitado	Múltiple	Múltiple
Consultas complejas	Mejorable para grandes conjuntos de datos	Buena gestión gracias a cubos MOLAP	Mejorable para grandes conjuntos de datos

Tabla 2-3 Comparativa final de motores analíticos

Tras la comparativa, se elige Apache Kylin por ser una herramienta OLAP líder adecuada para grandes volúmenes de datos, proporcionar una gran velocidad de consulta (incluso para consultas que resultan más complejas) gracias a la tecnología de los cubos MOLAP almacenando información en Hbase, ofrecer gran variedad de posibilidades a la hora de escoger una herramienta de generación de

dashboards, emplear SQL como lenguaje para realizar las consultas sin las limitaciones observables en las otras herramientas y por mostrar una gran compatibilidad con Kafka.

También se ha tenido en cuenta que presenta una cómoda interfaz (sencilla de manipular) y gran variedad de documentación de cómo configurarlo e implementarlo en diversas plataformas.

2.6 Revisión de herramientas de visualización

Una vez establecido como motor analítico Apache Kylin, el siguiente paso a dar es la elección de una herramienta de visualización y generación de *dashboards* adecuada. La visualización de los datos, como se ha mencionado en el apartado 2.2.1, es una de las 7 Vs del *Big Data* y cobra especial importancia en la consecución del objetivo del presente trabajo ya que es necesaria una presentación ordenada y armoniosa que permita una fácil interpretación de los datos expuestos para la rápida toma de decisiones.

En la actualidad abundan las aplicaciones, plataformas y herramientas especializadas en la creación de paneles de control y es necesaria la realización de un estado del arte de estas ya que todas presentan cualidades y características que las hacen diferentes al resto. A la hora de realizar el estudio, se ha tenido en cuenta una serie de criterios y características base sobre los cuales se ha sustentado la decisión final. Los criterios base a los que se presta especial atención son los siguientes:

- Demostrada compatibilidad con Apache Kylin: se busca una herramienta capaz de realizar consultas directamente de Kylin sin necesidad de pasos intermedios para poder hacer una exploración de los datos de los cubos Kylin generados y crear un cuadro de mando que se pueda compartir con los usuarios finales.
- *Open-source*: la herramienta debe ser *open source*. Las ventajas de imponer este requisito quedan explicadas al inicio del apartado 2.5.
- Capacidad de realizar operaciones simples como sumas, productos, mezclar fuentes de datos en mismos gráficos y tablas, etc.
- Limitación en el número de *dashboards*.
- Limitación en el número de usuarios.
- Aprendizaje y uso sencillo.
- Aplicación web.

2.6.1 Descripción de las herramientas de visualización

Resulta necesario ahondar en las capacidades y características principales de las múltiples herramientas que ofrece el mercado para obtener una descripción clara de cada una de ellas y entender qué servicios son capaces de satisfacer en relación con el producto que se desea obtener.

De entre la gran variedad de opciones disponibles se han seleccionado las más populares y de mayor mención en comparativas de software de generación de paneles de control, así como las que han demostrado ya excelentes resultados al haber sido implementadas por empresas de renombre.

1. Cyfe

Cyfe [59] es una aplicación de creación de *dashboards* basada en la nube y que está diseñada para analizar, transformar y presentar los datos adquiridos de diferentes fuentes (véase Figura 2-19).

Esta aplicación es definida por diferentes páginas web como una de las herramientas más completas para la creación de cuadros de mando avanzados [60] [61].

Tiene capacidad de conexión a una gran variedad de fuentes diferentes y permite el empleo de diferentes *widgets* independientes para cada conjunto de datos que se quiera presentar dotando a la aplicación de una gran capacidad de personalización de sus *dashboards*.

Posee un modo de uso TV, en el cual el panel de control se sitúa en pantalla completa dando una visión TV para la monitorización de los datos.

Sin embargo, la prueba gratuita que ofrece está muy limitada tanto en el número de usuarios (solo permite uno) como en el de *dashboards* (permite hasta dos). Tampoco permite la disponibilidad de todas las opciones de creación de *dashboards* y su duración es de quince días.

El precio de la versión completa es de 89\$ al mes y proporciona un límite de 20 *dashboards* a un número de clientes ilimitado.

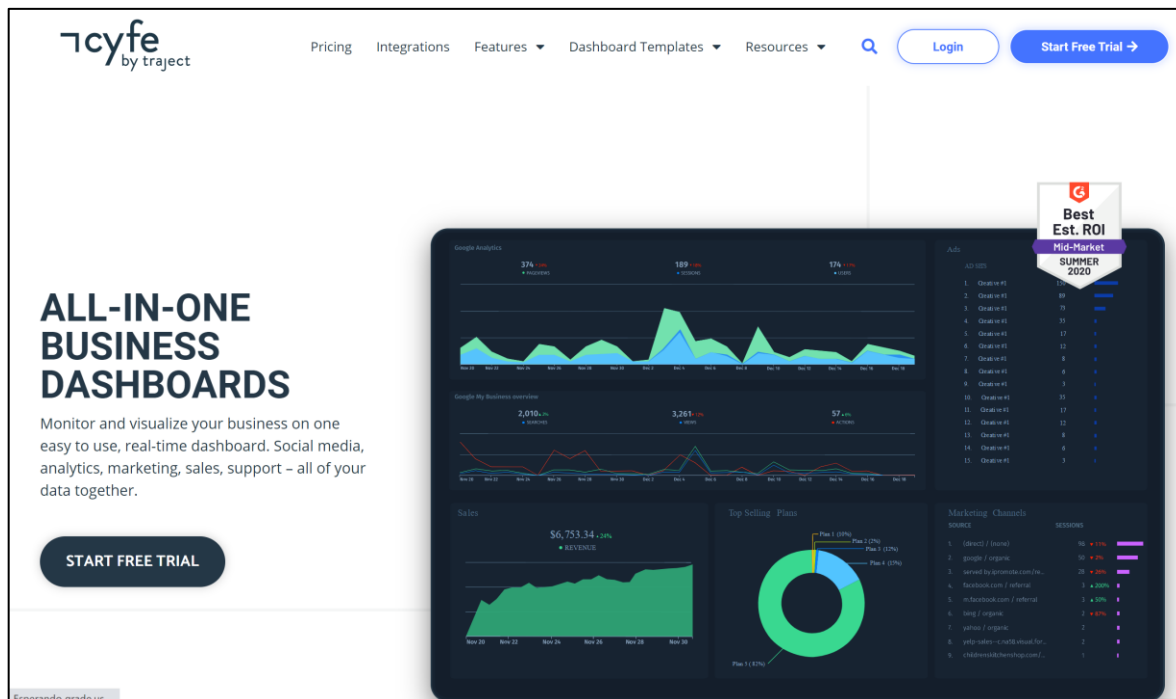


Figura 2-19 Página principal del sitio web de Cyfe

2. Klipfolio

Klipfolio [62] es una aplicación *online* dedicada a la creación de paneles de negocios en tiempo real en diversas plataformas (navegadores web, monitores TV, dispositivos móviles, etc.) y ofrece una gran combinación de opciones desde las que importar los datos (bases de datos, la nube, el propio equipo, etc.) (véase Figura 2-20).

Klipfolio destaca por la gran cantidad de plantillas preconstruidas que proporciona (más de 300), aunque también permite la creación de *dashboards* completamente personalizados desde cero, así como por la gran interacción que permite al usuario con todas las gráficas y tablas.

Su funcionamiento se basa en el empleo de *Klips* y *PowerMetrics*. Los primeros son tablas y gráficos de datos que pueden ser agregados y calculados mediante fórmulas, fácilmente editables. Los segundos son valores cuantificables utilizados para evaluar el rendimiento en las diferentes áreas a lo largo del tiempo o para estudiar una determinada tendencia.

Klipfolio dispone de una versión gratuita, de catorce días de duración, sin límites de uso con prácticamente las mismas capacidades que la versión de pago. Sin embargo, la versión de pago más completa llega hasta los 399 \$/mes proporcionando una capacidad de generación de 60 *dashboards* y usuarios ilimitados. Versiones más económicas reducen sus capacidades tanto en número de *dashboards* como de usuarios permitidos.

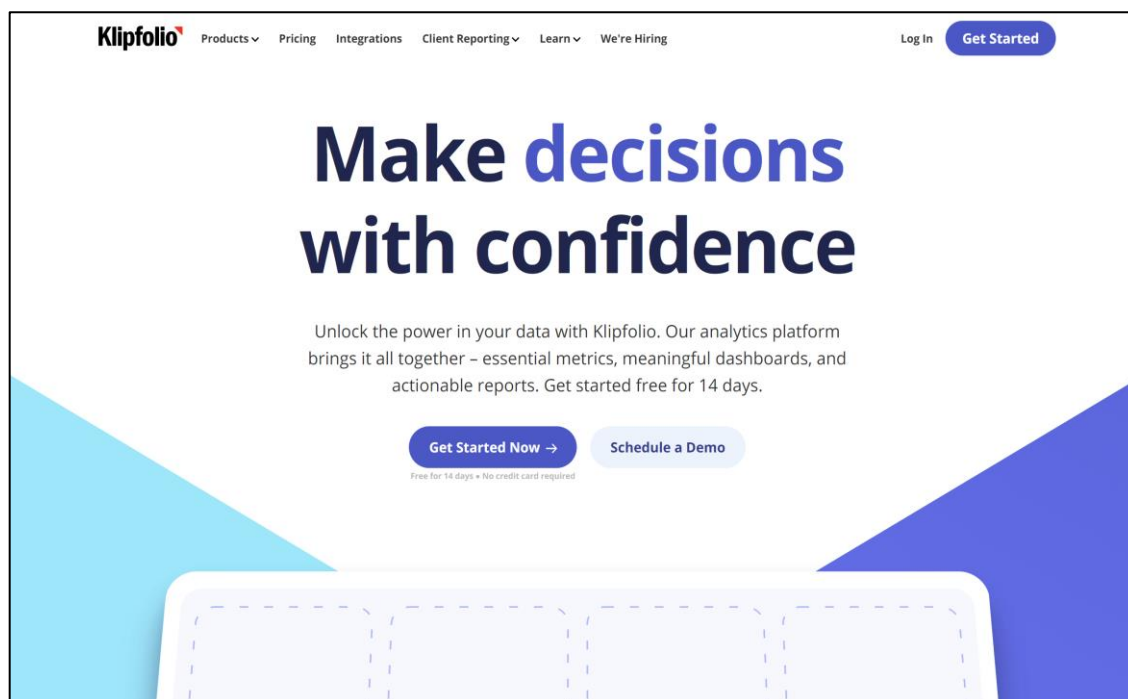


Figura 2-20 Página principal del sitio web de Klipfolio

3. Geckoboard

Geckoboard [63] es una herramienta de creación de paneles de control en tiempo real (véase Figura 2-21). Permite manejar datos de todo tipo de fuentes y actualizar de forma automática estos.

Se caracteriza por ofrecer más de 100 *widgets* diferentes para la creación de *dashboards* así como la aplicación de una gran variedad de potentes filtros para la clasificación de la información a la hora de la presentación.

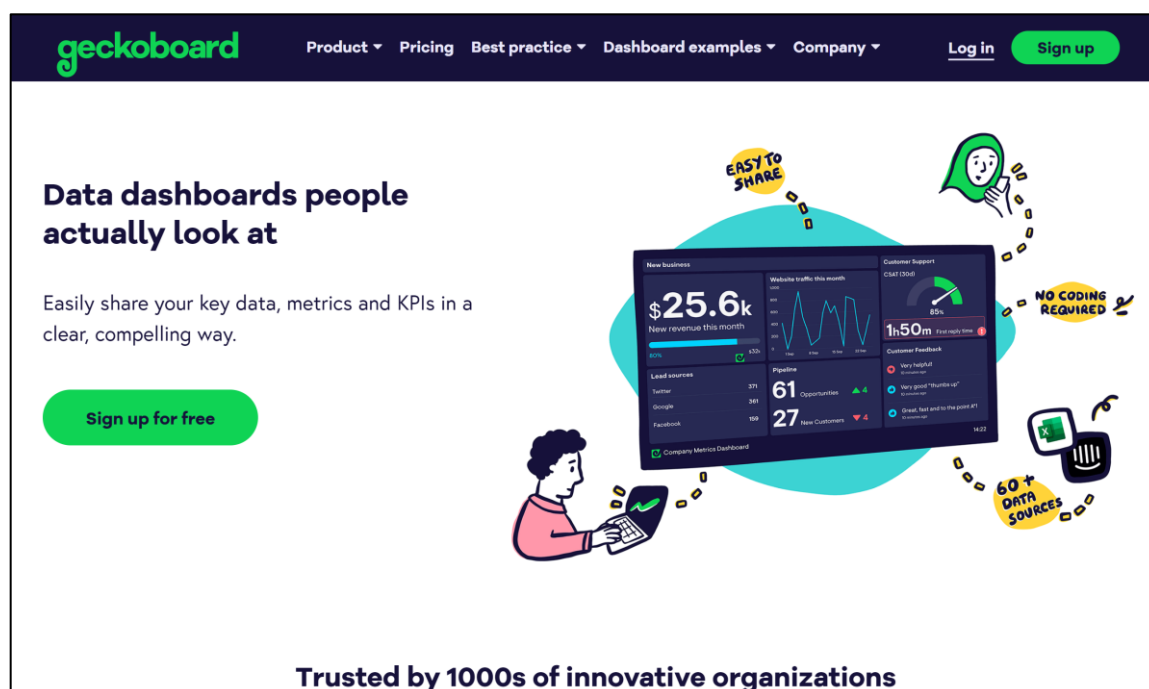


Figura 2-21 Página principal del sitio web de Geckoboard

Según [61] se sitúa en un punto intermedio entre lo básico y lo avanzado. Es de uso sencillo y no requiere de conocimientos avanzados para poder realizar paneles de control de calidad. Dispone de una versión gratuita de prueba de treinta días. La versión empresarial es la más completa y dispone de un número ilimitado tanto de usuarios como de *dashboards*. Sin embargo, esta versión llega hasta los 500\$ al mes. En otras versiones que ofrece la compañía el precio se reduce, pero también lo hacen las capacidades y el número de usuarios y *dashboards* permitidos.

4. Apache Superset

Apache Superset [64] es una herramienta *open-source* de visualización de datos moderna desarrollada por Airbnb (véase Figura 2-22). Cuenta con un gran repertorio de funciones que le permiten al usuario explorar, filtrar y organizar los datos para crear *dashboards* altamente personalizados. También destaca por la gran variedad de representaciones gráficas que ofrece.

Superset supone una alternativa muy atractiva ya que permite el manejo, organización y exploración de cada uno de los paneles generados de forma detallada. Además, hace posible la exploración de datos de los cubos Kylin mediante el lenguaje SQL.

Otra característica que hace de esta herramienta una opción de gran interés para el presente trabajo es la seguridad que proporciona. Superset permite definir una lista de usuarios y otra de funcionalidades predeterminadas pudiendo asociar ambas como al administrador le plazca, permitiendo a determinados conjuntos de usuarios acceder a ciertas herramientas y vistas o no. También permite ver las estadísticas de los distintos usuarios, así como su registro de acciones.

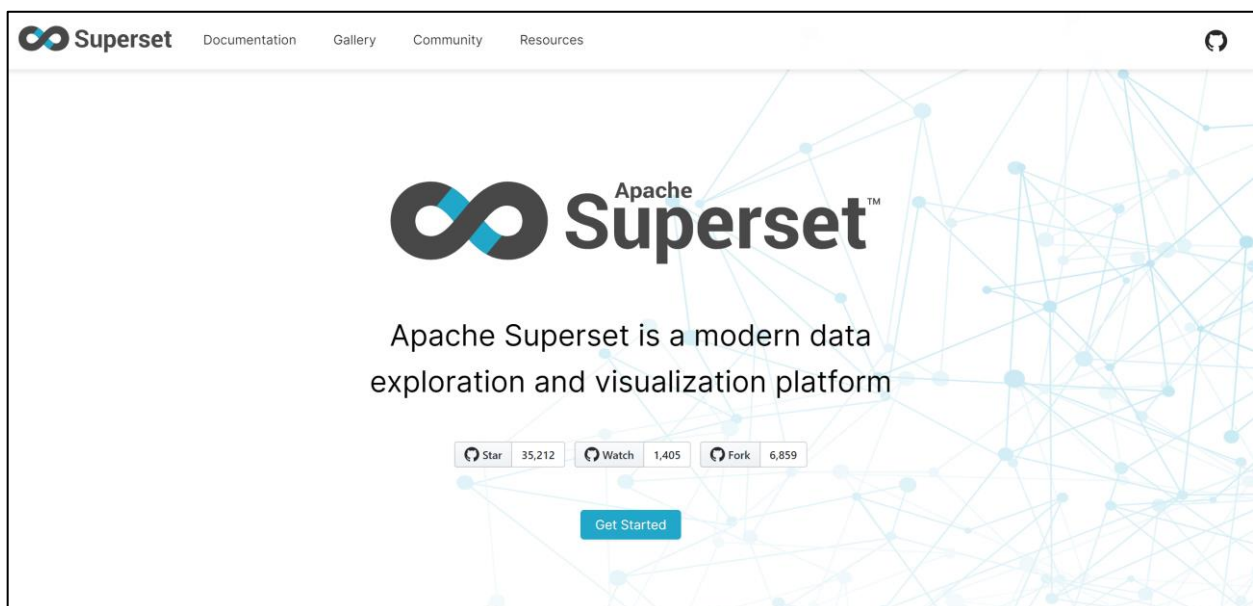


Figura 2-22 Página principal del sitio web de Superset

Para probar su funcionamiento, existe una demo online [65] que muestra como con Kylin y Superset se puede generar y publicar un cuadro de mando que usa como fuente un conjunto de datos de tipo *Big Data* relativos al rendimiento académico de una universidad en los últimos quince años. En esta demostración se cuenta con 100 millones de medidas como los créditos relativos a asignaturas aprobadas, suspendidas o matriculadas. Se muestra cómo el conjunto de herramientas permite analizar la información en base a distintos contextos de análisis como la calificación o el año académico.

5. Tableau

Tableau [66] es una plataforma de análisis integral y eficaz de datos (véase Figura 2-23). Se trata de una potente herramienta de exploración de datos que ofrece un elevado grado de detalle sobre la información que proporcionan los datos con los que se está trabajando. Tableau ofrece una gran variedad

de productos, no se limita a la creación de *dashboards*; sin embargo, los que se especializan en este campo son *Tableau Desktop*, *Tableau Online* y *Tableau Server*.

Tableau Desktop ofrece diversas opciones de diseño de *dashboards* para la representación de datos obtenidos de todas las fuentes comunes como Cloud apps, MySQL, Excel o Google Analytics. Pone a disposición de profesores y estudiantes una licencia gratuita de un año. Esta opción es fácilmente integrable con Apache Kylin, permitiendo una conexión directa para la realización de consultas SQL. Existe una demo similar a la de Superset que muestra la combinación de ambas herramientas [65].

Tableau Online se trata de una versión en la nube de la opción anterior, proporcionando las mismas capacidades de gestión y visualización además de su funcionamiento desde aplicaciones móviles.

Tableau Server es una herramienta similar a las dos anteriores, pero con un enfoque a su empleo dentro de organizaciones y requiere de su implementación específica.

Supone una opción segura, lo cual queda reflejado en su gran comunidad de usuarios activos, especialmente en el ámbito empresarial, gracias a la capacidad de fácil autenticación y control de los permisos de los usuarios, el contenido y los datos.

Tableau, sin embargo, no es una herramienta sencilla de usar ni facilita un rápido aprendizaje.

Los precios para obtener las versiones mencionadas sin limitaciones varían entre 35\$ y 70\$ al mes.

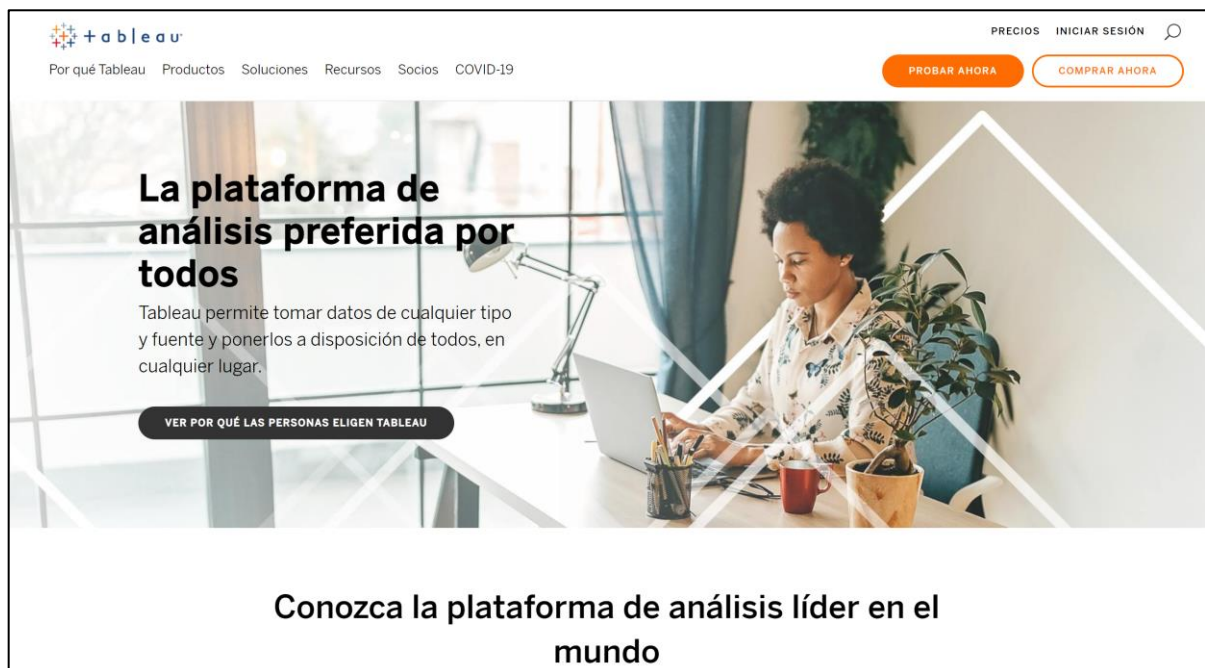


Figura 2-23 Página principal del sitio web de Tableau

6. Grafana

Grafana [30] es otra opción *open-source* para la generación de paneles de control (véase Figura 2-24). Permite la obtención de información de diversas fuentes como MySQL y Microsoft SQL Server. Esta herramienta permite realizar cambios en tiempo real a los *dashboards* generados.

El diseño y utilización de los paneles generados es muy sencillo e intuitivo. Emplea plugins para personalizar el número de paneles disponibles, siendo capaz de presentar resultados de varias fuentes de datos simultáneamente. También dispone de variables que funcionan como filtros para la representación selectiva de datos.

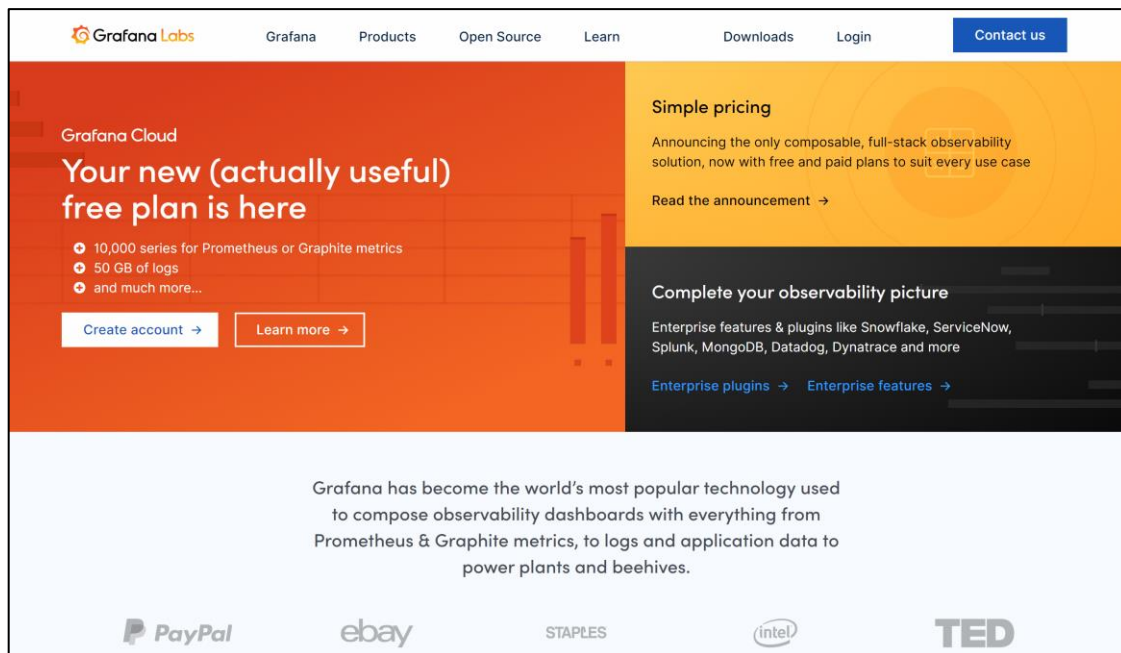


Figura 2-24 Página principal del sitio web de Grafana

Permite su instalación en diferentes sistemas operativos como Windows y Linux. También permite familiarizarse con todas sus características y funcionalidades en una demo sorprendentemente completa que presenta en su página web.

7. InetSoft

InetSoft [67] es una aplicación web (véase Figura 2-25) de visualización de datos enfocada en la obtención de datos de diversas fuentes para poder visualizarlos en un mismo panel de control (a esto se le denomina *mash-up*).

Esta herramienta destaca por su capacidad integrada de *machine learning* (aprendizaje automático) durante los procesos de análisis de los datos para la generación de *dashboards*. InetSoft genera informes automáticos con información sobre los datos analizados con formatos como Microsoft Word y Excel.

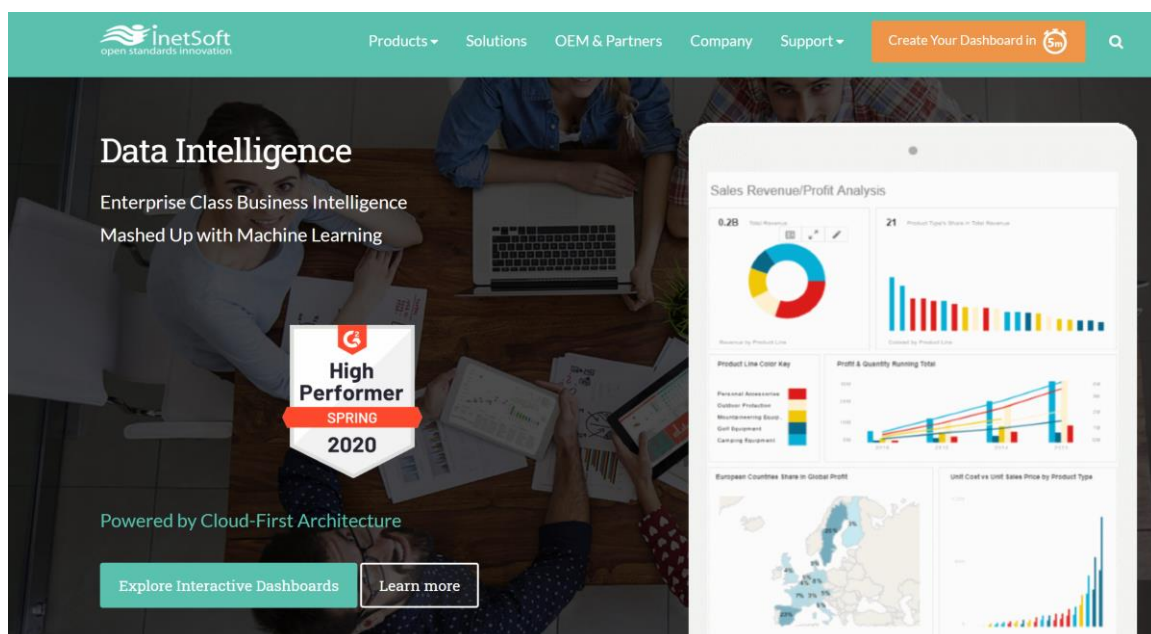


Figura 2-25 Página principal del sitio web de InetSoft

Esta herramienta supone una opción segura y así lo demuestran sus más de 5000 usuarios actuales y su uso en grandes organizaciones como la NASA o el Departamento de Defensa de los Estados Unidos. Además, proporciona una interfaz simple y de fácil aprendizaje. Es integrable en múltiples sistemas operativos tales como Windows, Linux y Mac.

Este software tiene un precio de 2800\$ al año (pago anual) y no impone límites en el número de usuarios.

8. Apache Zeppelin

Apache Zeppelin [68] es una herramienta web *open-source* de análisis de datos para su visualización en *Notebooks* (véase Figura 2-26). En un *Notebook* es posible analizar y exponer datos de distintas fuentes *Big Data* de forma textual o mediante el empleo de la gran variedad de gráficos y tablas que incluye. Estos *Notebooks* son personalizables y se pueden compartir vía URL con otros usuarios.

Apache Zeppelin hace posible la exploración de los datos de los cubos Kylin mediante lenguaje SQL y la creación de cuadros de mando. Esto se consigue de manera sencilla mediante la configuración de un intérprete de Kylin que permite conectarse a este desde la máquina en la que Zeppelin se haya instalado.

Existe una demo [65] similar a la mencionada para Superset y Tableau, en la que se muestra la arquitectura del sistema Kylin-Zeppelin, y además permite observar las diferentes formas de presentación de información que la herramienta proporciona mediante un ejemplo de cuadro de mando.

Zeppelin gestiona la conexión de datos de diversas fuentes, así como la generación de *Notebooks* de manera muy intuitiva y de fácil aprendizaje.

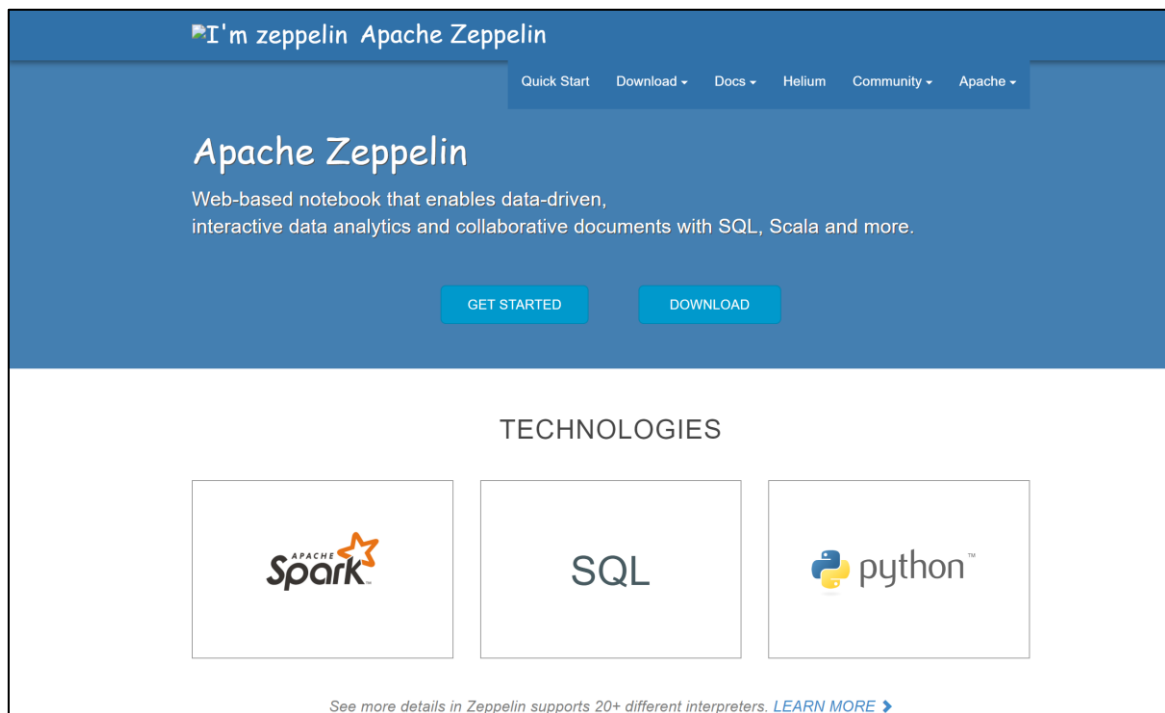


Figura 2-26 Página principal del sitio web de Zeppelin

9. Clicdata

Según varias fuentes en las que se comparan herramientas de visualización y creación de *dashboards*, como [60] y [61], Clicdata [69] es mucho más que una herramienta para crear *dashboards*. Presenta ciertas limitaciones para la creación de estos, pero tiene una excelente capacidad de resumir la información de inmensos volúmenes de datos y crear avanzadas infografías de estos.

Es de uso y aprendizaje sencillo, y dispone de una gran variedad de *widgets* personalizables que facilitan la interacción con esta herramienta.

Clicdata es instalable tanto en Windows como en Linux y facilita la capacidad de crear un almacén de datos propio para los usuarios, conectado a una gran variedad de fuentes como Dropbox o Drive. Proporciona además una política de protección de datos basada en cifrado.

Clicdata dispone de una versión gratuita con capacidad limitada de 1 GB para poder familiarizarse con el software (véase Figura 2-27). Dispone también de versiones por 71\$/mes que incluyen capacidad de creación de *dashboards* ilimitados. Su versión más completa llega hasta 890\$/mes y tiene una limitación de 100 usuarios.

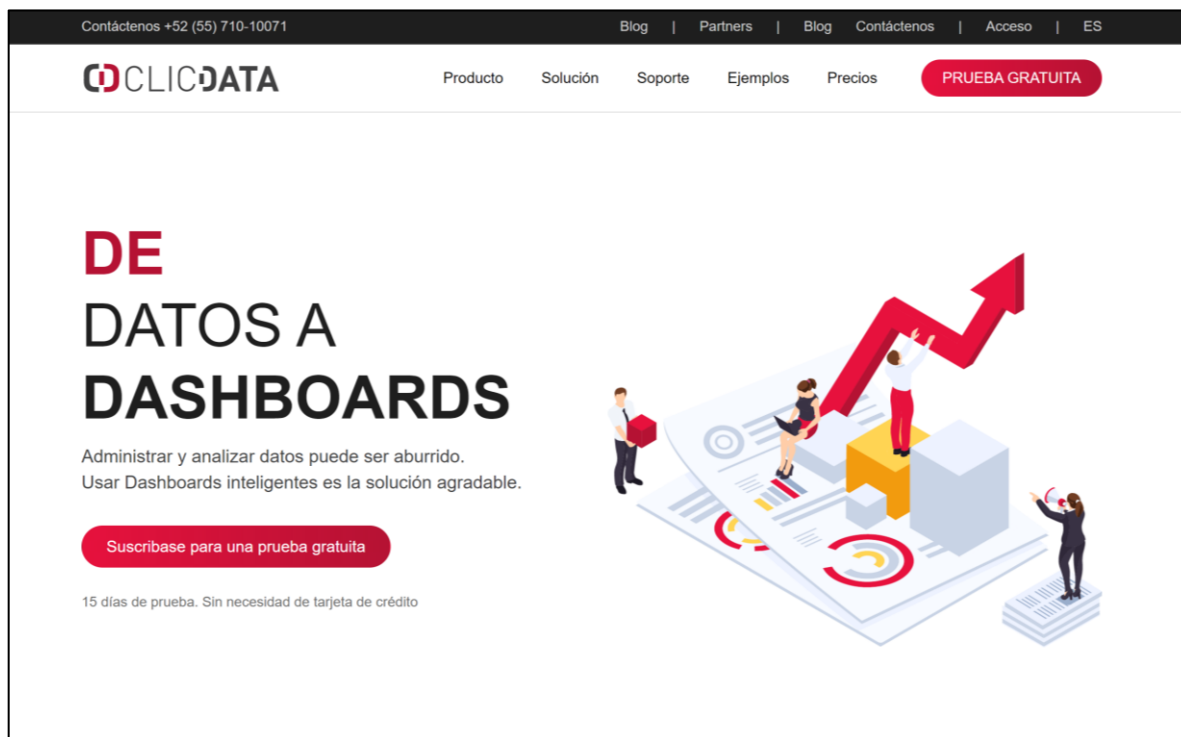


Figura 2-27 Página principal del sitio web de Clicdata

10. Chartio

La herramienta Chartio es una aplicación basada en la nube enfocada en *Data Analytics* [70] (véase Figura 2-28). Esta herramienta queda muy bien explicada en un estudio [71] de diferentes herramientas (en la cual también se analizan otras herramientas en común con el presente trabajo como Klipfolio y Tableau). Este estudio la define como una herramienta sencilla que permite un fácil manejo de los datos obtenidos y la edición avanzada de los KPIs. Cuenta con soporte SQL para la edición de estos KPIs para obtener *dashboards* altamente personalizados.

Esta herramienta es capaz de importar los datos de una gran variedad de fuentes como MySQL o Google Sheets, entre estas fuentes se excluye la opción de importar archivos Microsoft Excel.

La versión más completa, denominada *Professional*, proporciona la capacidad de generación de *dashboards* ilimitados y tiene un precio de 60\$/mes.

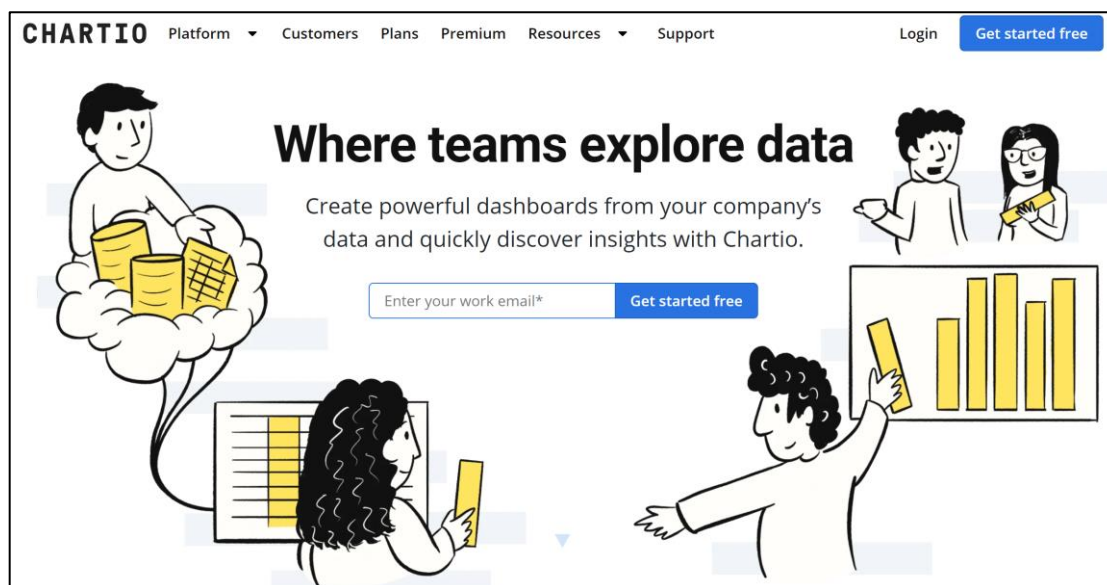


Figura 2-28 Página principal del sitio web de Chartio

2.6.2 Conclusión

Se presenta en la Tabla 2-4 una comparativa final de todas las herramientas expuestas en el apartado anterior con el fin de visualizar gráficamente la relación existente entre cada una de estas y los criterios base que no todas las herramientas cumplen. En esta tabla se muestra en verde el cumplimiento del criterio especificado. Como en la Tabla 2-2, la columna de sencillez se valorará siguiendo el mismo código de colores con respecto al nivel de conocimiento necesario para su empleo (rojo-avanzado, amarillo-medio y verde-básico).

	<i>Open-source</i>	Fácil implementación con Kylin	Usuarios ilimitados	Dashboards ilimitados	Sencillez (subjetivo)
Cyfe					
Klipfolio					
Geckoboard					
Superset					
Tableau					
Grafana					
InetSoft					
Zeppelin					
Clicdata					
Chartio					

Tabla 2-4 Comparativa de herramientas de visualización

Tomando en consideración el estudio de las herramientas realizado en el apartado 2.6.1 y la Tabla 2-1 se opta por escoger Apache Superset como herramienta más adecuada acorde a los intereses y objetivos de este trabajo, es decir, para la generación del panel de control que mostrará la información de interés para el COVAM, así como permitir su análisis de manera sencilla, en detalle y ordenada. Se selecciona esta herramienta por su gran compatibilidad y demostrada eficiencia con Apache Kylin y por la gran variedad de funciones, gráficos, tablas y paneles de los que dispone para la generación de dashboards personalizados.

3 DESARROLLO DEL TFG

En este capítulo, habiendo seleccionado Apache Kylin como motor analítico y Apache Superset como herramienta de creación y visualización de *dashboards*, se describirá, paso a paso, la instalación y configuración de ambas herramientas, así como el proceso seguido para la creación del panel de mando y control. Para ello se ha de explicar también la situación inicial de la que partimos en lo que a los datos AIS se refiere. Por último, se especifica el tipo de información que puede resultar de interés y se presentan los *dashboards* diseñados que harán la función de panel de mando y control.

3.1 Datos AIS

Los datos utilizados en el desarrollo de este proyecto son datos AIS proporcionados por el COVAM. Como ya se ha mencionado en el apartado 1.1, el sistema AIS es un sistema satelital que proporciona información cinemática relevante de las embarcaciones emisoras a otros buques, estaciones costeras y VTS (*Vessel Traffic Server*). Entre la información transmitida por este sistema destacan los datos referentes a la identidad, dimensiones, velocidad, rumbo y posicionamiento de las embarcaciones.

El gran volumen de datos a gestionar por el COVAM es inabarcable sin un sistema software que permita su procesamiento eficiente, análisis en profundidad y puesta en pantalla de manera ordenada, clara y concisa. A diferencia de los trabajos predecesores (explicados brevemente en el apartado 2.4), en los cuales el análisis se limita a la información obtenida de dos días específicos, para este proyecto se dispone de un gran volumen de datos remitidos por el COVAM y almacenados en un servidor con sistema operativo Linux situado en el CUD. Estos datos serán accesibles para el sistema a emplear mediante la creación y población de dos colas Kafka. Estas colas Kafka, gracias a un procesamiento previo que no forma parte del objeto de estudio de este trabajo, contendrán únicamente datos considerados de relevancia e interés para el COVAM. Este procesamiento previo es necesario para evitar duplicidades, realizar agrupaciones convenientes, filtrar los datos y editar los campos que definen la información, entre otras, e inicialmente está pensado para desarrollarse con Apache Storm.

La primera cola Kafka contiene mensajes estáticos con un gran número de campos que permanecen constantes para cada barco pero que, sin embargo, llegan con relativa frecuencia. A esta cola Kafka se le denominó *ship_info*. La segunda cola Kafka contiene mensajes con un mayor número de campos que son emitidos con mayor frecuencia. Esta cola contiene datos definidos por campos variables en el tiempo (como pueden ser la posición, el rumbo o la velocidad) que permiten el seguimiento de la ruta de las embarcaciones y será referida como cola *routes* en adelante.

A continuación, se enumeran y describen los campos, contenidos en ambas colas, que se emplearán como fuente de datos para los cubos en Apache Kylin. Estos datos se encuentran descritos en [72].

3.1.1 Cola ship_info

```
{ "currenttime": "2021-03-07 17:24:24", "eta": "208926", "draught": "57", "mmsi": "232419000", "name": "SEVEN NAVICA", "callsign": "MZHR7", "shipType": "Other", "imo": "9177856", "dest": "HARBOR ISLAND USA", "posType": "1", "dimensions": "108", "timestamp": "1615137327" }
```

Figura 3-1 Registro de datos de la cola Kafka *routes*

En la Figura 3-1 se pueden observar los datos contenidos en esta cola de valores constantes, estos datos son los siguientes:

1. MMSI (*Maritime Mobile Service Identity*)

El MMSI se conforma por nueve dígitos que se envían en forma digital por radiofrecuencia y que identifica de forma única a las distintas estaciones transmisoras, ya sean embarcadas, costeras o en tierra firme.

2. Name

Indica el nombre original de la embarcación transmisora.

3. Callsign

Se trata de un distintivo de llamada que identifica de manera única a cada embarcación. A menudo los nombres de los barcos pueden repetirse o ser muy similares, sin embargo, con este identificador se evitan las posibles confusiones. Este identificador puede estar conformado tanto por letras como por números. Suelen ser asignados formalmente por agencias gubernamentales.

4. IMO number (*International Maritime Organization*)

El IMO es un identificador único para buques y compañías registradas de gestión de buques. Consta de siete dígitos asignados a cada buque precedidos por las letras IMO. A diferencia del *callsign* su objetivo no es el de ser enviado por los diversos métodos de comunicaciones sino el de aumentar la seguridad en la mar mediante la reducción de posibles intentos de fraude.

5. ETA (*Estimated Time of Arrival*)

Tiempo en el que se estima que una embarcación llegue a su destino. Se trata de un campo temporal y mide los segundos aproximados que le queda a una embarcación para llegar a su destino en el momento que se envía el mensaje.

6. Dimension

Indica la eslora en metros de las embarcaciones. El sistema AIS muestra este parámetro desglosado en las distancias desde la antena transmisora hasta la proa del buque y de la antena a la popa. Tras un preprocesado previo, este dato se obtiene como la suma de ambas distancias.

7. Draught

Este campo indica el calado del buque en metros.

8. Destination

El campo *destination* hace referencia al puerto al que se dirige la embarcación, es decir, su próxima recalada prevista.

9. Currenttime

Es un campo temporal que indica el momento en el que la emisión se ha producido. Se emplea el uso horario UTC debido a la posible diferencia horaria entre las estaciones transmisoras y las receptoras.

Apache Kylin permite el desglose de este campo en los siguientes indicadores: YEAR_START (año), QUARTER_START (cuarto de año), MONTH_START (mes), WEEK_START (semana), DAY_START (día), HOUR_START (hora) y MINUTE_START (minuto).

10. Postype

Este campo indica el método de obtención de la posición. Es decir, indica el sistema de posicionamiento que emplea la embarcación. Es un parámetro numérico que va desde el 0 al 15. En la Tabla 3-1 se puede observar qué representa cada valor.

Type Code	Descripción	Type Code	Descripción
0	No definido	6	Sistema de Navegación Integrado
1	GPS	7	Surveyed
2	GLONASS	8	Galileo
3	Combinado GPS/GLONASS	9-14	Fuera de uso
4	Loran-C	15	No definido
5	Chayka		

Tabla 3-1 Descripción del valor *postype* [73]

11. Shiptype

Este parámetro numérico está compuesto de dos cifras y se trata de un número identificativo, que va desde el 10 al 99, cuyo valor especifica el tipo de embarcación de la que se trata (véase la Tabla 3-2). Tras un procesado previo de agrupación, el registro de la cola Kafka mostrará este parámetro como la descripción del tipo de barco y no como su valor numérico, facilitando así la visualización de la información.

Type Code	Descripción	Type Code	Descripción
10 - 19	Reserved for future use	51	Search and Rescue vessel
20 - 29	Wing in ground (WIG), all ships of this type	52	Tug
30	Fishing	53	Port Tender
31 - 32	Towing	54	Anti-pollution equipment
33	Dredging or underwater ops	55	Law Enforcement
34	Diving ops	56 - 57	Spare - Local Vessel
35	Military ops	58	Medical Transport
36	Sailing	59	Noncombatant ship
37	Pleasure Craft	60 - 69	Passenger, all ships of this type
38 - 39	Reserved	70 - 79	Cargo, all ships of this type
40 - 49	High speed craft (HSC), all ships of this type	80 - 89	Tanker, all ships of this type
50	Pilot Vessel	90 - 99	Other Type, all ships of this type

Tabla 3-2 Descripción del valor de *shiptype* [73]

3.1.2 Cola routes

```
{ "eslora": "50", "currenttime": "2021-03-09 05:53:13", "mmsi": "503002930", "destination": "SYDNEY - JBW", "sog": "0", "lon": "151.196163333333", "draught": "30", "zone": "Remainder", "rot": "128", "shipType": "Fishing", "cog": "2756", "navStatus": "0", "lat": "-33.8628716666667", "timestamp": "1615268589" }
```

Figura 3-2 Registro de datos de la cola Kafka *ship_info*

Como se muestra en la Figura 3-2, esta cola permite el seguimiento y monitorización de las embarcaciones. Además de los campos MMSI, *currenttime*, *dimension*, *draught* y *shiptype* (explicados en el apartado anterior); los registros que proporciona esta cola incluyen los siguientes campos:

1. Longitude

Indica la longitud geográfica en la que se encuentra la embarcación. Se expresa en grados, incluyendo decimales, e indica con el signo (-) si es una coordenada sur.

2. Latitude

Indica la latitud geográfica en la que se encuentra la embarcación. Se expresa en grados, incluyendo decimales, e indica con el signo (-) si es una coordenada oeste.

3. Zone

Se trata de un indicador geográfico obtenido del preprocesamiento previo. Dependiendo de la posición, las embarcaciones son agrupadas en las siguientes zonas: Península, Canarias, Mar Rojo, Mar Negro, Golfo de Adén, Mediterráneo, Golfo de Guinea y *Remainder* (el resto de zonas).

Estas zonas fueron seleccionadas a petición del COVAM por tratarse de áreas de interés nacional, ya sea por contener aguas de soberanía nacional o lindantes a estas, o por tratarse de áreas de interés militar o comercial. La Figura 3-3 muestra una captura de pantalla de Google Maps [74] sobre la cual se han marcado dichas zonas.

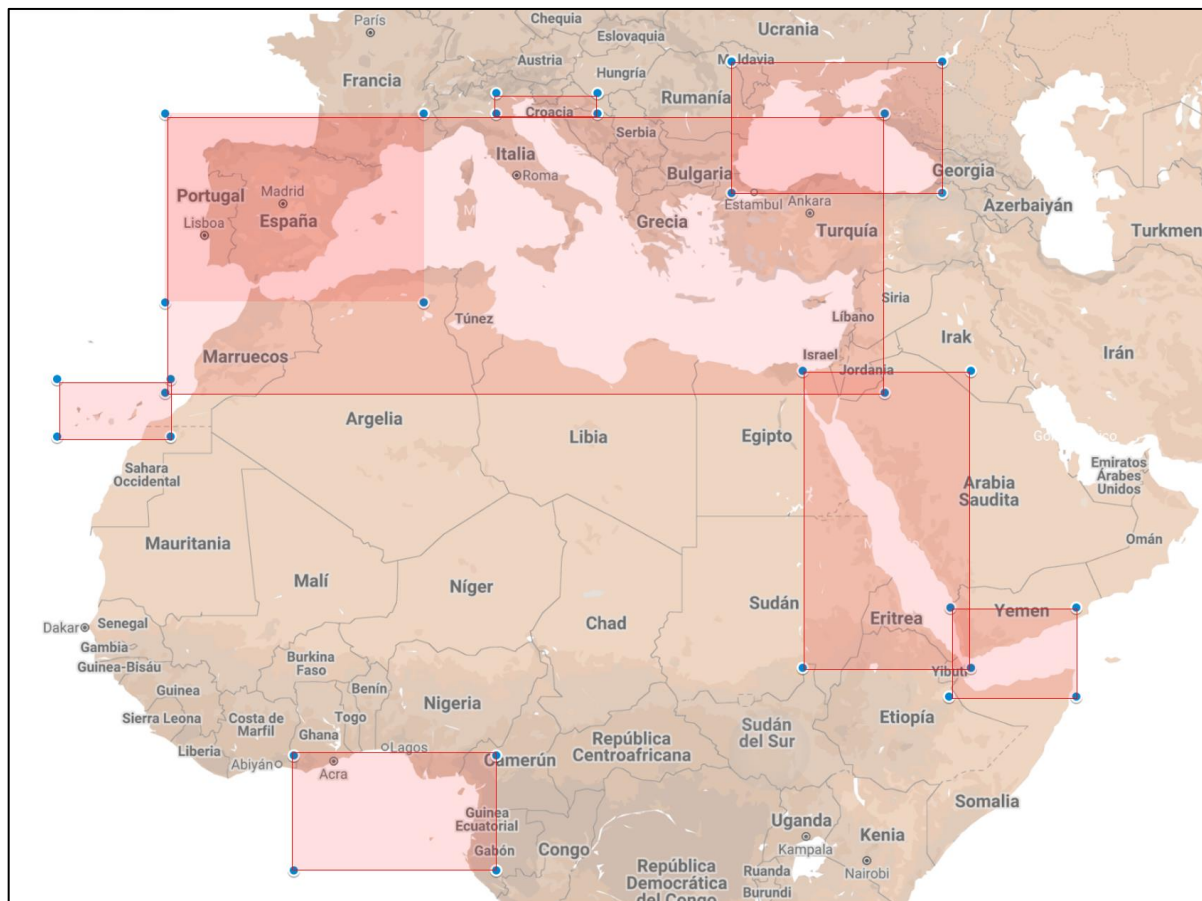


Figura 3-3 Zonas definidas para la cola *routes* [elaboración propia]

4. ROT (*Rate of Turn*)

El campo ROT define la velocidad de giro de una embarcación. Debido a cálculos internos realizados por el sistema AIS, este parámetro se muestra como un número entre 0 y 127, para decodificarlo y obtener la velocidad de giro en grados por minuto se ha de dividirse entre 4,733 y luego elevarlo al cuadrado. Este campo también indica la banda a la que se está cayendo, mediante el empleo del signo (-) si la caída es a babor. En caso de que el valor indicado sea 128 significa que no hay información acerca de este parámetro.

5. COG (*Course over Ground*)

Rumbo con respecto al fondo marino al que está navegando una embarcación. Se mide en decenas de grados (0-3600).

6. SOG (*Speed over Ground*)

Este parámetro numérico indica la velocidad sobre el fondo marino. Este campo tiene una resolución de 0,1 nudos y mide velocidades hasta 102 nudos (indicado por 1020). El valor 1023 indica que este dato no está disponible.

7. NavStatus

Este campo hace referencia al estado de navegación en la que se encuentra la embarcación. Sus valores y significados se muestran en la Tabla 3-3.

<i>Navigational Status</i>	Descripción	<i>Navigational Status</i>	Descripción
0	Navegando a motor	8	Navegando a vela
1	Fondeado	9	Reservado para futuras modificaciones
2	Sin control	10	Reservado para futuras modificaciones
3	Maniobrabilidad restringida	11	Remolcando a motor por la popa
4	Restringido por su calado	12	Remolcando a motor empujando o por el costado
5	Amarrado	13	Reservado para futuras modificaciones
6	Encallado	14	AIS-SART (activo), MOB-AIS, EPIRB-AIS (balizas de transmission AIS)
7	Dedicado a la pesca	15	No definido (por defecto)

Tabla 3-3 Descripción de los valores de *navstatus* [75]

3.2 Arquitectura del sistema

Cuando se desarrolló por primera vez el motor analítico Apache Kylin, el propósito que se tenía en mente era el de resolver las necesidades de análisis de información de las grandes organizaciones en entornos *Big Data*. La arquitectura típica es la mostrada en la Figura 2-15. Este sistema, a grandes rasgos, emplea Hive como *Data Warehouse* y HBase como almacén de los cubos generados.

Y es cierto que las latencias obtenidas con este método tradicional son muy reducidas, permitiendo hablar de consultas en tiempo real, una vez los datos son agregados y almacenados en el sistema. Sin embargo, esta tecnología de procesamiento no permite la realización de consultas al mismo tiempo que se están generando los datos. Esto, según la documentación de Apache Kylin [76] relacionado con la materia, se debe a la existencia de dos procesos fundamentales para el sistema que generan largos tiempos de espera: la extracción de los datos mediante ETL (explicado en el apartado 2.2.2) al *Data Warehouse* y la extracción de los datos del *Data Warehouse* para la generación de los cubos (véase Figura 3-4).

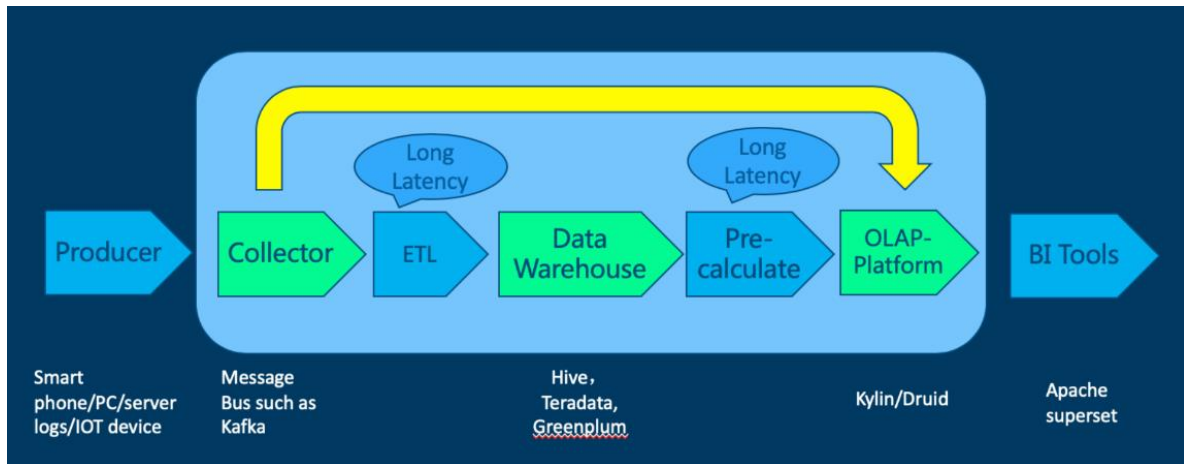


Figura 3-4 Flujo de datos tradicional [76]

Por tanto, es necesario saltarse estos dos procesos de manera que se pueda pasar del colector directamente a la plataforma OLAP. Así se decidió que el método adecuado para el cumplimiento del objetivo del presente trabajo es el denominado *Real-Time OLAP*.

Este proceso, muy reciente y solo presente en las tecnologías *Big Data* más punteras, se puede encontrar detallado paso a paso en la referencia mencionada anteriormente. Se basa en la entrada del flujo de datos (las colas Kafka en este caso) directamente a un nodo denominado *Streaming Receiver*. Este nodo, que consiste básicamente en una serie de procesos Java, tiene como función principal el consumo de los datos en tiempo real y la generación de cubos a partir de estos. Un conjunto de *Streaming Receivers* es denominado *Receiver Cluster*, y el encargado de su coordinación es el *Streaming Coordinator* o *Master Coordinator*. Existe la opción de generar réplicas de estos *Streaming Receivers*, *Replica Sets*, para añadir redundancia y seguridad al sistema.

Una vez generados los cubos, la información contenida en ellos se actualiza periódicamente mediante el establecimiento de un campo de partición (columna temporal de las tablas de datos) y un *Stream Cube Window* (modificable en las configuraciones de Kylin) que indica el tamaño, en tiempo, de los segmentos particionados. Una vez finalizada la duración de un segmento, dejando margen de llegada a mensajes tardíos (*Slide Window*), se dice que un segmento es inmutable y existen dos políticas respecto a su gestión. Por un lado, PURGE se emplea para la eliminación de los segmentos inmutables. Por otro lado, FULL_BUILD vuelca los segmentos en HBase mediante el sistema de archivado de directorios de Hadoop (HDFS) y MapReduce, que comprime y reprocesa los segmentos en disco. MapReduce se emplea como *Build Engine* para la construcción de cubos estáticos, a partir de los segmentos existentes en HDFS, y su agregación en HBase.

El método que se empleará en este trabajo será el segundo, para posibilitar la obtención de históricos. El esquema, por tanto, del flujo de datos y consultas queda definido como se muestra en la Figura 3-5.

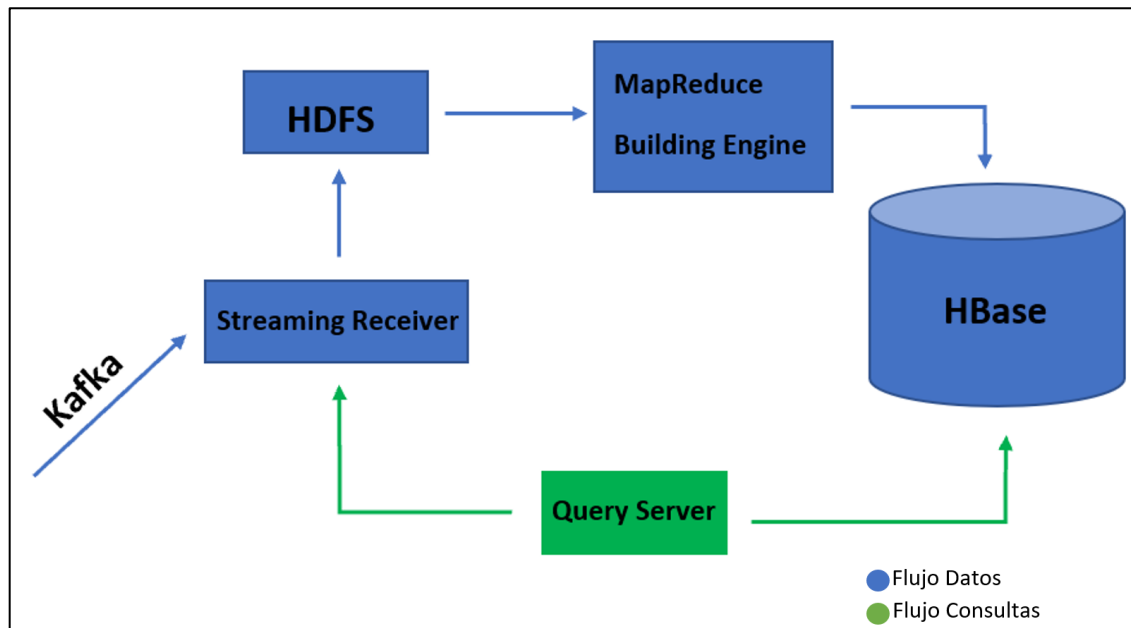


Figura 3-5 Flujo de datos del sistema *Real-Time OLAP* [elaboración propia]

En la Figura 3-6 se muestra la arquitectura global de este TFG. Cabe destacar que, a pesar de que en el presente trabajo no se vaya a emplear, Hive es necesario para la instalación de Kylin y la posibilidad de almacenamiento tradicional en esta herramienta persiste.

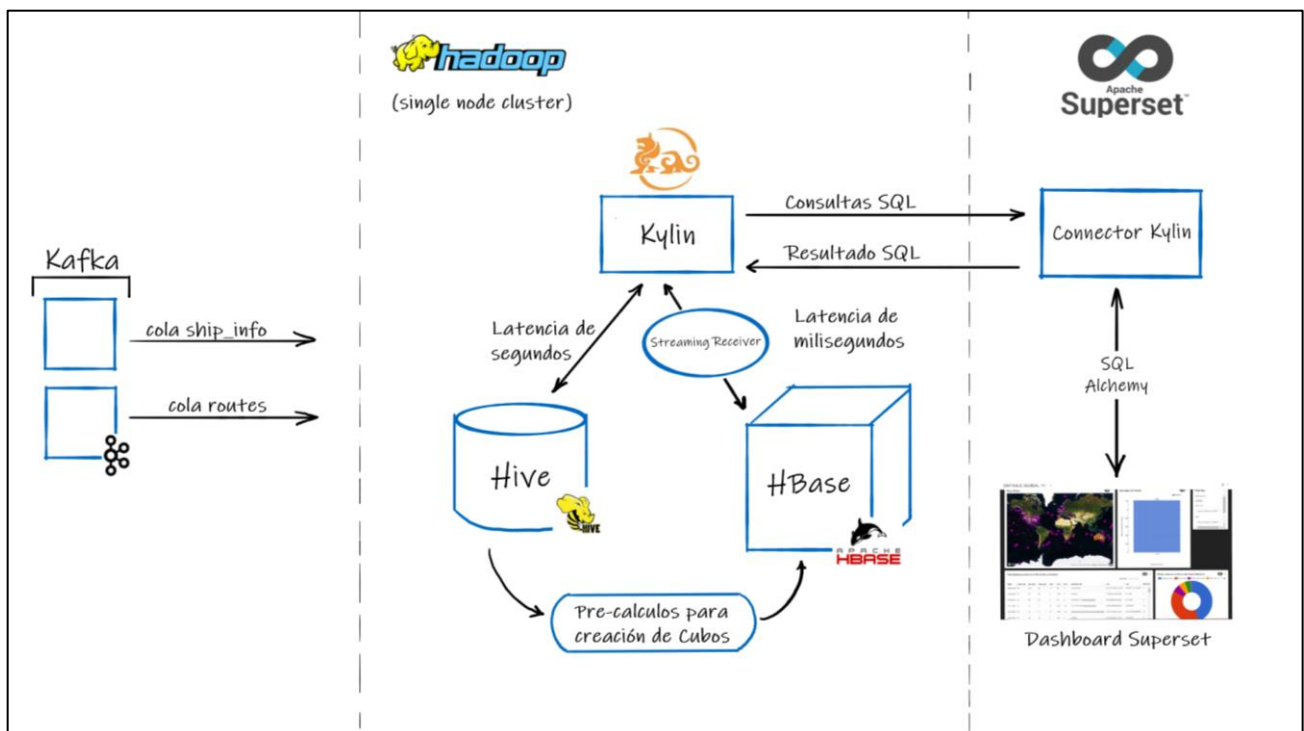


Figura 3-6 Arquitectura del sistema Apache Kylin-Apache Superset [elaboración propia]

3.3 Instalación y configuración

Para la obtención de este sistema es necesaria una instalación y configuración previa de cada una de las herramientas mostradas en el esquema anterior. Para ello se estableció un acceso de forma remota a un servidor con sistema operativo Linux (Ubuntu server 20.04) perteneciente al CUD. Este acceso se

consiguió mediante el empleo del protocolo SSH desde una máquina con sistema operativo Windows (desde el Intérprete de comandos). Se accede al servidor con los siguientes datos:

- IP: 193.146.212.193
- Usuario: cuquerella
- Contraseña: la que corresponda

3.3.1 Requisitos previos

La herramienta Apache Kylin requiere de una serie de requisitos previos de los que ha disponer el servidor anfitrión. Estos requisitos son listados en la guía de instalación de la página web de Apache Kylin [77] indicando también las versiones aceptadas de cada uno:

- Hadoop: versión 3.1 o superior.
- Hive: versión 1.2.1 o superior.
- HBase: versión 2.0.
- Spark (opcional): versión 2.3.0 o superior.
- Kafka (opcional): versión 1.0.0 o superior.
- Java Development Kit (JDK): versión 1.8 o superior.
- Sistema Operativo Linux: CentOS versión 6.5 o superior o Ubuntu versión 16.0.4 o superior.

La configuración mínima de un servidor que ejecuta Kylin es:

- CPU de 4 núcleos.
- 16 GB de RAM.
- 100 GB de disco duro.

En cuanto a la instalación de Apache Hadoop, existen dos métodos de configuración: multi node cluster y single node cluster. En el caso de este TFG trabajaremos con un único servidor que realizará la función de todos los nodos (*NameNode*, *DataNode*, *ResourceManager* y *NodeManager*), por tanto, se instalará como *single node cluster*. Destacar que en caso de disponer de más servidores dedicados al análisis y gestión de los datos se podría instalar cómo *multi node cluster*.

Lo primero que se tuvo que instalar fue el *Java Development Kit* (JDK) ya que el *framework* de Hadoop está escrito en Java. El JDK que se instaló fue *openjdk-8* con el comando `$ sudo apt install openjdk-8-jdk -y`.

Lo siguiente fue crear un usuario hadoop (Figura 3-7) para la ejecución de todos los servicios asociados a Apache Kylin. Luego se inició sesión en el usuario recién creado y se generó un par de claves pública/privada para la identificación automática en SSH mediante claves RSA. La clave pública se incluyó en el directorio de contraseñas autorizadas.

Tras esto, se procedió a la instalación de las herramientas software necesarias para la instalación de Kylin. En el caso del presente trabajo estas herramientas son Hadoop, Hive y Hbase. Para la descarga de estas herramientas se empleó el comando *wget* y las páginas web de las que se obtuvieron son las referenciadas en [78], [79] y [80] respectivamente. Las versiones que se instalaron, en la ruta */usr/local*, son:

- Hadoop-3.2.2
- Hive-3.1.2
- HBase-2.4.0

```
cuquerella@tfg:~$ sudo adduser hadoop
Adding user `hadoop' ...
Adding new group `hadoop' (1002) ...
Adding new user `hadoop' (1002) with group `hadoop' ...
Creating home directory `/home/hadoop' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hadoop
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
cuquerella@tfg:~$ sudo su hadoop
hadoop@tfg:/home/cuquerella$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Created directory '/home/hadoop/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:LunyJGx7R3VI2/DKT72cjPYyT0q26sw7yj/wOPTGoA hadoop@tfg
The key's randomart image is:
+---[RSA 3072]-----+
|
|      o
|     . *
|    + +
|   So o..
|  .. ooo o+. .
| .+o+O...oo+ o
| E..*=+O.. +o=
| .+=BB*oo...
|
+---[SHA256]-----+
hadoop@tfg:/home/cuquerella$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Figura 3-7 Creación de usuario hadoop

Es importante ir indicando los *path* y variables de las distintas aplicaciones y herramientas, a medida que se van instalando, para que el sistema pueda consultarlos automáticamente. Con esta finalidad se creó un fichero denominado *hadoop_java.sh* (en el directorio */etc/profile.d*) en el cual se fueron añadiendo dichas variables. La Figura 3-8 muestra el contenido de este fichero al finalizar la configuración de todo el sistema.

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
#default-java
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=/usr/local/hadoop/lib/native"
export HBASE_HOME=/usr/local/HBase
export HIVE_HOME=/usr/local/hive
export KYLIN_HOME=/home/hadoop/apache-kylin-3.0.0-bin-hadoop3
export KAFKA_HOME=/home/hadoop/kafka_2.12-2.5.0
export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$HBASE_HOME/bin:$HIVE_HOME/bin:$KYLIN_HOME/bin:$KAFKA_HOME/bin
```

Figura 3-8 Fichero *hadoop_java.sh*

1. Configuración de Apache Hadoop

Para una correcta instalación de Hadoop fue necesaria la configuración de los cinco ficheros mostrados en la Figura 3-9.

```

hadoop@tfg:/usr/local/hadoop/etc/hadoop$ ls
capacity-scheduler.xml      kms-log4j.properties
configuration.xml           kms-site.xml
container-executor.cfg      log4j.properties
core-site.xml               mapred-env.cmd
hadoop-env.cmd              mapred-env.sh
hadoop-env.sh               mapred-queues.xml.template
hadoop-metrics2.properties mapred-site.xml
hadoop-policy.xml           shellprofile.d
hadoop-user-functions.sh.example  ssl-client.xml.example
hdfs-site.xml               ssl-server.xml.example
httpfs-env.sh               user_ec_policies.xml.template
httpfs-log4j.properties     workers
httpfs-signature.secret     yarn-env.cmd
httpfs-site.xml             yarn-env.sh
kms-acls.xml                yarn-site.xml
kms-env.sh                  yarnservice-log4j.properties

```

Figura 3-9 Ficheros de configuración de Hadoop

Para la configuración de *hadoop-env.sh*, dedicado a la configuración del entorno Hadoop, fue necesario introducir el *path* del JDK como variable en el fichero (Figura 3-10).

```

# Many of the options here are built from the perspective that users
# may want to provide OVERWRITING values on the command line.
# For example:
#
# JAVA_HOME=/usr/java/testing hdfs dfs -ls
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append. If append
# is preferable, modify this file accordingly.

```

Figura 3-10 Fichero *hadoop-env.sh* con el *path* de Java añadido

Se tuvo que configurar el archivo *core-site.xml*, que define los componentes *core* (núcleo) HDFS y *MapReduce*. En este archivo de configuración se debe indicar dónde se procesa *NameNode* (encargado de gestionar el *MetaData* que indica en qué *DataNode* se almacenan los diversos datos y archivos). Para ello fue necesario crear un directorio temporal (*hadooptmpdata*) y añadir su nombre y *path* en las propiedades del documento de *core-site.xml* (Figura 3-11). También se debe indicar a HDFS que se ejecute en el *localhost* con puerto 9000.

```

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
    <description>The default file system URI</description>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/hadoop/hadooptmpdata</value>
  </property>
</configuration>

```

Figura 3-11 Fichero *core-site.xml* con las propiedades añadidas

El archivo *hdfs-site.xml*, contiene los datos de configuración de *NameNode*, *Secondary NameNode* (que lee el *MetaData* de *NameNode* y lo añade al disco duro) y *DataNode*. Se añadieron en las propiedades de configuración los siguientes datos: el factor de replicación (1 al ser *single node cluster*) y los *path* a los directorios de *NameNode* y *DataNode* (Figura 3-12). Para ello hizo falta crear ambos directorios con antelación.

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>

  <property>
    <name>dfs.name.dir</name>
    <value>file:///hadoop/hdfs/name</value>
  </property>

  <property>
    <name>dfs.data.dir</name>
    <value>file:///hadoop/hdfs/data</value>
  </property>
</configuration>
```

Figura 3-12 Fichero *hdfs-site.xml* con las propiedades añadidas

El siguiente archivo que se ha de configurar es *mapred-site.xml* que se encarga de la configuración del *MapReduce*. Se añadieron las propiedades que se muestran en la Figura 3-13.

El último archivo que se configuró fue *yarn-site.xml*. A este archivo le debemos añadir las propiedades de configuración mostradas en la Figura 3-14. Este fichero se encarga de la gestión de los recursos (*ResourceManager*), para permitir el procesamiento de los datos por parte de los distintos motores analíticos (Apache Kylin en nuestro caso) y los distintos métodos empleados (como *streaming* o *batch*).

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>
```

Figura 3-13 Fichero *mapred-site.xml* con las propiedades añadidas

```
<configuration>

<!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

Figura 3-14 Fichero *yarn-site.xml* con las propiedades añadidas

Una vez completada la configuración, es importante formatear el *NameNode* cuando se emplea Hadoop por primera vez. Esto se realiza ejecutando el comando: *hdfs namenode -format*. Una vez hecho esto, se puede iniciar Hadoop con los comandos *start-dfs.sh* y *start-yarn.sh*, esto iniciará el funcionamiento de *DataNode*, *NameNode*, *Secondary NameNode*, *ResourceManager* y *NodeManager*. Se puede confirmar su ejecución mediante el comando *jps* (véase Figura 3-15).

```
hadoop@tfg:/usr/local$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [tf]
hadoop@tfg:/usr/local$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@tfg:/usr/local$ jps
3328 NodeManager
2580 NameNode
3670 Jps
3164 ResourceManager
2956 SecondaryNameNode
2733 DataNode
hadoop@tfg:/usr/local$
```

Figura 3-15 Inicio de Apache Hadoop y confirmación con el comando *jps*

2. Configuración de Apache Hive

El siguiente requisito es la instalación de Apache Hive en el *single node cluster* de Hadoop.

Lo primero fue actualizar el *path* de Hadoop en el archivo de configuración de Hive, como se muestra en la Figura 3-16. Este archivo de configuración (*hive-confiq.sh*) se encuentra dentro de la carpeta *bin* de la carpeta *hive*.

```
# Default to use 256MB
export HADOOP_HEAPSIZE=${HADOOP_HEAPSIZE:-256}
export HADOOP_HOME=/usr/local/hadoop
```

Figura 3-16 Path de Hadoop en *hive-config.sh*

Para comprobar que Hive está correctamente instalado y configurado se introduce el comando *hive* y el resultado es la inicialización de Hive shell (véase Figura 3-17).

```
hadoop@tfg:/usr/local/hive/bin$ hive
/usr/local/hadoop/libexec/hadoop-functions.sh: line 2366: HADOOP_ORG.APACHE.HADOOP.HBASE.UTIL.GETJAVAPROPERTY_USER: invalid variable name
/usr/local/hadoop/libexec/hadoop-functions.sh: line 2461: HADOOP_ORG.APACHE.HADOOP.HBASE.UTIL.GETJAVAPROPERTY_OPTS: invalid variable name
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = c7590e53-f8ce-4468-bf9b-a8b80144f20f
Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
```

Figura 3-17 Hive shell

3. Configuración de Apache HBase

Para la correcta instalación de Apache HBase fue necesaria la configuración del fichero *hbase-site.xml*, añadiendo las líneas de código mostradas en la Figura 3-18. Se recomienda instalar HBase en modo pseudodistribuido, lo cual significa que, aunque el programa se ejecute en un único *host*, cada *daemon* de HBase (*HMaster*, *HRegionServer* y *ZooKeeper*) ejecutará un proceso separado e

independiente. Para eso el directorio raíz establecido es el *localhost* de Hadoop en el puerto 9000, y el clúster de HBase se establece como distribuido.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://localhost:9000/hbase</value>
  </property>

  <property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>/home/hadoop/zookeeper</value>
  </property>

  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property></configuration>
```

Figura 3-18 Configuración del fichero *hbase-site.xml*

En el fichero *hbase-env.sh* se indica el JDK que se emplea y su *path* correspondiente (ver Figura 3-19).

```
# The java implementation to use.  Java 1.8+ required.
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Figura 3-19 Configuración del fichero *hbase-env.sh*

Para iniciar HBase, se emplea el comando *start-hbase.sh*. Una vez lanzado el comando, podemos entrar en la HBase *shell* (ver Figura 3-20).

```
hadoop@tfg:/usr/local/HBase$ hbase shell
/usr/local/hadoop/libexec/hadoop-functions.sh: line 2366: HADOOP_ORG.APACHE.HADOOP.HBASE.UTIL.GETJAVAPROPERTY_
USER: invalid variable name
/usr/local/hadoop/libexec/hadoop-functions.sh: line 2461: HADOOP_ORG.APACHE.HADOOP.HBASE.UTIL.GETJAVAPROPERTY_
OPTS: invalid variable name
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4
j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/HBase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/
slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.0, r282ab70012ae843af54a6779543ff20acbcbb629, Thu Dec 3 09:58:52 PST 2020
Took 0.0052 seconds
hbase:001:0> exit
```

Figura 3-20 HBase *shell*

4. Instalación de Apache Kafka

Apache Kafka será el método de entrada de los datos en el sistema. Para ello fue necesaria su instalación en el servidor, empleando el comando *wget*. La herramienta se descargó, al igual que el resto, del directorio de descargas de *Apache Software Foundation* [81]. Para inicializar Apache Kafka, se empleó el comando *./kafka-server-start.sh config/server.properties*.

3.3.2 Instalación y configuración de Apache Kylin

Apache Kylin se descargó utilizando el comando `wget` del directorio de descargas de Apache [82], para posteriormente descomprimirlo e instalarlo en la ruta `/home/hadoop`.

Después, fue necesario configurar la variable `$KYLIN_HOME` apuntando al directorio recién descomprimido (Figura 3-21). Esta variable también se ha de añadir al ya mencionado fichero `hadoop_java.sh`.

```
hadoop@tfg:~$ cd apache-kylin-3.0.0-bin-hadoop3/
hadoop@tfg:~/apache-kylin-3.0.0-bin-hadoop3$ export KYLIN_HOME=`pwd`
hadoop@tfg:~/apache-kylin-3.0.0-bin-hadoop3$ echo $KYLIN_HOME
/home/hadoop/apache-kylin-3.0.0-bin-hadoop3
```

Figura 3-21 variable `$KYLIN_HOME`

Kylin se procesa en un cluster Hadoop del que exige unos requisitos como las versiones de las herramientas correctas o los permisos de acceso. Para evitar problemas a la hora de instalar la herramienta, se puede comprobar que el entorno en el que se pretende instalarlo es adecuado. Esto se consigue con el siguiente comando: `$KYLIN_HOME/bin/check-env.sh`. Si la respuesta obtenida no muestra errores, se puede iniciar el proceso de Apache Kylin.

Para el inicio de Apache Kylin se lanza el comando `$KYLIN_HOME/bin/kylin.sh start`. Como se puede ver en la Figura 3-22, Kylin exige la descarga de Spark la primera vez que se lanza, aunque inicialmente en la documentación no se exigía. Esta descarga se muestra en la Figura 3-23. Tras su finalización ya se pudo iniciar Kylin sin errores (véase figura Figura 3-24).

La página de instalación de Kylin recomienda la clonación del directorio general de Kylin para poder lanzar, por un lado, Kylin en el modo de funcionamiento normal y, por otro, el modo *streaming* (mediante el comando `kylin.sh streaming start`).

```
hadoop@tfg:~$ $KYLIN_HOME/bin/kylin.sh start
Retrieving hadoop conf dir...
/usr/local/hadoop/libexec/hadoop-functions.sh: line 2366: HADOOP_ORG.APACHE.HADOOP.HBASE
.UTIL.GETJAVAPROPERTY_USER: invalid variable name
/usr/local/hadoop/libexec/hadoop-functions.sh: line 2461: HADOOP_ORG.APACHE.HADOOP.HBASE
.UTIL.GETJAVAPROPERTY_OPTS: invalid variable name
KYLIN_HOME is set to /home/hadoop/apache-kylin-3.0.0-bin-hadoop3
Retrieving hive dependency...
Retrieving hbase dependency...
/usr/local/hadoop/libexec/hadoop-functions.sh: line 2366: HADOOP_ORG.APACHE.HADOOP.HBASE
.UTIL.GETJAVAPROPERTY_USER: invalid variable name
/usr/local/hadoop/libexec/hadoop-functions.sh: line 2461: HADOOP_ORG.APACHE.HADOOP.HBASE
.UTIL.GETJAVAPROPERTY_OPTS: invalid variable name
Retrieving hadoop conf dir...
/usr/local/hadoop/libexec/hadoop-functions.sh: line 2366: HADOOP_ORG.APACHE.HADOOP.HBASE
.UTIL.GETJAVAPROPERTY_USER: invalid variable name
/usr/local/hadoop/libexec/hadoop-functions.sh: line 2461: HADOOP_ORG.APACHE.HADOOP.HBASE
.UTIL.GETJAVAPROPERTY_OPTS: invalid variable name
Retrieving kafka dependency...
Retrieving Spark dependency...
spark not found, set SPARK HOME, or run bin/download-spark.sh
```

Figura 3-22 Error en inicio de Kylin

```
hadoop@tfg:~$ $KYLIN_HOME/bin/download-spark.sh
Downloading spark package...
No binary file found, start to download package to /tmp/spark_package
--2021-02-16 11:25:30-- http://archive.apache.org/dist/spark/spark-2.3.2/spark-2.3.2-bin-hadoop2.7.tgz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 225875602 (215M) [application/x-gzip]
Saving to: 'spark-2.3.2-bin-hadoop2.7.tgz'

spark-2.3.2-bin-hadoo 100%[=====] 215.41M 1.27MB/s in 6m 3s
```

Figura 3-23 Descarga de Apache Spark

```
A new Kylin instance is started by hadoop. To stop it, run 'kylin.sh stop'  
Check the log at /home/hadoop/apache-kylin-3.0.0-bin-hadoop3/logs/kylin.log  
Web UI is at http://tfg:7070/kylin
```

Figura 3-24 Iniciación correcta de Kylin

El terminal indica que Kylin ya esta funcionando y señala el enlace para acceder a la interfaz gráfica (en el puerto 7070). El usuario y contraseña por defecto para abrir dicha interfaz es ADMIN/KYLIN. La interfaz se puede observar en la Figura 3-25, donde se muestra la página principal de esta antes de la creación de cualquier tabla, modelo o cubo.

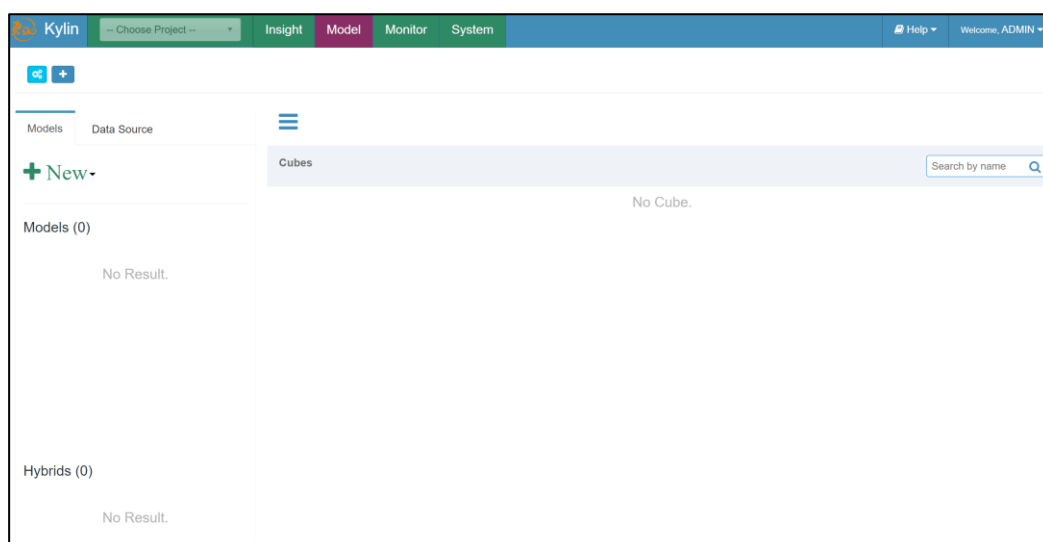


Figura 3-25 Página principal de la interfaz gráfica de Kylin

3.3.3 Instalación y configuración de Superset

El siguiente paso a dar para la creación del sistema software con el que se generará el cuadro de mando y control para el COVAM es la instalación de la herramienta de visualización Apache Superset. Esta instalación se ha llevado a cabo siguiendo los pasos de la guía de instalación de la página web de Superset [83].

Primero fue necesario asegurar que el servidor tuviese instaladas las dependencias de Python requeridas. Estas dependencias se constituyen fundamentalmente por librerías criptográficas de Python para la encriptación de las contraseñas de las conexiones a las distintas bases de datos. Para ello se utilizó el comando mostrado en la Figura 3-26.

```
hadoop@tfg:~$ sudo apt-get install build-essential libssl-dev libffi-dev python3-dev python3-pip libsasl2-dev libldap2-dev  
[sudo] password for hadoop:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
libffi-dev is already the newest version (3.3-4).  
libsasl2-dev is already the newest version (2.1.27+dfsg-2).  
python3-dev is already the newest version (3.8.2-0ubuntu2).  
build-essential is already the newest version (12.8ubuntu1.1).  
libldap2-dev is already the newest version (2.4.49+dfsg-2ubuntu1.6).  
libssl-dev is already the newest version (1.1.1f-1ubuntu2.2).  
python3-pip is already the newest version (20.0.2-5ubuntu1.1).  
0 upgraded, 0 newly installed, 0 to remove and 29 not upgraded.
```

Figura 3-26 Dependencias con Python

También se considera buena práctica asegurar que se dispone de las últimas versiones de las herramientas *pip* y *setuptools* para la instalación. Esto se consigue mediante el comando *pip3 install --upgrade setuptools pip*.

En la guía de instalación se recomienda instalar Superset en un *virtual environment*. Para crearlo y activarlo se usaron los comandos mostrados en la Figura 3-27.

```
hadoop@tfg:~$ sudo apt-get install python3-venv
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-venv is already the newest version (3.8.2-0ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 29 not upgraded.
hadoop@tfg:~$ . venv/bin/activate
(venv) hadoop@tfg:~$
```

Figura 3-27 Creación y activación de un entorno virtual

Una vez creado y activado este entorno virtual, se procedió a la instalación de Superset (Figura 3-28) y la inicialización de su base de datos (Figura 3-29). Tras esto, fue necesario crear un usuario administrador (Figura 3-30). Lo último que se tuvo que hacer antes de poder lanzar el comando para iniciar Superset fue el establecimiento de los roles y permisos (manteniendo los establecidos por defecto), como se muestra en la Figura 3-31.

```
(venv) hadoop@tfg:~$ sudo pip3 install apache-superset
Collecting apache-superset
  Downloading apache-superset-1.0.1.tar.gz (23.9 MB)
    | 23.9 MB 200 kB/s
Requirement already satisfied: PyJWT<2,>=1.7.1 in /usr/lib/python3/dist-packages (from apache-superset) (1.7.1)
Collecting backoff<1.8.0
  Downloading backoff-1.10.0-py2.py3-none-any.whl (31 kB)
Collecting bleach<4.0.0,>=3.0.2
  Downloading bleach-3.3.0-py2.py3-none-any.whl (283 kB)
    | 283 kB 3.7 MB/s
Collecting cachelib<0.2,>=0.1.1
  Downloading cachelib-0.1.1-py3-none-any.whl (13 kB)
Collecting celery!=4.4.1,<5.0.0,>=4.3.0
  Downloading celery-4.4.7-py2.py3-none-any.whl (427 kB)
    | 427 kB 2.1 MB/s
Requirement already satisfied: click<8 in /usr/lib/python3/dist-packages (from apache-superset) (7.0)
Requirement already satisfied: colorama in /usr/lib/python3/dist-packages (from apache-superset) (0.4.3)
Collecting contextlib2
```

Figura 3-28 Instalación de Apache Superset

```
(venv) hadoop@tfg:~$ superset db upgrade
logging was configured successfully
INFO:superset.utils.logging_configurator:logging was configured successfully
```

Figura 3-29 Inicialización de la base de datos

```
(venv) hadoop@tfg:~$ export FLASK_APP=superset
(venv) hadoop@tfg:~$ superset fab create-admin
logging was configured successfully
INFO:superset.utils.logging_configurator:logging was configured successfully
/usr/local/lib/python3.8/dist-packages/flask_caching/__init__.py:191: UserWarning: Flask
-Caching: CACHE_TYPE is set to null, caching is effectively disabled.
  warnings.warn(
No PIL installation found
INFO:superset.utils.screenshots:No PIL installation found
Username [admin]: Juan
User first name [admin]: Juan
User last name [user]: Cuquerella
Email [admin@fab.org]:
Password:
Repeat for confirmation:
Recognized Database Authentications.
Admin User Juan created.
```

Figura 3-30 Creación de usuario administrador

```
(venv) hadoop@tfg:~$ superset init
logging was configured successfully
INFO:superset.utils.logging_configurator:logging was configured successfully
/usr/local/lib/python3.8/dist-packages/flask_caching/__init__.py:191: UserWarning: Flask
-Caching: CACHE_TYPE is set to null, caching is effectively disabled.
  warnings.warn(
No PIL installation found
INFO:superset.utils.screenshots:No PIL installation found
Syncing role definition
INFO:superset.security.manager:Syncing role definition
Syncing Admin perms
INFO:superset.security.manager:Syncing Admin perms
Syncing Alpha perms
INFO:superset.security.manager:Syncing Alpha perms
Syncing Gamma perms
INFO:superset.security.manager:Syncing Gamma perms
Syncing granter perms
INFO:superset.security.manager:Syncing granter perms
Syncing sql_lab perms
INFO:superset.security.manager:Syncing sql_lab perms
Fetching a set of all perms to lookup which ones are missing
INFO:superset.security.manager:Fetching a set of all perms to lookup which ones are miss
ing
```

Figura 3-31 Configuración por defecto de los Roles y permisos

El último paso consiste en lanzar el comando que inicia Superset (véase Figura 3-32). Este comando permite definir el puerto en el que se iniciará Apache Superset, modo de ejecución o host a los que permitirá conectarse, entre otros parámetros. Para acceder a la interfaz gráfica de esta herramienta se empleará, en el caso del presente proyecto, el puerto 8070. En la Figura 3-33 se observa la página de acceso a la interfaz.

En el presente TFG se optó por instalar la herramienta en modo *development server*, por ser lo recomendado en la página de instalación de Superset, sin embargo, es importante destacar que también existe la opción de instalarlo en modo *production server* lo cual permitiría dejar lanzado el programa sin necesidad de iniciarlo cada vez que se quiera acceder a este.

```
(venv) hadoop@tfg:~$ superset run -p 8070 --with-threads --reload --debugger --host 0.0.0.0
logging was configured successfully
INFO:superset.utils.logging_configurator:logging was configured successfully
/usr/local/lib/python3.8/dist-packages/flask_caching/__init__.py:191: UserWarning: Flask-Caching: CACHE_TYPE is set to null, caching is effectively disabled.
  warnings.warn(
No PIL installation found
INFO:superset.utils.screenshots:No PIL installation found
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
INFO:werkzeug: * Running on http://0.0.0.0:8070/ (Press CTRL+C to quit)
INFO:werkzeug: * Restarting with stat
logging was configured successfully
INFO:superset.utils.logging_configurator:logging was configured successfully
/usr/local/lib/python3.8/dist-packages/flask_caching/__init__.py:191: UserWarning: Flask-Caching: CACHE_TYPE is set to null, caching is effectively disabled.
  warnings.warn(
No PIL installation found
INFO:superset.utils.screenshots:No PIL installation found
WARNING:werkzeug: * Debugger is active!
INFO:werkzeug: * Debugger PIN: 176-647-210
```

Figura 3-32 Inicio de Apache Superset

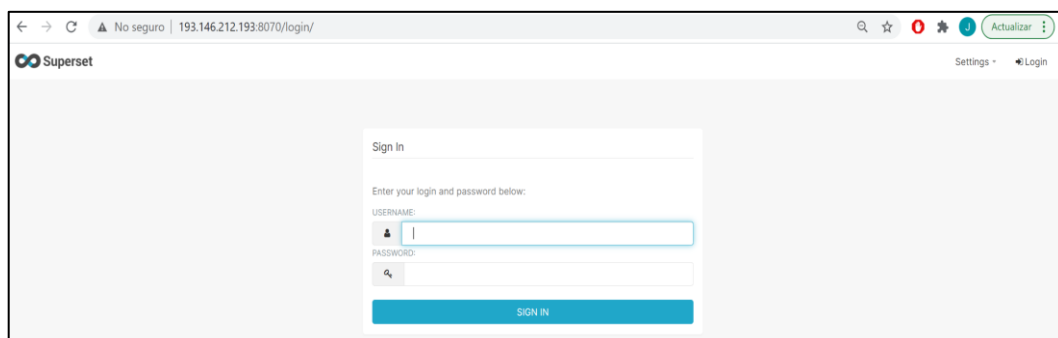


Figura 3-33 Página de acceso a la interfaz web de Apache Superset

3.4 Creación del panel de mando y control

3.4.1 Población de las colas Kafka

Como ya se ha mencionado anteriormente, la entrada de datos al sistema esta constituida por dos colas Kafka, las cuales hay que crear y poblar de información.

La creación de estas colas Kafka se realiza mediante los comandos mostrados en la Figura 3-34 (para la cola *ship_type*, para *routes* es equivalente), donde se indica el *bootstrap-server*, que señala la dirección de acceso al *broker* encargado de recibir los mensajes del productor y permitir su acceso al consumidor. El acceso a los mensajes se realiza mediante el empleo de *topics*. La creación de estas colas no es objeto de estudio del presente trabajo.

```
>> ./kafka-topics.sh --create --topic ship_info --bootstrap-server
localhost:9092
>>
>> ./kafka-configs.sh --bootstrap-server localhost:9092 --entity-type topics
--entity-name ship_info --alter --add-config retention.ms=10000
```

Figura 3-34 Comandos empleados para la creación de la cola *ship_type*.

Para lanzar el productor que puebla las colas se ha de emplear el comando mostrado en la Figura 3-35. Una vez ejecutado, podemos comprobar que, efectivamente, podemos consumir datos de esa cola Kafka mediante el comando mostrado en la Figura 3-36.

```
hadoop@tfg:~$ cd Productor/
hadoop@tfg:~/Productor$ ./exec
[INFO] Scanning for projects...
[INFO]
[INFO] -----< es.uvigo.cud.cemai:proyectokafka >-----
[INFO] Building proyectokafka 1.0
[INFO] -----[ jar ]-----
[WARNING] The POM for dk.dma.enav:enav-model:jar:0.6-SNAPSHOT is missing, no dependency
information available
[WARNING] The POM for dk.dma.commons:dma-commons-util:jar:0.5-SNAPSHOT is missing, no de
pendency information available
[WARNING] The POM for dk.dma.enav:enav-util:jar:0.6-SNAPSHOT is missing, no dependency i
nformation available
[WARNING] The POM for dk.dma.commons:dma-commons-app:jar:0.5-SNAPSHOT is missing, no dep
endency information available
[INFO]
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ proyectokafka ---
SLF4J: Class path contains multiple SLF4J bindings
```

Figura 3-35 Lanzamiento del programa productor de registros para las colas Kafka

```
{ "currenttime": "2021-03-07 17:24:24", "eta": "591952", "draught": "0", "mmsi": "368006610", "name": "JAMES K ELLIS", "callsign": "WDJ7428", "shipType": "Other", "imo": "0", "dest": "UDC>UT2",
"postype": "1", "dimensions": "46", "timestamp": "1615137327" }
{ "currenttime": "2021-03-07 17:24:24", "eta": "76160", "draught": "0", "mmsi": "367622850", "name": "FMS ENDURANCE", "callsign": "WDH4937", "shipType": "Wing in ground", "imo": "9209075", "dest": "EI 338", "postype": "15", "dimensions": "0", "timestamp": "1615137327" }
{ "currenttime": "2021-03-07 17:24:24", "eta": "213367", "draught": "28", "mmsi": "367728240", "name": "HONDO", "callsign": "WDI7389", "shipType": "Spare - Local Vessel", "imo": "0", "dest": "BATON ROUGE LA", "postype": "1", "dimensions": "305", "timestamp": "1615137327" }
{ "currenttime": "2021-03-07 17:24:24", "eta": "208926", "draught": "57", "mmsi": "232419000", "name": "SEVEN NAVICA", "callsign": "MZHR7", "shipType": "Other", "imo": "9177856", "dest": "HARBOR ISLAND USA", "postype": "1", "dimensions": "108", "timestamp": "1615137327" }
{ "currenttime": "2021-03-07 17:24:24", "eta": "170112", "draught": "26", "mmsi": "366919160", "name": "BOBBY JONES", "callsign": "WX5233", "shipType": "Other", "imo": "100502672", "dest": "USFPO>USGSM>USFPO", "postype": "1", "dimensions": "38", "timestamp": "1615137327" }
{ "currenttime": "2021-03-07 17:24:24", "eta": "211264", "draught": "27", "mmsi": "368070420", "name": "MITCHELL", "callsign": "WDK6012", "shipType": "Spare - Local Vessel", "imo": "0", "dest": "HOUSTON TX", "postype": "1", "dimensions": "83", "timestamp": "1615137327" }
{ "currenttime": "2021-03-07 17:24:24", "eta": "215296", "draught": "100", "mmsi": "255805849", "name": "ULANGA", "callsign": "CQBU000", "shipType": "Cargo", "imo": "9690880", "dest": "KM YVAG", "postype": "1", "dimensions": "185", "timestamp": "1615137327" }
```

Figura 3-36 Comprobación de la existencia de registros en la cola *ship_info*

3.4.2 Generación del cubo OLAP

El primer paso para la generación de cubos es la creación de un proyecto nuevo en la interfaz gráfica de Kylin. Como se muestra en la Figura 3-37 el proyecto creado se denominó "ejemplo".

En cuanto a los cubos OLAP, Apache Kylin ofrece la capacidad de generarlos principalmente a partir de dos fuentes de datos distintas: una base de datos Hive o *streaming* con Kafka. Dentro de la opción *streaming*, se puede escoger entre el método *Near Real-Time* o *Real-Time* OLAP. La primera opción consiste en la ingesta de los datos de manera *micro-batch*. En este trabajo escogeremos la segunda opción debido a que el objetivo perseguido es realizar las consultas a datos en tiempo real.

Para ello, seleccionando la pestaña *Data Source*, el primer paso fue escoger la opción *Add Streaming Table V2* (véase Figura 3-37) y crear dos tablas, una para los datos de *routes* y otra para los de *ship_info*, a partir de las cuales se generarán los cubos. En estas tablas se establecen como columnas todos los campos que definen los datos. Es importante indicar el *topic* Kafka, el *Bootstrap Server* y el tipo de dato que se agrupará bajo cada columna (int, bigint y double si son números simples, date si se trata de un dato temporal en formato de fecha, etc.). También es necesario establecer una columna que se utilizará como *timestamp* (marca temporal que servirá de cronomarcador), en el caso de las colas Kafka empleadas ya venían con un parámetro establecido para ello (también hay que indicar si el tiempo a partir de este sello temporal se contará en segundos o milisegundos). Se debe añadir una muestra de los mensajes que conformarán la tabla para su correcta creación. En la Figura 3-38 y en la Figura 3-39 se muestra un ejemplo de este proceso con los datos *ship_info*.

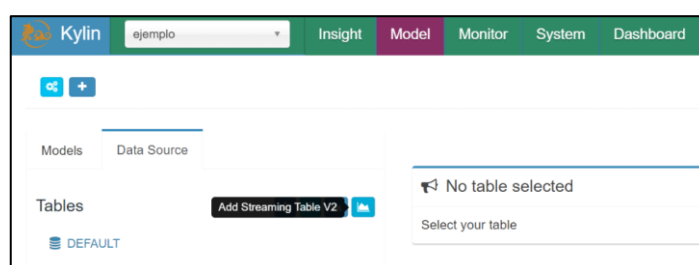


Figura 3-37 Elección de proyecto y tipo de *Data Source*

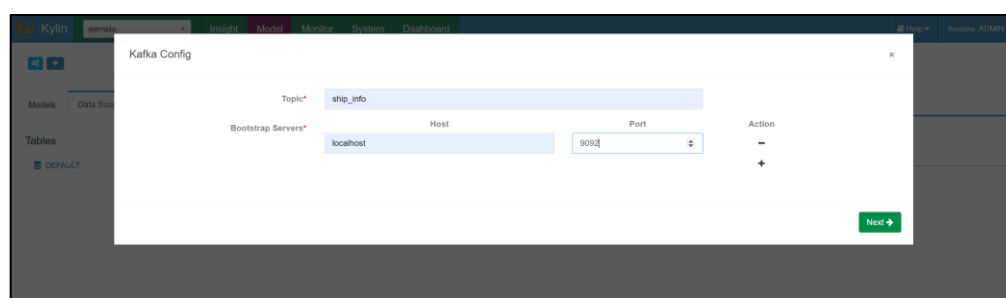


Figura 3-38 Elección del *topic* y del *Bootstrap Server*

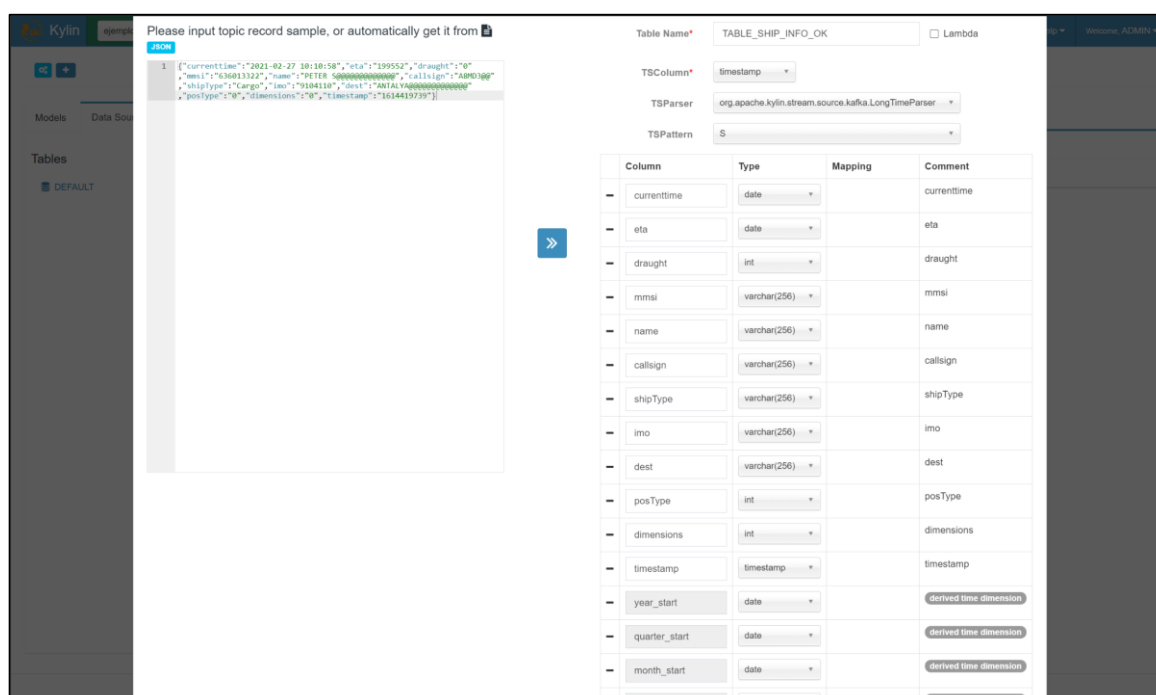


Figura 3-39 Creación de la tabla *ship_info*

Una vez creadas las tablas, fue necesaria la generación de modelos para emplear en la posterior creación de los cubos. Para la creación de estos modelos, se debe seleccionar la tabla de la que los cubos obtendrán los datos (Figura 3-40), las dimensiones que se emplearán (Figura 3-41) y la columna de partición (Figura 3-42). Al tratarse de *streaming* en tiempo real y debido a la gran cantidad de mensajes recibidos por segundo, en este proyecto se selecciona la columna MINUTE_START como columna de partición.

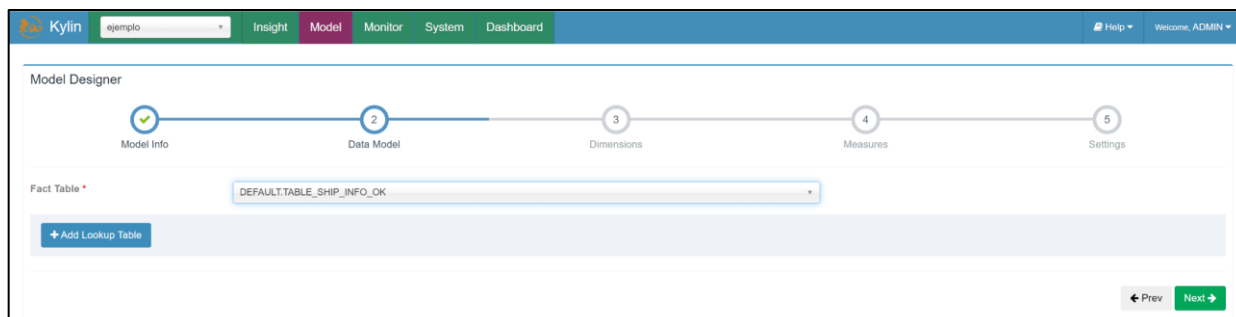


Figura 3-40 Selección de la tabla para el modelo

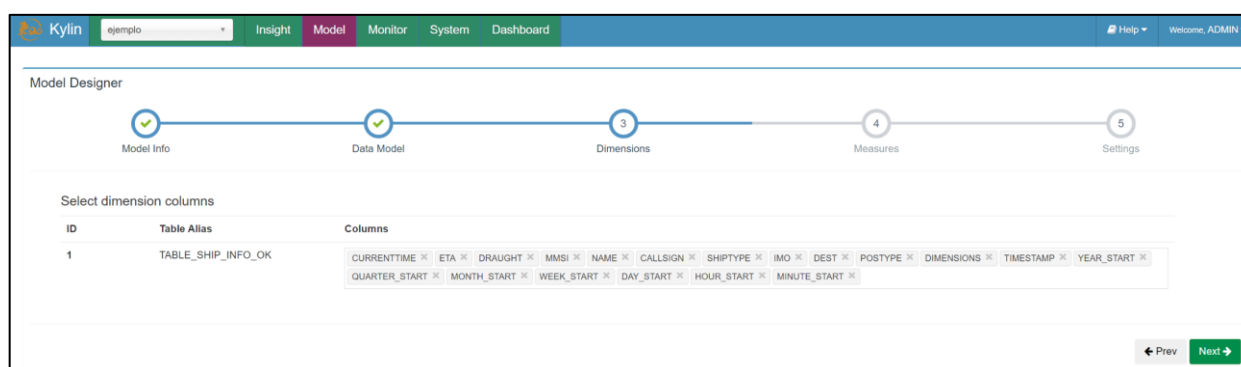


Figura 3-41 Definición de las dimensiones del modelo

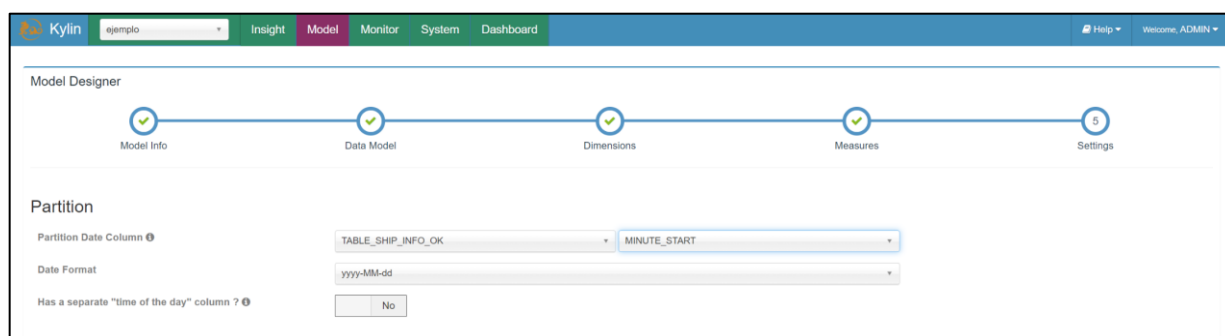


Figura 3-42 Definición de la columna de partición del modelo

El siguiente paso fue ya la creación de los cubos. Para ello hubo que indicar el modelo empleado (Figura 3-43), las dimensiones seleccionadas (Figura 3-44), las medidas (que en este caso se dejó la establecida por defecto: *count*), los umbrales para la compresión automática y la fecha del comienzo de particionamiento (Figura 3-45), la jerarquía entre ciertas dimensiones para facilitar y agilizar las consultas y la unión de dimensiones (en el caso del cubo de *routes* se pueden unir las dimensiones *latitude* y *longitude* para que siempre se muestren juntas) (Figura 3-46), el orden a establecer entre las entidades de una partición (Figura 3-47), la *Building Engine* (Figura 3-48), y el método de gestión de los segmentos inmutables (método *FULL_BUILD*) (Figura 3-49).

En cuanto al orden de las entidades de las particiones, o *Rowkey*, mencionado en el párrafo anterior, es recomendable, para un mejor procesamiento y análisis de los datos, colocar las dimensiones que

pueden servir de filtros antes, así como las dimensiones que toman valores únicos o poco comunes para cada partición (también llamados *high cardinality values*).

Cube Designer

1 2 3 4 5 6 7
Cube Info Dimensions Measures Refresh Setting Advanced Setting Configuration Overwrites Overview

Model Name * MODEL_SHIP_INFO_OK

Cube Name * CUBE_SHIP_INFO_OK

Notification Email List Comma Separated

Notification Events * ERROR x DISCARDED x SUCCEED x

Description

Next →

Figura 3-43 Definición del modelo del cubo

1 2 3 4 5 6 7
Cube Info Dimensions Measures Refresh Setting Advanced Setting Configuration Overwrites Overview

Add Dimensions Filter ...

ID	Name	Table Alias	Type	Column	Actions
1	CURRENTTIME	TABLE_SHIP_INFO_OK	normal	CURRENTTIME	
2	ETA	TABLE_SHIP_INFO_OK	normal	ETA	
3	DRAUGHT	TABLE_SHIP_INFO_OK	normal	DRAUGHT	
4	MMSI	TABLE_SHIP_INFO_OK	normal	MMSI	
5	NAME	TABLE_SHIP_INFO_OK	normal	NAME	
6	CALLSIGN	TABLE_SHIP_INFO_OK	normal	CALLSIGN	
7	SHIPTYPE	TABLE_SHIP_INFO_OK	normal	SHIPTYPE	
8	IMO	TABLE_SHIP_INFO_OK	normal	IMO	
9	DEST	TABLE_SHIP_INFO_OK	normal	DEST	
10	POSTYPE	TABLE_SHIP_INFO_OK	normal	POSTYPE	
11	DIMENSIONS	TABLE_SHIP_INFO_OK	normal	DIMENSIONS	
12	TIMESTAMP	TABLE_SHIP_INFO_OK	normal	TIMESTAMP	
13	YEAR_START	TABLE_SHIP_INFO_OK	normal	YEAR_START	
14	QUARTER_START	TABLE_SHIP_INFO_OK	normal	QUARTER_START	
15	MONTH_START	TABLE_SHIP_INFO_OK	normal	MONTH_START	
16	WEEK_START	TABLE_SHIP_INFO_OK	normal	WEEK_START	
17	DAY_START	TABLE_SHIP_INFO_OK	normal	DAY_START	
18	HOURL_START	TABLE_SHIP_INFO_OK	normal	HOURL_START	
19	MINUTE_START	TABLE_SHIP_INFO_OK	normal	MINUTE_START	

Figura 3-44 Definición de las dimensiones del cubo para *ship_info*

Cube Designer

1

2

3

4

5

6

7

Cube Info

Dimensions

Measures

Refresh Setting

Advanced Setting

Configuration Overwrites

Overview

Auto Merge Thresholds ⓘ

7

days

-

28

days

-

1

days

-

4

hours

-

0.5

hours

-

New Thresholds +

Volatile Range ⓘ

0

Retention Threshold ⓘ

0

Partition Start Date

2021-02-27 00:55:00

← Prev

Next →

Figura 3-45 Definición de los umbrales de compresión para el cubo ship_info

Cube Designer

1

2

3

4

5

6

7

Cube Info

Dimensions

Measures

Refresh Setting

Advanced Setting

Configuration Overwrites

Overview

Aggregation Groups

Visit [aggregation group](#) for more about aggregation group.

ID

Aggregation Groups

ⓘ Max Dimension Combination: 0

1

Includes

TABLE_SHIP_INFO_OK.CURRENTTIME ×

TABLE_SHIP_INFO_OK.ETA ×

TABLE_SHIP_INFO_OK.DRAUGHT ×

TABLE_SHIP_INFO_OK.MMSI ×

TABLE_SHIP_INFO_OK.NAME ×

TABLE_SHIP_INFO_OK.CALLSIGN ×

TABLE_SHIP_INFO_OK.SHIPTYPE ×

TABLE_SHIP_INFO_OK.IMO ×

TABLE_SHIP_INFO_OK.DEST ×

TABLE_SHIP_INFO_OK.POSTYPE ×

TABLE_SHIP_INFO_OK.DIMENSIONS ×

TABLE_SHIP_INFO_OK.TIMESTAMP ×

TABLE_SHIP_INFO_OK.YEAR_START ×

TABLE_SHIP_INFO_OK.QUARTER_START ×

TABLE_SHIP_INFO_OK.MONTH_START ×

TABLE_SHIP_INFO_OK.WEEK_START ×

TABLE_SHIP_INFO_OK.DAY_START ×

TABLE_SHIP_INFO_OK.HOUR_START ×

TABLE_SHIP_INFO_OK.MINUTE_START ×

Mandatory Dimensions

Select Column...

Hierarchy Dimensions

TABLE_SHIP_INFO_OK.YEAR_START ×

TABLE_SHIP_INFO_OK.QUARTER_START ×

TABLE_SHIP_INFO_OK.MONTH_START ×

TABLE_SHIP_INFO_OK.WEEK_START ×

TABLE_SHIP_INFO_OK.DAY_START ×

TABLE_SHIP_INFO_OK.HOUR_START ×

TABLE_SHIP_INFO_OK.MINUTE_START ×

New Hierarchy +

Joint Dimensions

New Joint +

Figura 3-46 Definición de la jerarquía y unión entre dimensiones para el cubo ship_info

Rowkeys ⓘ

Important: Dimension's position on HBase rowkey is critical for performance. You can drag and drop to adjust the sequence. In short, put filtering dimension before non-filtering dimension, and put high cardinality dimension before low cardinality dimension.

ID	Column	Encoding	Length	Shard By
1	TABLE_SHIP_INFO_OK.MINUTE_START	dict	0	false by default
2	TABLE_SHIP_INFO_OK.HOUR_START	dict	0	false by default
3	TABLE_SHIP_INFO_OK.DAY_START	dict	0	false by default
4	TABLE_SHIP_INFO_OK.WEEK_START	dict	0	false by default
5	TABLE_SHIP_INFO_OK.MONTH_START	dict	0	false by default
6	TABLE_SHIP_INFO_OK.QUARTER_START	dict	0	false by default
7	TABLE_SHIP_INFO_OK.YEAR_START	dict	0	false by default
8	TABLE_SHIP_INFO_OK.SHIPTYPE	dict	0	false by default
9	TABLE_SHIP_INFO_OK.DIMENSIONS	dict	0	false by default
10	TABLE_SHIP_INFO_OK.DRAUGHT	dict	0	false by default
11	TABLE_SHIP_INFO_OK.POSTTYPE	dict	0	false by default
12	TABLE_SHIP_INFO_OK.CURRENTTIME	dict	0	false by default
13	TABLE_SHIP_INFO_OK.ETA	dict	0	false by default
14	TABLE_SHIP_INFO_OK.MMSI	dict	0	false by default
15	TABLE_SHIP_INFO_OK.NAME	dict	0	false by default
16	TABLE_SHIP_INFO_OK.CALLSIGN	dict	0	false by default

Figura 3-47 Definición del *rowkey order* para el cubo *ship_info*

Cube Engine ⓘ

Engine Type : MapReduce

Figura 3-48 Definición del *cube engine* para el cubo *ship_info*

Cube Designer

✓ Cube Info
 ✓ Dimensions
 ✓ Measures
 ✓ Refresh Setting
 ✓ Advanced Setting
 6 Configuration Overwrites
 7 Overview

Kylin Properties

[+ Property](#)

Streaming Properties

kylin.cube.algorithm	INMEM
kylin.stream.cube.window	3600
kylin.stream.cube.duration	3600
kylin.stream.index.checkpoint.intervals	300
kylin.stream.segment.retention.policy	fullBuild purge

Tips

1. Cube level properties will overwrite configuration in kylin.properties

← Prev
Next →

Figura 3-49 Definición de las propiedades de *Kylin streaming* para el cubo *ship_info*

Por último, fue necesario añadir el *Streaming Receiver* disponible a un *Replica Set*, este proceso se muestra en la Figura 3-50 y en la Figura 3-51.

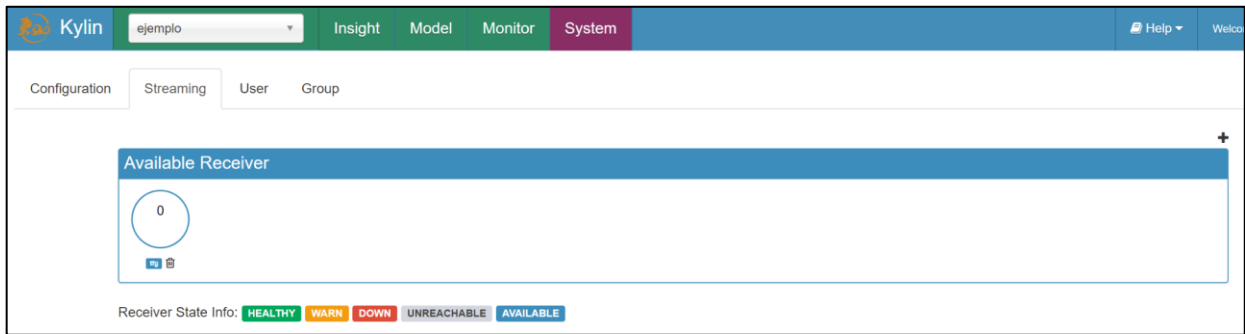


Figura 3-50 Definición de los receptores accesibles para el cubo *ship_info*

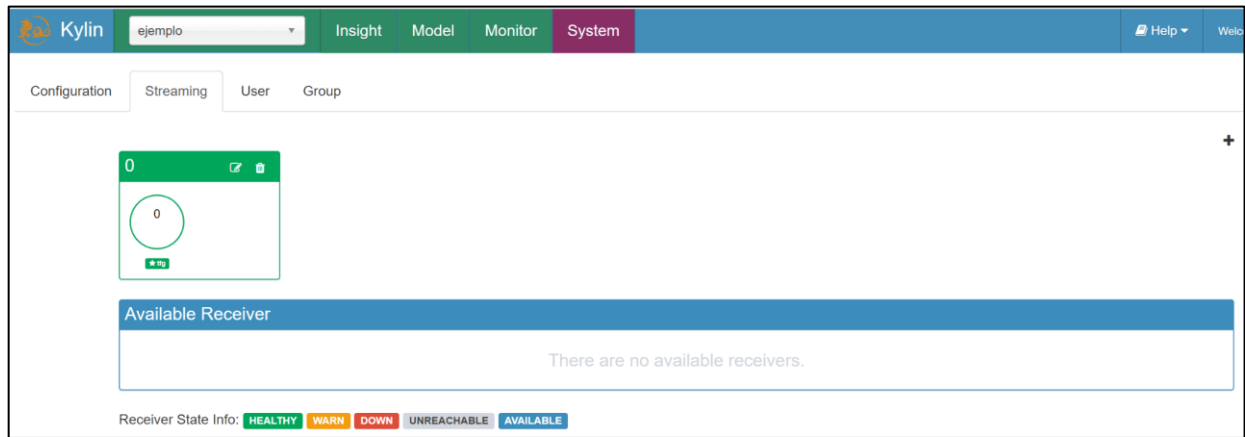


Figura 3-51 Definición de los *replica set* establecidos para el cubo *ship_info*

La creación del cubo asociado a los datos de la cola *routes* es equivalente en todos los pasos.

Para activar o desactivar el cubo, se selecciona el desplegable mostrado en la Figura 3-52, donde también se puede observar que efectivamente el cubo está recibiendo mensajes.

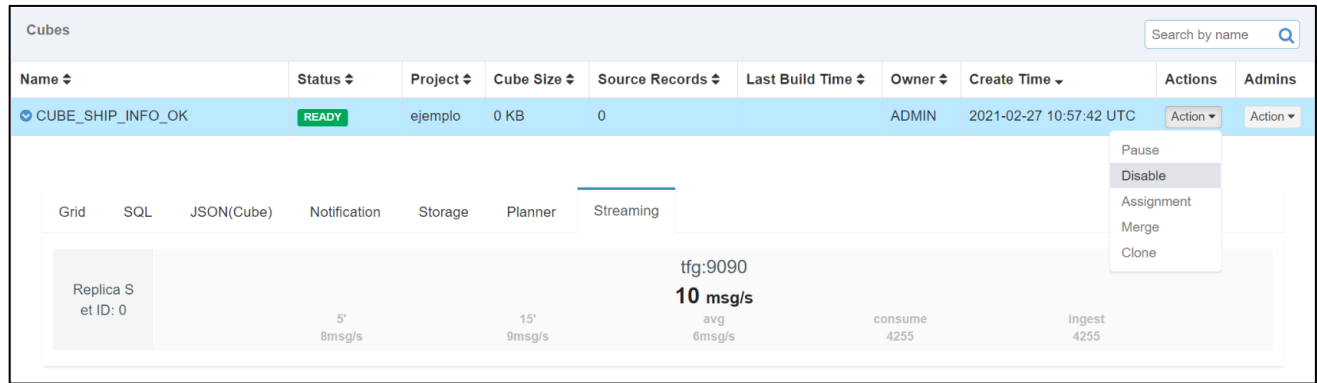


Figura 3-52 Cubo *ship_info* activado y recibiendo mensajes de la cola Kafka

Una vez completado este proceso se puede comprobar el funcionamiento del cubo realizando consultas a las tablas de datos a través de estos. En la Figura 3-53 se demuestra la gran capacidad de procesamiento del cubo mediante la consulta a la tabla de datos *ship_info*, donde se observa la obtención de los 606 mensajes AIS recibidos hasta el momento en 0,15 segundos.

The screenshot displays the Apache Superset web interface. At the top, there are navigation tabs: Insight, Model, Monitor, System, and Dashboard. Below these, there are sections for 'New Query', 'Saved Queries', and 'Query History'. The 'New Query' section contains a SQL editor with the query: `SELECT * FROM TABLE_SHIP_INFO_OK;`. Below the editor, there are fields for 'Project' (ejemplo), 'LIMIT' (50000), and a 'Submit' button. The 'Results' section shows the query string, status (Success), project (ejemplo), and cubes (CUBE[name=CUBE_SHIP_INFO_OK]). The results table has 606 rows and 10 columns: CURRENTTIM., ETA, DRAUGHT, MMSI, NAME, CALLSIGN, SHIPTYPE, IMO, DEST, and POSTYPE. The first row of data is: 2021-02-27, 1970-01-01, 0, 123432109, CAPTAIN CON..., 100200@, Other, 0, @@@@..., 0.

Figura 3-53 Consulta a tabla de datos *ship_info*

3.4.3 Establecimiento de conexión entre Apache Kylin y Apache Superset

El establecimiento de la conexión entre las dos herramientas principales del sistema, Apache Kylin y Apache Superset, es relativamente sencillo.

Primero hay que descargar *kylinpy* (*Apache Kylin Python Client Library*) que establece un cliente Kylin basado en Python y permite las consultas a cualquier aplicación que use el dialecto SQL Alchemy (véase Figura 3-54).

```
hadoop@tfq:~$ pip3 install kylinpy
Processing ./cache/pip/wheels/0a/7b/dd/e1ba8fb917935b6805e40731d94f1456fb8bc27bca8373aba2/kylinpy-2.8.4-
Installing collected packages: kylinpy
Successfully installed kylinpy-2.8.4
```

Figura 3-54 Instalación de *kylinpy*

Tras esto se puede acceder a los datos almacenados en Apache Kylin desde la interfaz gráfica de Apache Superset mediante el empleo del identificador mostrado en la Figura 3-55. Este identificador tiene la estructura siguiente: *kylin://userpassword@host:port/database*. Apache Superset permite comprobar la calidad de la conexión que se pretende realizar antes de guardarla.

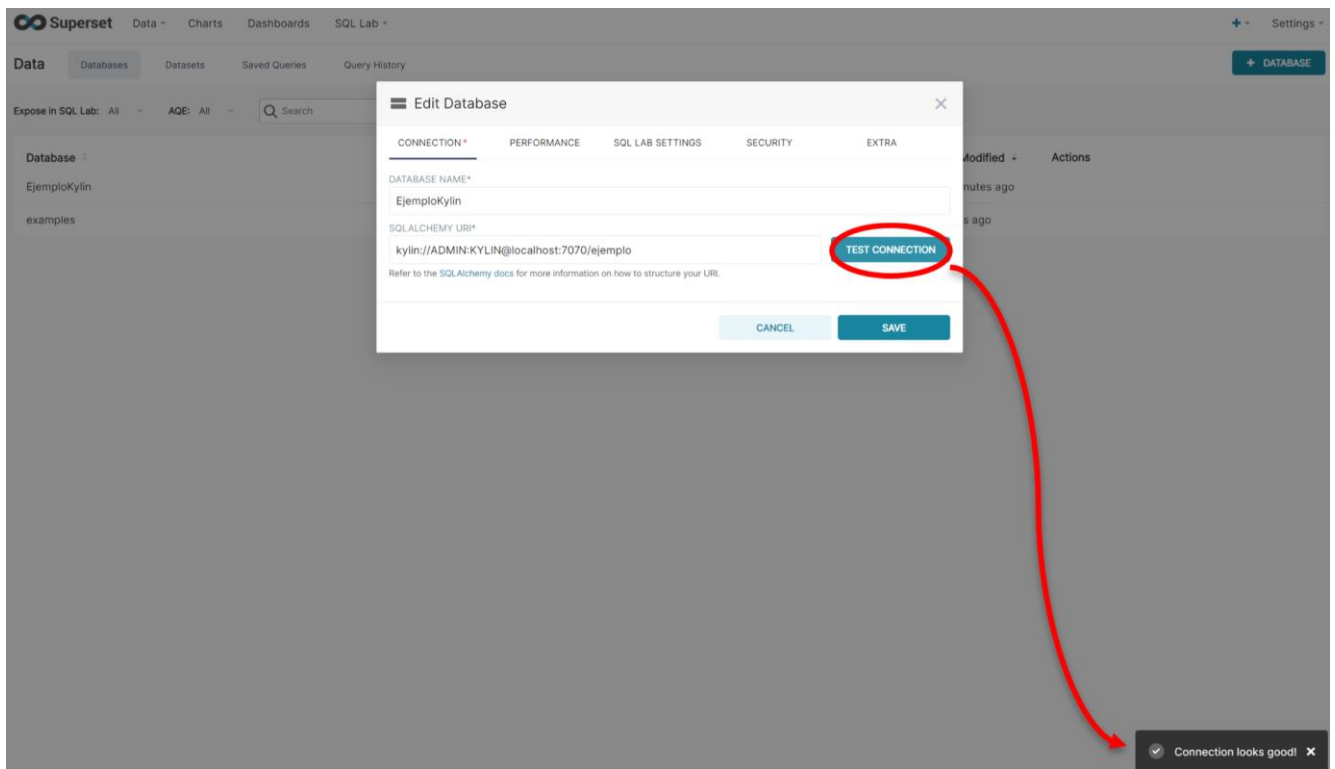


Figura 3-55 Conexión Apache Superset-Apache Kylin

3.4.4 Análisis de la información

El primer paso para la creación de los dashboards que servirán de cuadro de mando y control para el COVAM es seleccionar la fuente de datos de la que se extraerá la información. Para ello, en el desplegable de la pestaña *Data* seleccionamos *Datasets*, ahí podremos añadir una nueva fuente como se muestra en la Figura 3-56 y en la Figura 3-57.

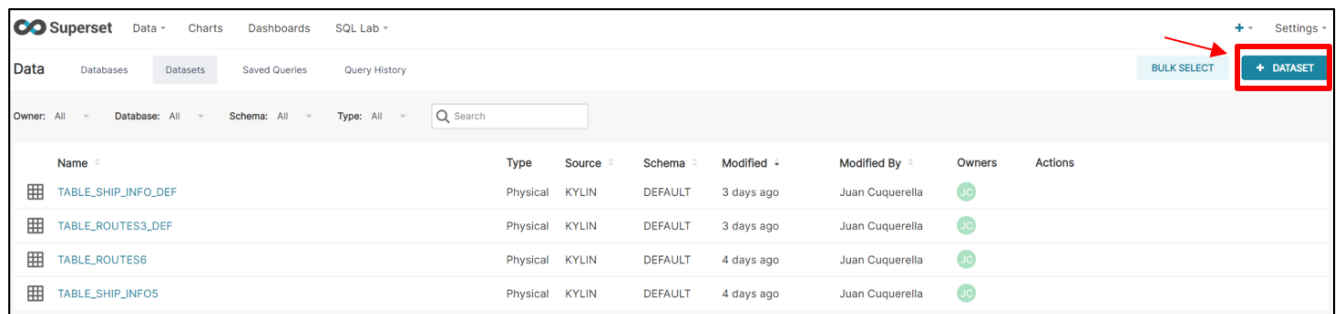


Figura 3-56 Datasets de la interfaz gráfica de Superset

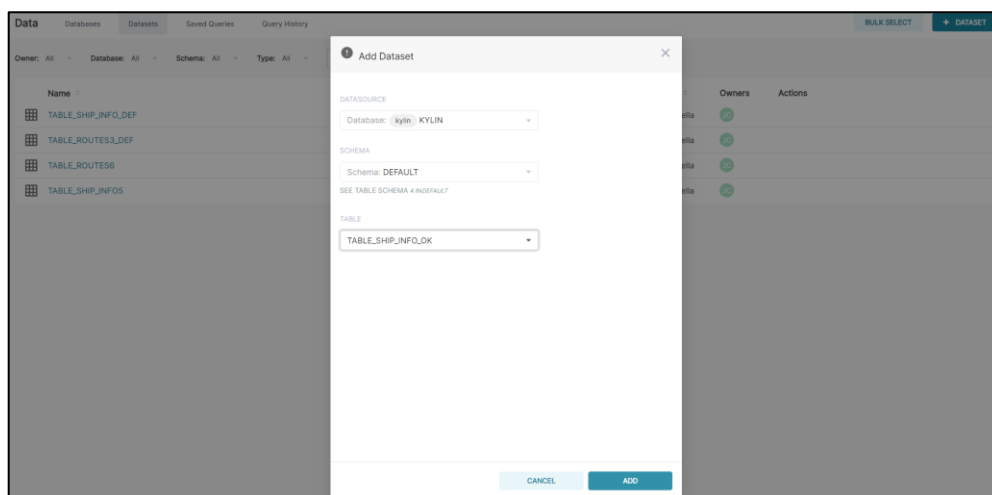


Figura 3-57 Adición de *Datasets*

Una vez añadidas las conexiones a las fuentes de datos, se pudo comenzar a crear los dashboards. Para ello, Superset ofrece una gran variedad de gráficos, tablas, mapas y figuras para la presentación de la información. A cada uno de estos elementos se le denomina *chart*.

Aunque es posible realizar consultas SQL de forma nativa, mediante la opción *SQL Lab*, Superset proporciona una sencilla interfaz para consultas en la que, mediante desplegaables, permite seleccionar la operación a realizar, métrica empleada, columna y granularidad del tiempo, agrupaciones necesarias, filtros que se desean establecer y el tipo de *chart* que se desea utilizar, entre otras. Ofrece además un elevado nivel de personalización en el diseño de cada uno de los *charts* empleados.

Como ejemplo, en la Figura 3-58 se puede observar la realización de una consulta sencilla (obtener el número de embarcaciones, agrupadas según el *Shiptype*, en la zona establecida como Península) empleando el método *SQL Lab*. La Figura 3-59 muestra la misma consulta empleando el método de desplegaables. Comparando ambas imágenes se puede observar que el segundo método proporciona una rapidez y sencillez de uso muy útil para consultas simples. Sin embargo, debido a las limitaciones del segundo método, las consultas más complejas se realizarán empleando *SQL Lab*. Estas figuras también sirven de demostración del gran potencial de procesamiento y análisis del sistema, obteniendo resultados de más de diez mil mensajes AIS, en menos de medio segundo.

SHIPTYPE	count
Cargo	3163
Other	2150
Fishing	1559
Tanker	1249
Tug	1083
Passager	964
Spare - Local Vessel	548
Pleasure	481
Pilot Vessel	215

Figura 3-58 Ejemplo de consulta con *SQL Lab*

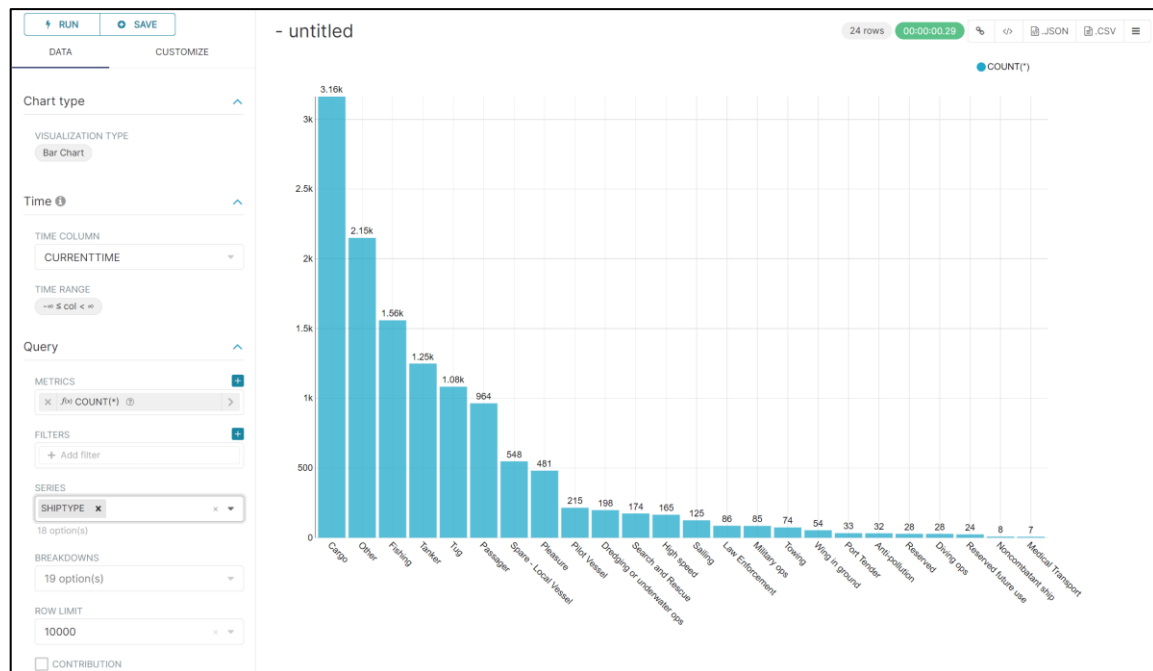


Figura 3-59 Ejemplo de consulta con el método de desplegables de Superset

A continuación, en la Figura 3-60 se muestran todas las alternativas de *charts* que proporciona Superset. Sin embargo, solo se destacarán los que han sido empleados para la creación de los *dashboards* por considerarse de mayor relevancia para la obtención de resultados de interés para el COVAM.

1. Table

Este método de visualización permite obtener una tabla en la que se muestra el número total de los mensajes recibidos agrupándolos según los diferentes campos seleccionados, además de la posibilidad de filtrar la información empleando las distintas dimensiones de la fuente de datos que se esté usando. Proporciona la capacidad de ordenar los mensajes en orden creciente o decreciente de una dimensión específica. Para el funcionamiento de este *chart* es importante seleccionar la columna y granularidad temporal, así como el rango de tiempo de los datos que se pretenden analizar, es decir, permite establecer un filtro temporal.

Este *chart* consta de un apartado de diseño que permite personalizar parámetros como el cambio de color de las celdas que contengan valores negativos, escoger el formato de la tabla o la adición de una *search box* para facilitar una rápida búsqueda de un dato específico.

2. Filter Box

Mediante la adición de un *Filter Box* a un *dashboard* se puede realizar el filtrado de todos los demás *charts* pertenecientes a este, incluso si el *dashboard* ha sido ya creado, publicado y/o exportado previamente. Los filtros son seleccionados de entre las diferentes dimensiones de la tabla fuente. En caso de no pertenecer algún *chart* al mismo *Data Set* que el *Filter Box*, este solo actualizará sus datos filtrando los campos que tengan en común. Entre las opciones de configuración que presenta este *chart* destaca la opción *instant filtering*, que permite actualizar los datos de los *charts* a medida que se van seleccionando los filtros.

También requiere de la selección de una columna y granularidad temporal, así como un rango de tiempo.

Esta herramienta sirve como demostración de la gran capacidad de personalización de los *dashboards* creados de la que dispone Superset como herramienta de visualización de datos.

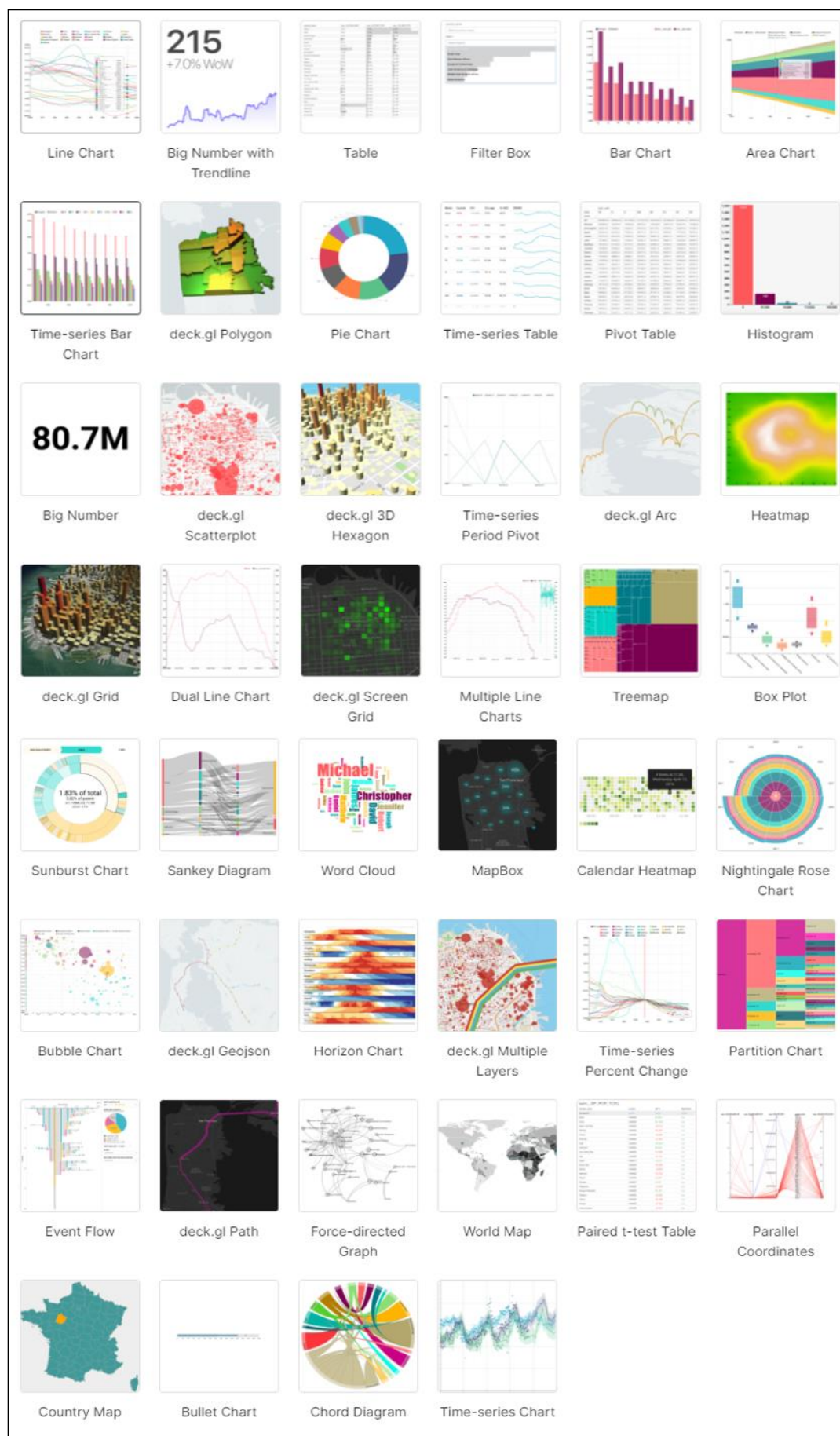


Figura 3-60 Conjunto de *charts* disponibles en Apache Superset

3. Map Box

Este *chart* consiste en la visualización de datos geolocalizados en un mapa y, por tanto, exige la existencia y selección de dimensiones longitud y latitud. También permite la selección de la columna temporal y el *time range* (aunque puede seleccionarse como *no filter*).

Para la creación de este *chart* es necesario una instalación previa del *token* (mapa) sobre el que se va a representar la información. Para ello se hace uso de la página web *Map Box* [84], en la cual se puede crear un mapa propio o adquirir la URL de uso público. La URL del mapa seleccionado se ha de añadir en el fichero de configuración *config.py* dentro del directorio de configuración de Superset (véase Figura 3-61). La Figura 3-62 muestra la página principal de *Map box*.

```
# pass
QUERY_LOGGER = None

# Set this API key to enable Mapbox visualizations
MAPBOX_API_KEY = "pk.eyJ1IjoianVhbmN1cXVlcmVsbGEiLCJhIjoIY2tsaDB0dHB2MGV6b2Jvb2JlI3dDM4c2c4NiJ9.rXbjfCNBCYXL5ZdwYDH30w"

# Maximum number of rows returned from a database
# in async mode, no more than SQL_MAX_ROW will be returned and stored
# in the results backend. This also becomes the limit when exporting CSVs
SQL_MAX_ROW = 100000
```

Figura 3-61 Adición del URL público de un mapa *Map Box* en *config.py*

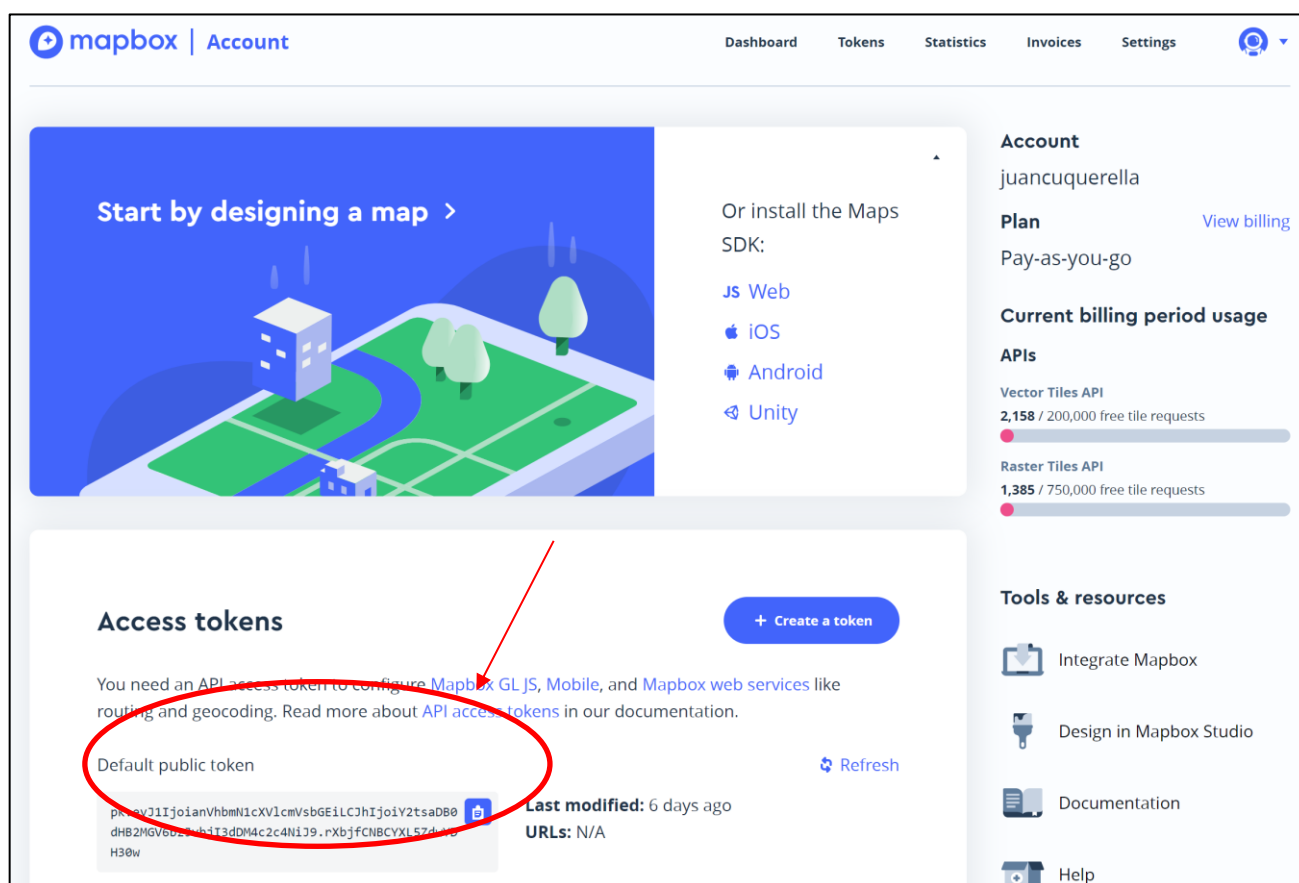


Figura 3-62 Página principal de *Map Box*

Entre las opciones de configuración que permite este *chart* se encuentran las siguientes: selección del radio de *clustering* (agrupa los datos que se encuentran dentro de este radio, indicando el número de estos, para dotar de mayor claridad al mapa), establecimiento de etiquetas a cada punto mostrado en el mapa para una fácil identificación de estos, establecimiento de filtros y una gama de estilos de mapa entre los que elegir.

4. Time-series Bar Chart

Estos gráficos presentan datos en relación con el tiempo. Según la métrica empleada pueden mostrarse los datos en el eje de ordenadas como la suma de los valores (SUM), la cuenta total de datos o mensajes (COUNT), la media (AVG), el valor máximo (MAX) o el valor mínimo (MIN). En el eje de abscisas se muestra la serie temporal (la columna temporal escogida) dividida según la granularidad seleccionada.

Este gráfico permite agrupar los datos mediante la selección de uno o varios campos, establecer filtros y la personalización del diseño mediante la adición de leyendas, visualización de los valores de las barras, selección del tipo de cuadrícula de fondo, elección de los colores y la adición de títulos a los ejes.

Este *chart* también permite la opción de que se muestre en una etiqueta los datos de cada barra (valor, tiempo y agrupación a la que pertenece) al desplazar el cursor por encima.

5. Pie Chart

Esta opción muestra los datos agrupados según el campo seleccionado en secciones circulares. Presenta la información de manera intuitiva y sencilla y, por tanto, es de gran utilidad. Es necesaria la selección de una columna temporal, así como de un *time range*.

Permite establecer filtros en la información recibida para mostrar únicamente los datos deseados y consta de una gran capacidad de personalización mediante la selección de los colores, la adición de leyendas y etiquetas y el control del radio exterior e interior (en caso de querer mostrar el gráfico en forma de disco).

Según la métrica establecida mostrará los valores como una de las opciones definidas en *Time-series Bar Chart*.

6. Big Number

Este *chart* muestra el resultado de una consulta simple como un único número de gran tamaño, ideal para mostrar datos de importancia o urgencia.

Como el resto exige el establecimiento de una columna temporal y un *time range*, y permite el filtrado de la información. Este número puede resultar de la operación de cualquiera de las métricas definidas en *Time-series Bar Chart*.

Permite mostrar un subtítulo debajo del número y cambiar el formato tanto del número como del subtítulo.

7. Sunburst Chart

Muy similar al *Pie Chart*, la diferencia es que se pueden escoger distintas dimensiones para la agrupación de los datos y establecer una jerarquía entre estas (representando las dimensiones jerárquicamente superiores como los anillos interiores y las inferiores como los anillos exteriores). Por tanto, es una herramienta capaz de mostrar un elevado contenido de información de manera sencilla y muy visual.

Dispone de etiquetas y leyendas interactivas que cambian a medida que el cursor se sitúa en las diferentes secciones de los diferentes anillos. Muestra en el centro el porcentaje que representa la sección seleccionada sobre el total y sobre la dimensión de la que depende.

Por último, se procede a explicar detalladamente el proceso de diseño de cada uno de los *charts* específicos que conformarán los *dashboards* que se diseñan en este TFG, así como su utilidad y la información que aportan.

Los primeros cinco elementos se enfocan a la obtención de información del Estrecho de Gibraltar, los siguientes cuatro elementos (del 6 al 9) están destinados al análisis de zonas y actividades de interés,

por último, con los siguientes cuatro elementos (del 10 al 13) se busca obtener una imagen global del entorno marítimo.

1. Mapa Estrecho de Gibraltar

En este chart se muestra un mapa en el que se pueden observar todas las embarcaciones que se encuentran transmitiendo en AIS en el Estrecho de Gibraltar. Esta herramienta es de gran utilidad para el COVAM ya que permite, con un solo vistazo, obtener una visión global y en tiempo real del tráfico existente en esta zona. Su utilidad incrementa notablemente con la adición de un *Filter Box* que permita realizar estudios más selectivos sobre los datos en esta zona.

Se empleó *SQL Lab* para la definición de la consulta, al ser esta relativamente compleja. Una vez guardada la consulta, se seleccionó *Map Box* como método de visualización de la información deseada.

En la Figura 3-63 se muestra la consulta realizada, utilizando como fuente de datos el flujo *routes*. Como se puede observar, se empleó *SELECT DISTINCT (MMSI)* para evitar mostrar en pantalla mensajes pertenecientes a una misma embarcación en diferentes momentos de emisión, seleccionando únicamente el último mensaje enviado por cada MMSI. También se estableció un *time range* de 30 minutos, evitando así la presentación en el mapa de embarcaciones que probablemente ya no se encuentren en la posición especificada (por ejemplo, por haber llegado a puerto y apagado el AIS). Por último, se delimitó la zona del Estrecho mediante un filtro en los campos de longitud y latitud, obteniendo así únicamente los datos pertenecientes al área seleccionada en la consulta.

En la Figura 3-64, Figura 3-65 y Figura 3-66 se puede observar cómo, mediante el empleo de este *chart*, se podría mantener una monitorización de la situación de tráfico en zonas de interés. Estas figuras muestran las embarcaciones que se encuentran cruzando el Estrecho en tres instantes de tiempo separados 5 minutos entre sí.

```
SELECT "LAT" AS "LAT",
       "LON" AS "LON"
FROM
  (SELECT DISTINCT(MMSI),
    ESLORA,
    CURRENTTIME,
    SHIPTYPE,
    NAVSTATUS,
    DESTINATION,
    LAT,
    LON,
    MINUTE_START
  FROM "DEFAULT"."TABLE_ROUTES_OK"
  ORDER BY MINUTE_START DESC) AS expr_qry
WHERE "MINUTE_START" ≥ CAST('2021-03-02 15:59:30' AS TIMESTAMP)
  AND "MINUTE_START" < CAST('2021-03-02 16:29:30' AS TIMESTAMP)
  AND "LAT" < 36.14
  AND "LAT" > 35.74
  AND "LON" > -6.3
  AND "LON" < -5.0
LIMIT 5000;
```

Figura 3-63 Consulta para el *chart* "Mapa Estrecho de Gibraltar"

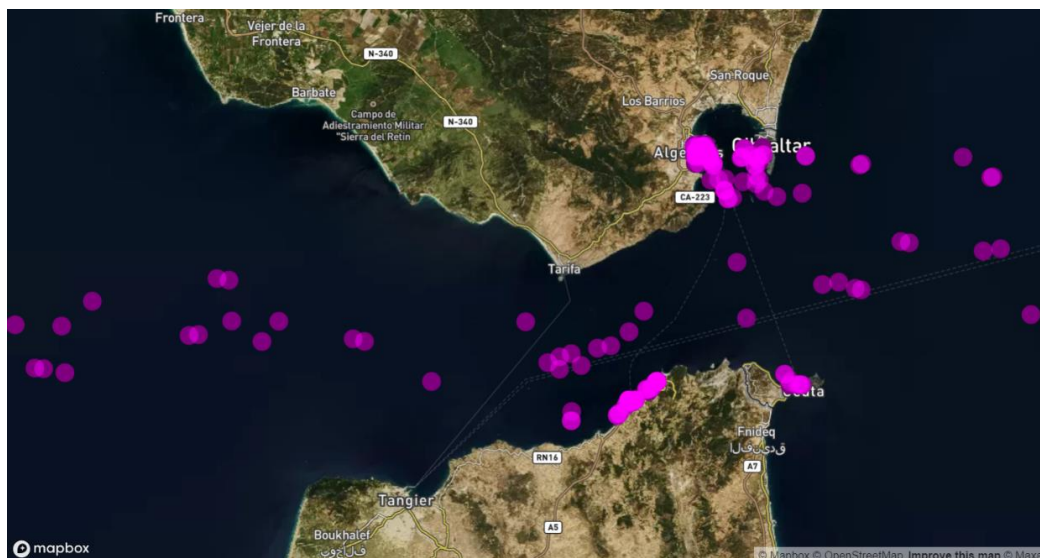


Figura 3-64 Mapa Estrecho de Gibraltar

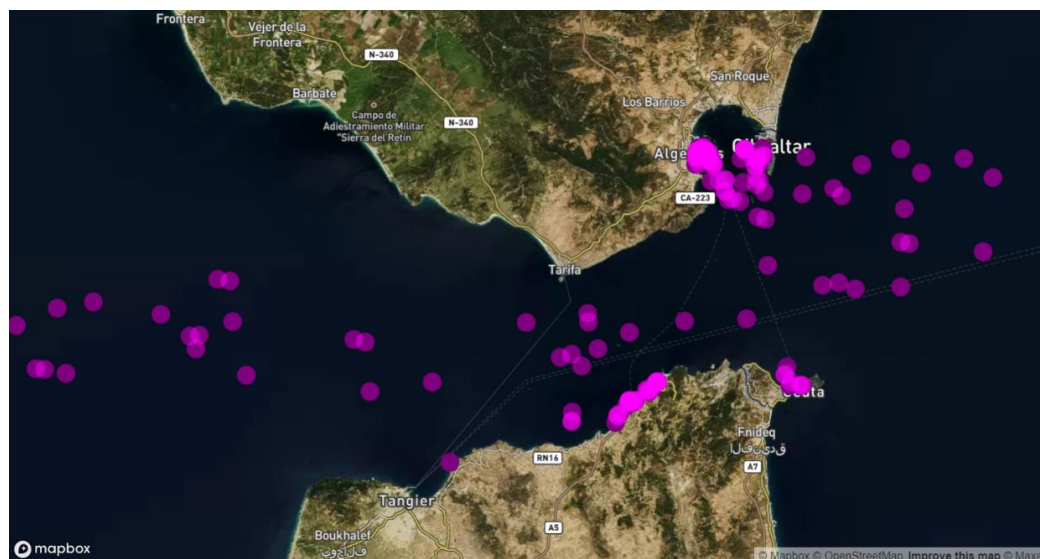


Figura 3-65 Mapa Estrecho de Gibraltar + 5 minutos

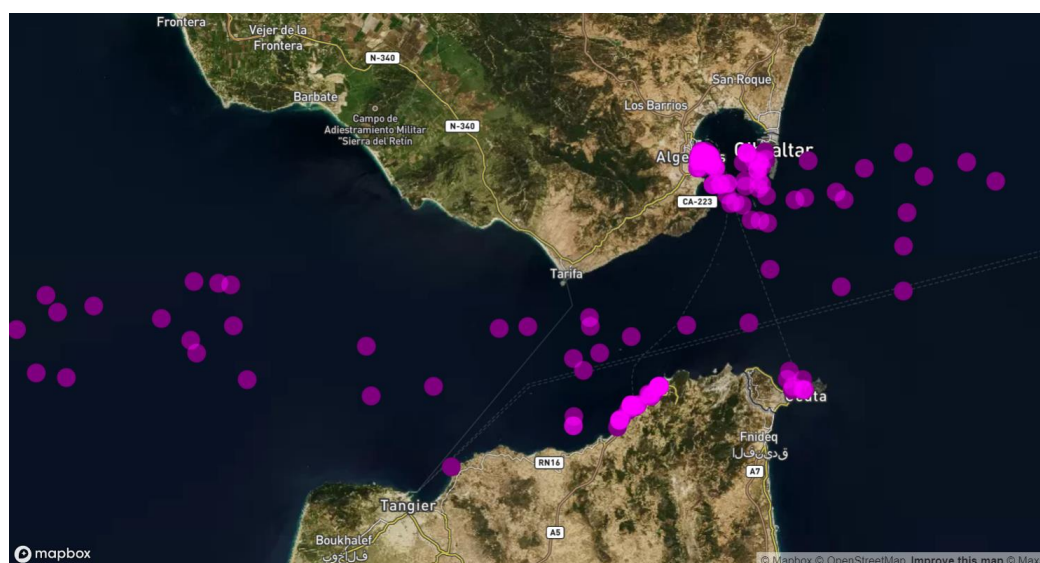


Figura 3-66 Mapa Estrecho de Gibraltar + 10 minutos

2. Filter Box

Como se ha mencionado en la definición del Mapa Estrecho de Gibraltar, este *chart* sirve para potenciar el análisis selectivo en el Estrecho. Mediante la agregación de este *Filter Box* al *dashboard*, se puede filtrar los mensajes AIS recibidos para así mostrar en el mapa únicamente los buques que cumplan ciertas características especificadas. Este filtrado se realiza automáticamente en el mapa a medida que se define en el *Filter Box* gracias a la selección de la opción *instant filtering*.

El diseño de este *chart* se realizó mediante el método de desplegables (véase Figura 3-67). De esta manera, se seleccionaron como campos filtro los siguientes: *shiptype*, *navstatus*, *destination*, *zone* y MMSI. Por tanto, se puede seleccionar la visualización en el mapa de un tipo de embarcación específico, embarcaciones que se encuentran en una situación de navegación concreta, los buques según su destino y, en caso de querer analizar un único barco en específico, un MMSI concreto.

Para el diseño de este *chart* se adquirieron las dimensiones establecidas en la tabla de datos de la cola *routes*.

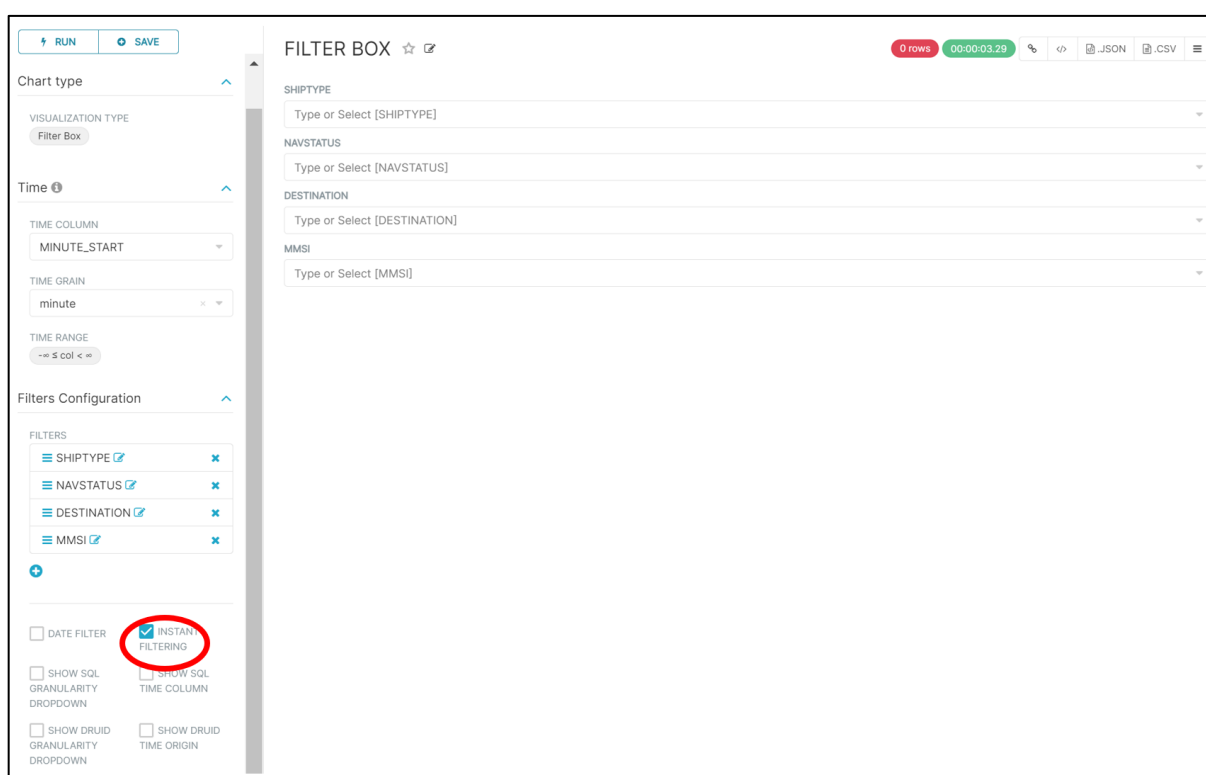


Figura 3-67 Diseño del *Filter Box*

3. Número de embarcaciones a la deriva

Empleando el *chart Big Number* se muestra el número de embarcaciones que, según los parámetros introducidos en AIS, se encuentran navegando sin control (Figura 3-68). El empleo de *Big Number* dota a este dato del grado de urgencia que le corresponde. Sin embargo, es posible que este parámetro haya sido establecido por defecto o sin querer en el campo *navstatus* del mensaje AIS de una embarcación y, por tanto, no ser siempre veraz. Para permitir un estudio de la situación más en profundidad y poder comprobar la veracidad del dato, se puede añadir al *dashboard* una tabla que proporcione más información al respecto (como se explicará en el siguiente *chart*).

Para el diseño de este *chart* se empleó como fuente de información los datos obtenidos de la cola *routes* y se establecerán, empleando el método de desplegables, los mismos límites geográficos y temporales establecidos en el *chart* Mapa Estrecho de Gibraltar. Además, se añadió el filtro de *navstatus* igual a 2 (sin control).



Figura 3-68 Diseño del *chart* "Número de embarcaciones a la deriva"

4. Embarcaciones a la deriva

Este *chart* se diseñó, en formato de tabla, para complementar el anterior, añadiendo información sobre las posibles embarcaciones a la deriva (Figura 3-69). Se decidió añadir pocos campos a esta tabla para no saturar la pantalla de datos y así poder obtener la información de estas embarcaciones de manera concisa, rápida y eficaz. Las dimensiones seleccionadas fueron el MMSI y la latitud y longitud. De esta forma, en caso de observar en el *Big Number* la existencia de una posible embarcación a la deriva, se podrá referir a este mediante radiocomunicaciones de manera inconfundible mediante el MMSI y ubicarlo en un RADAR o carta náutica mediante su latitud y longitud.

Los parámetros de diseño (fuente de datos, método de diseño y filtros empleados) son los mismos que en el *chart* anterior.



Figura 3-69 Diseño del *chart* "Embarcaciones a la deriva"

En la Figura 3-70 se muestra un ejemplo de estos dos últimos *charts* trabajando en conjunto.

A LA DERIVA

☰1

⋮

1

EMBARCACIONES A LA DERIVA

☰1

⋮

Time	MMSI	LON	LAT
2021-03-02 17:05	538009051	-5.146330000000001	36.08124166666666

Figura 3-70 Información de una embarcación a la deriva en un panel de control

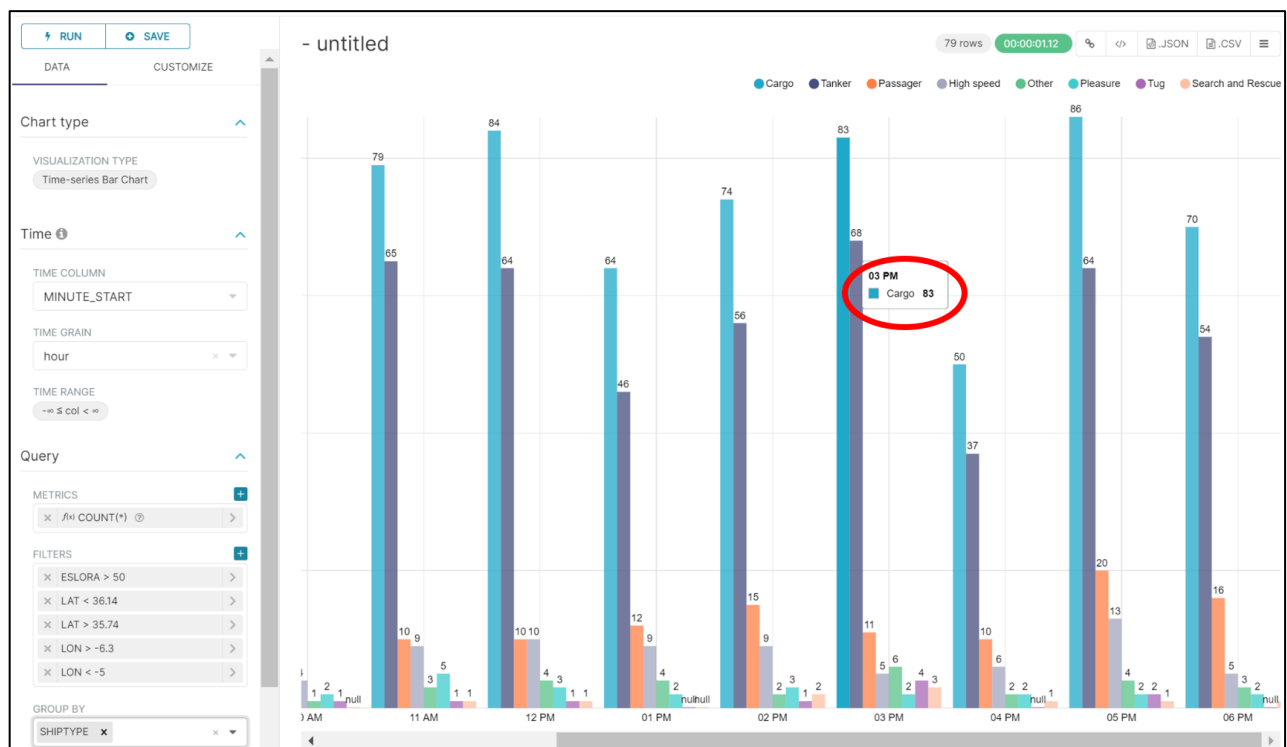
5. Gráfico de tráfico de buques de gran tamaño

Este gráfico se diseñó para poder obtener un histórico de los buques considerados de gran tamaño y, por tanto, de especial relevancia que cruzan el Estrecho de Gibraltar. De esta manera, se pueden realizar estudios y análisis del tráfico marítimo en esta zona tan frecuentada, al contener unas de las rutas de tránsito marítimo más importantes del mundo. En el eje de ordenadas se indica el número de embarcaciones y en el eje de abscisas el tiempo transcurrido (agrupado en horas en este caso).

Este gráfico agrupa los mensajes según el tipo de embarcación que los emite, para ello se tuvo que señalar *shiptype* como dimensión de agrupación. Se indica con una leyenda qué color representa cada tipo de embarcación y se añadió la opción de mostrar, mediante una etiqueta, los valores de la barra sobre la que se sitúa el cursor, como se señala en la Figura 3-71.

Se consideraron buques de gran tamaño aquellos cuya eslora es superior a 50 metros, como indica el RIPAM (Reglamento Internacional para Prevenir los Abordajes en la Mar) [85]. Por tanto, empleando el método de desplegables, se filtró el campo *dimensions*. También se estableció el filtro geográfico mencionado en el *chart* Mapa Estrecho de Gibraltar.

Se emplearon los datos obtenidos de la cola *routes* y el tipo de *chart* empleado fue *Time-series Bar Chart*.

Figura 3-71 Diseño del *chart* "Gráfico de tráfico de buques de gran tamaño"

6. Posible transbordo de pesca en la mar

El sistema AIS contiene información cinemática, relativa al viaje y a las características del buque emisor, cuyo análisis permite la detección de anomalías, que se producen cuando los datos recibidos no son coherentes con los del resto del tráfico marítimo o con los datos históricos almacenados del mismo barco. Una de estas anomalías que se pretende detectar, siguiendo unas pautas establecidas tras una serie de interacciones con el COVAM, es la detección de transbordos de pesca en la mar.

Es importante destacar que esta actividad no es ilícita de por sí, siempre que se haga conforme a la legislación vigente, pero el COVAM asegura la necesidad de su detección para poder verificar si se trata de una actividad autorizada o no.

Las pautas obtenidas para la detección de transbordo de pesca en la mar son las siguientes: pesqueros que interrumpen su actividad para permanecer parados o a menos de dos nudos en alta mar y, posteriormente, retoman la navegación.

Para ello se diseñó un mapa, empleando *Map Box*, en el cual se muestran únicamente los barcos pesqueros navegando a menos de 2 nudos en la zona de la Península. Como se puede observar en la Figura 3-73, todos los contactos obtenidos se encuentran situados en la costa, lo cual tiene sentido ya que estos barcos seguramente se encuentren faenando en aguas costeras o entrando y saliendo de puerto. Sin embargo, en caso de obtenerse dos contactos muy próximos entre sí y alejados de la costa podría tratarse de un transbordo de pesca en la mar.

Debido a la complejidad de la consulta (Figura 3-72), se empleó *SQL Lab* para su realización. Para evitar la saturación del mapa con mensajes pertenecientes a una misma embarcación se empleó el mismo método que el descrito en la definición del *chart* Mapa Estrecho de Gibraltar. Los filtros establecidos fueron los siguientes: *shiptype* igual a *fishing* (pesquero), SOG menor que 2 nudos (importante recordar que la velocidad en los mensajes AIS se muestra en decimas de nudo, como se expuso en el apartado 3.1.2) y *zone* igual a la Península.

Los datos mostrados en este mapa son obtenidos de la cola *routes*.

```
SELECT "LON" AS "LON",
       "LAT" AS "LAT"
FROM
  (SELECT *
   FROM
     (SELECT DISTINCT(MMSI),
                     ESLORA,
                     CURRENTTIME,
                     SHIPTYPE,
                     NAVSTATUS,
                     DESTINATION,
                     SOG,
                     ZONE,
                     LAT,
                     LON,
                     MINUTE_START
          FROM "DEFAULT"."TABLE_ROUTES_OK"
         ORDER BY MINUTE_START DESC)) AS expr_qry
 WHERE "MINUTE_START" ≥ CAST('2021-03-02 22:02:48' AS TIMESTAMP)
    AND "MINUTE_START" < CAST('2021-03-02 22:32:48' AS TIMESTAMP)
    AND "ZONE" = 'Península'
    AND "SHIPTYPE" = 'Fishing'
    AND "SOG" < 20
 LIMIT 1000;
```

Figura 3-72 Consulta para el *chart* "Posible transbordo de pesca en la mar"

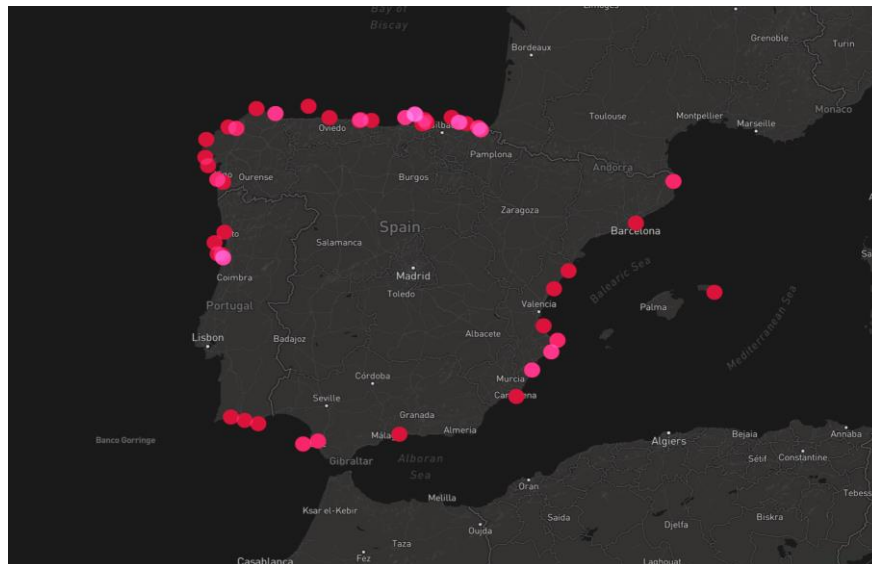


Figura 3-73 Posible transbordo de pesca en la mar

7. Posible *bunkering* en la mar

Otra de las actividades mencionadas en el *chart* anterior es el *bunkering*, abastecimiento de combustible en alta mar. Como en el caso anterior, esta actividad no es ilícita de por sí, pero es necesaria su detección y control.

Las pautas establecidas para la detección de *bunkering* en alta mar son las siguientes: buques mercantes de cualquier tipo que interrumpen su tránsito para permanecer parados o a menos de dos nudos en alta mar y posteriormente retoman la navegación.

Por tanto, se diseñó un mapa empleando *Map Box* como tipo de *chart* y la cola *routes* como fuente de datos que muestra todos los buques mercantes en la zona de la Península que se encuentran navegando a menos de dos nudos. Para ello, se establecieron filtros en los campos *shiptype* (*cargo*), *zone* (Mar Rojo) y *SOG* (menor que 2 nudos). En la Figura 3-74 se muestra la consulta empleada.

Como en el *chart* anterior, para evitar la presentación en el mapa de una misma embarcación en distintos momentos de emisión, se empleó el método descrito en el *chart* Mapa Estrecho de Gibraltar.

```
SELECT "LAT" AS "LAT",
       "LON" AS "LON"
FROM
  (SELECT *
   FROM
     (SELECT DISTINCT(MMSI),
                      ESLORA,
                      CURRENTTIME,
                      SHIPTYPE,
                      NAVSTATUS,
                      DESTINATION,
                      SOG,
                      ZONE,
                      LAT,
                      LON,
                      MINUTE_START
                     FROM "DEFAULT"."TABLE_ROUTES_OK"
                     ORDER BY MINUTE_START DESC) AS expr_qry
   WHERE "MINUTE_START" ≥ CAST('2021-03-02 23:25:27' AS TIMESTAMP)
     AND "MINUTE_START" < CAST('2021-03-02 23:55:27' AS TIMESTAMP)
     AND "SHIPTYPE" = 'Cargo'
     AND "ZONE" = 'Península'
     AND "SOG" < 20
  )
LIMIT 10000;
```

Figura 3-74 Consulta para el *chart* "Posible *bunkering* en la mar"



Figura 3-75 Posible *bunkering* en la mar

8. Derrota del J.S. ELCANO

Este mapa se incluye para demostrar la capacidad del sistema de seguimiento y monitorización de un contacto específico en caso de que fuese necesario. Para ello, se escogió el Buque Escuela "Juan Sebastián Elcano" debido a su ubicación remota, habiendo partido de la isla de Guam en el Pacífico hace pocos días, y por tratarse de un buque emblemático de la Armada Española cuya derrota es seguida por un gran número de familiares y aficionados desde tierra.

Para ello se seleccionó *Map Box* como tipo de *chart* y se impuso un filtro en el campo MMSI con el MMSI del Elcano (224977000). No se estableció ningún filtro temporal, de esta manera se guardan las posiciones de todos los mensajes emitidos por el buque, trazando así su derrota (Figura 3-76). Es importante destacar que este rastreo se puede realizar con cualquier buque, incluso si no se conociese el MMSI (teniendo información suficiente para el filtrado correspondiente), y, por tanto, supone una herramienta de gran utilidad para el COVAM.

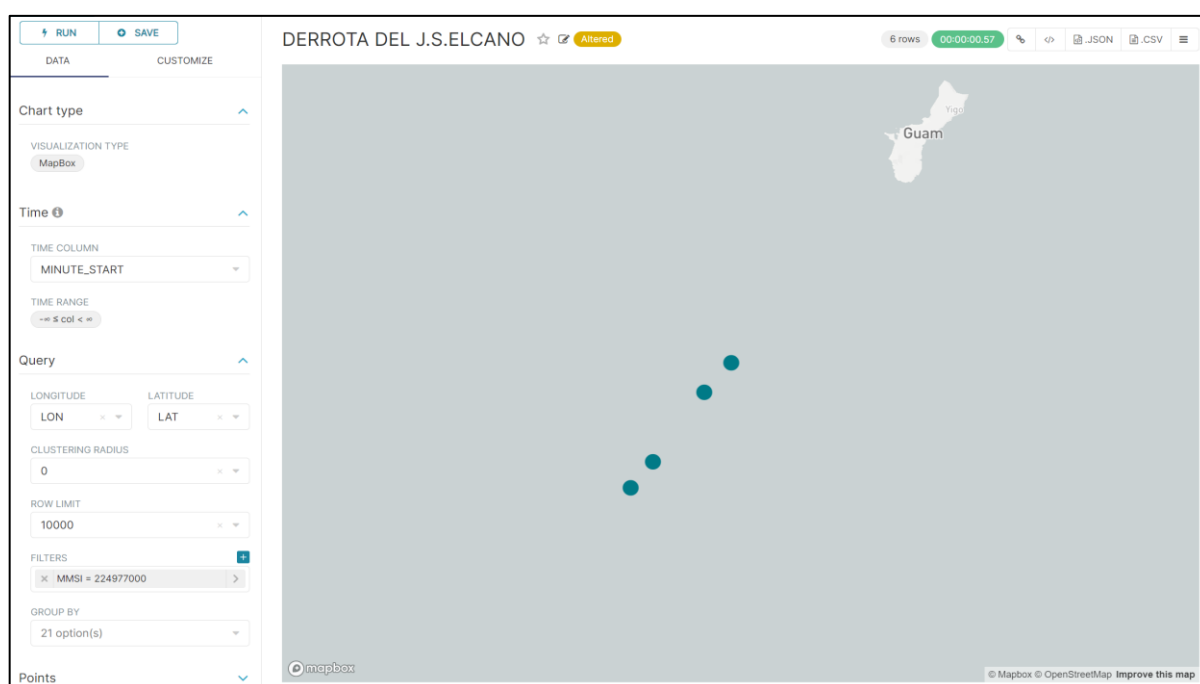


Figura 3-76 Diseño del *chart* "Derrota del J.S. Elcano"

9. Tipos de buques según zonas de interés

Para el diseño de este *chart* (Figura 3-77) se empleó *Sunburst Chart*, estableciendo *shiptype* y *zone* como dimensiones de agrupación, dependiendo *shiptype* jerárquicamente de *zone*. Se filtraron los datos que no pertenecían a ninguna de las zonas de interés nacional (*Remainder*) para únicamente obtener información que pueda resultar de interés. También se impuso un filtro temporal de media hora, es decir, únicamente se muestran los datos recibidos en la última media hora.

En la Figura 3-77 se observa cómo al situar el cursor sobre una sección específica, se muestra una leyenda con el tipo de barco seleccionado y la zona en la que se encuentra, incluyendo el porcentaje de un tipo de barco específico en la zona seleccionada y el porcentaje sobre el total.

Este gráfico se diseñó empleando el método de desplegables y la cola *routes* como fuente de datos.

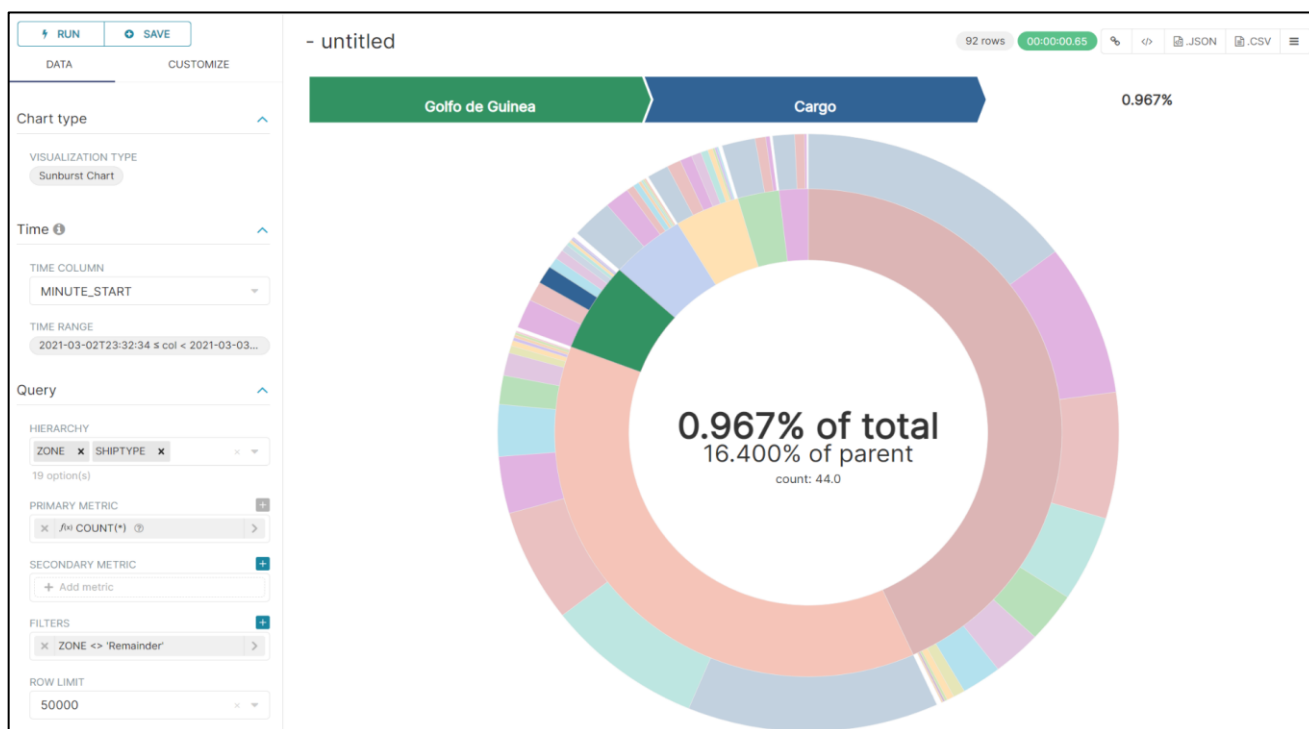


Figura 3-77 Diseño del *chart* "Tipos de buques según zonas de interés"

10. Mapa global

Se diseñó un mapa (Figura 3-79), empleando el *chart Map Box*, que muestra todos los mensajes AIS emitidos en la última hora (se estableció un filtro temporal de una hora como se puede comprobar en la Figura 3-78).

Con el fin de presentar únicamente las últimas posiciones de las embarcaciones transmisoras, se empleó el mismo método de selección del último mensaje AIS recibido por cada MMSI que se utilizó en el resto de los *charts* similares (como se muestran en las respectivas consultas SQL).

Junto a este mapa se agregará un *Filter Box* idéntico al descrito anteriormente para potenciar su utilidad en el análisis de embarcaciones específicas.

Este *chart* se actualiza con los datos provenientes de la cola *routes*.

```
SELECT "LON" AS "LON",  
       "LAT" AS "LAT"  
FROM  
  (SELECT *  
   FROM  
     (SELECT DISTINCT(MMSI),  
                    ESLOA,  
                    CURRENTTIME,  
                    SHIPTYPE,  
                    NAVSTATUS,  
                    DESTINATION,  
                    SOG,  
                    ZONE,  
                    LAT,  
                    LON,  
                    MINUTE_START  
     FROM "DEFAULT"."TABLE_ROUTES_OK"  
     ORDER BY MINUTE_START DESC)) AS expr_qry  
LIMIT 50000;
```

Figura 3-78 Consulta para el chart "Mapa global"



Figura 3-79 Mapa Global

11. Mensajes AIS totales

Para este *chart* se empleó un *Time-series Bar Chart*. Este gráfico muestra el número de embarcaciones totales (eje de ordenadas) en el tiempo (eje de abscisas) sin la aplicación de ningún filtro para así poder obtener un histórico de todos los mensajes AIS emitidos. Al igual que el *chart* anterior, la utilidad de este gráfico se potencia con el empleo del *Filter Box*.

La granularidad temporal establecida fue de minutos. Por tanto, en la Figura 3-80 se puede observar el número totales de mensajes AIS durante los minutos que se lanzó el productor.

Este *chart* se ha creado empleando el método de desplegaables y se actualiza empleando el flujo de datos de la cola *ship_info*.

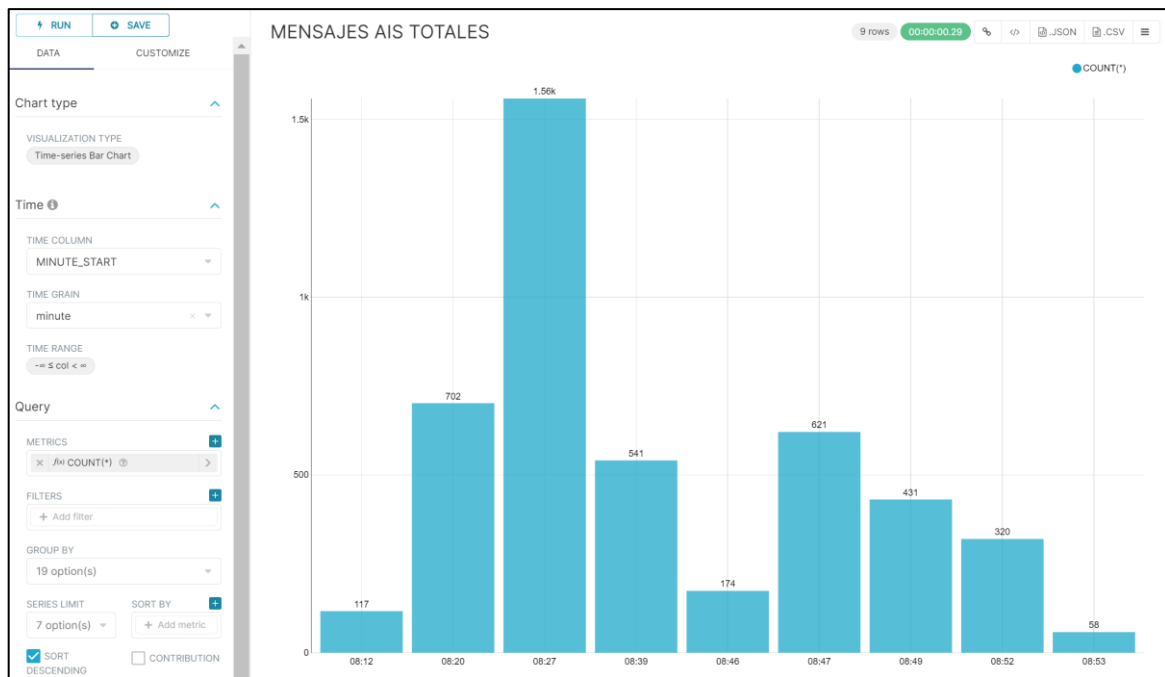


Figura 3-80 Diseño del chart "Mensajes AIS totales"

12. Embarcaciones en zonas de interés

Empleando *Pie Chart* se muestra la cantidad de embarcaciones que se encuentran en las zonas de interés definidas por el campo *zone*. Aparte de los filtros que se pueden realizar con *Filter Box*, a este *chart* se le impuso un filtro específico para eliminar los datos que pertenecen a la zona *Remainder* (al no pertenecer estos mensajes a ninguna de las áreas consideradas de interés). La agrupación de los datos se realizó empleando la dimensión *zone*.

Para la obtención de los mensajes AIS más recientes de cada embarcación se empleó la consulta mostrada en la Figura 3-81, basándose en la misma consulta empleada en el resto de *charts* que exigen esta misma condición (que no se muestre más de un mensaje AIS de cada barco).

Como se muestra en la Figura 3-82, este *chart* permite un elevado nivel de personalización, facilitando la inclusión de leyendas, marcadores que indican la información al situar el cursor encima, selección de colores y especificación de la dimensión de los radios, entre otras.

Este *chart* fue diseñado empleando los mensajes de la cola *routes* como fuente de datos.

```
SELECT "ZONE" AS "ZONE",
       count(*) AS "count"
FROM
  (SELECT *
   FROM
     (SELECT DISTINCT(MMSI),
                      ESLORA,
                      CURRENTTIME,
                      SHIPTYPE,
                      NAVSTATUS,
                      DESTINATION,
                      SOG,
                      ZONE,
                      LAT,
                      LON,
                      MINUTE_START
                     FROM "DEFAULT"."TABLE_ROUTES_OK"
                     ORDER BY MINUTE_START DESC)) AS expr_qry
   WHERE "ZONE" ≠ 'Remainder'
  GROUP BY "ZONE"
 ORDER BY "count" DESC
```

Figura 3-81 Consulta para el chart "Embarcaciones en zonas de interés"

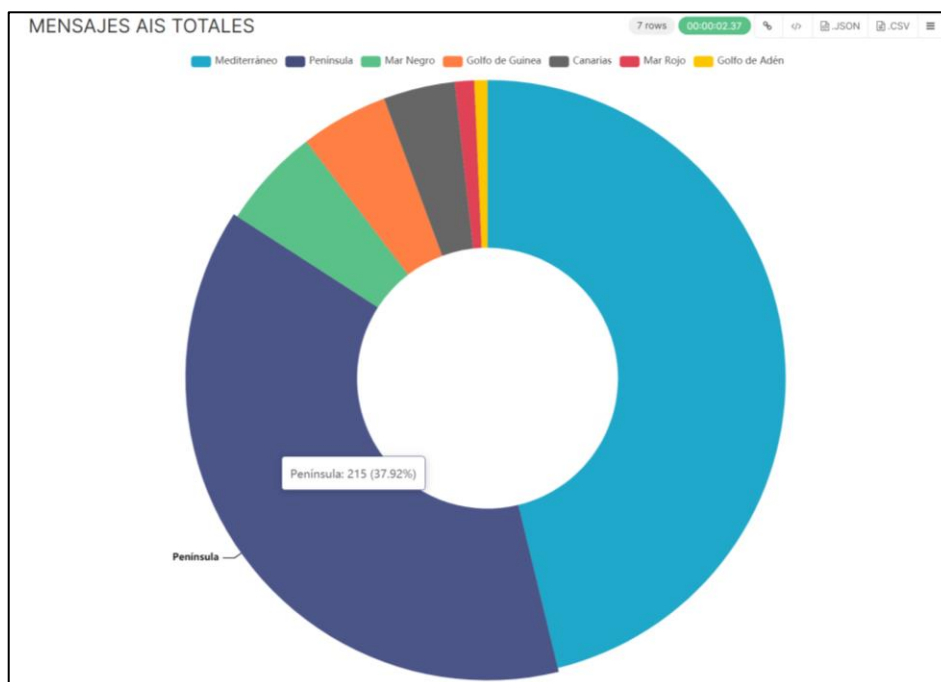


Figura 3-82 Embarcaciones en zonas de interés

13. Tabla mensajes AIS totales

Este *chart* se creó para permitir disponer de información detallada de todas y cada una de las embarcaciones que se encuentran transmitiendo mensajes AIS (Figura 3-83) en todo el mundo. Para sacar el máximo partido de la excesiva información que presenta esta tabla, se le añadió una *search box* que permite la búsqueda de uno o varios barcos específicos mediante la señalización del valor correspondiente de alguno de sus campos. Por ejemplo, si se pretende obtener datos de una embarcación a la que no se consigue localizar en un mapa o RADAR, es posible introducir su MMSI en la *search box* y así obtener información detallada de esta.

Time	MMSI	NAME	CALLSIGN	IMO	ETA	DIMENSIONS	DRAUGHT	SHIPTYPE	DEST	POSTYPE	COUNT(*)
2021-03-03 08:00	244780123	VERWACHTING	PH7466	0	588872	20	0	Pleasure	AMSTERDAM	0	17
2021-03-03 08:00	353992000	GINGA PUMA	3EFS5	9343780	207360	160	70	Tanker	BE ANR-NL RTM	0	13
2021-03-03 08:00	368018310	TRISTEN	WDJ8628	1	201280	21	31	Towing	TAMPA	15	13
2021-03-03 08:00	244700987	KORTENOORD	PD7603	0	849403	70	30	Cargo	ROTTERDAM	0	13
2021-03-03 08:00	236112059	ATLANTIC TWIN	ZDNZ77	9341328	203520	149	85	Tanker	MALMO	1	13
2021-03-03 08:00	224049240	VB ANIBAL	EBWG	9257668	370974	24	43	Tug	CARTAGENA	1	13
2021-03-03 08:00	214182745	TUGRA	ER2745	8420103	186560	0	0	Cargo	TR IZM	0	13
2021-03-03 08:00	273398890	IZUMRUD	UBEP9	9168104	701888	40	60	Fishing	REZ	3	13
2021-03-03 08:00	251856110	BARDUR	TFFP	9861005	1596	27	20	Fishing	OLAFSVIK	0	13
2021-03-03 08:00	235095109	EMMA LOUISE	2FXQ4	202220000	1596	0	0	Fishing	WHELKS CH16/BINFO	15	13
2021-03-03 09:00	992446049	K58		0	1596	0	0	Other		0	13
2021-03-03 09:00	228285000	F/V COELACANTHE	FHJL	822919500	1596	20	0	Fishing		1	13
2021-03-03 09:00	636017942	HEMINGWAY	DSNL6	9778416	189184	300	116	Cargo	ROBERTS BANK	1	13
2021-03-03 08:00	219022256	DANPILOT ALFA	OKXB2	9812030	604318	20	15	Pilot Vessel	GEDSER	1	13
2021-03-03 09:00	316011810	OCEAN K. RUSBY	CFH8966	9345556	528832	30	57	Tug	QUEBEC	1	13
2021-03-03 08:00	224261460	PIRATA DE ONZA	ECMY	0	1596	18	18	Passager	MOANA VIGO MOANA	1	13
2021-03-03 09:00	368029450	EXPORTER	WDJ9792	9852602	508155	28	49	Tug	USFPO<-USFPO	1	13
2021-03-03 09:00	366996360	INTRA REBEL	WDC2682	0	543454	23	29	Spare - Local Vessel	US HOU<-US HOU	0	13
2021-03-03 09:00	316163000	ATLANTIC HEMLOCK	CFD8488	9127588	830144	30	50	Tug	WORKING SAINT JOHN	1	13
2021-03-03 08:00	636017469	TALIMEN	DSLC5	9718026	223488	229	132	Cargo	CN HUH	1	13

Figura 3-83 Diseño del *chart* "Tabla mensajes AIS totales"

Para el diseño de esta tabla se seleccionaron las siguientes dimensiones presentadas en la tabla de datos que almacena la información de la cola *ship_info*: *MMSI*, *name*, *callsign*, *IMO*, *dimensions*, *draught*, *shyptype*, *destination* y *postype*. En la tabla también se puede visualizar el tiempo y el número de mensajes recibidos por cada embarcación.

4 PRESENTACIÓN DE RESULTADOS

En este capítulo se mostrarán los *dashboards* diseñados como producto final del sistema software desarrollado en este trabajo. Mediante el empleo de estos paneles se pretende facilitar ciertas actividades de investigación, análisis y estudio del entorno marítimo proporcionando una imagen general de la situación en tiempo real, así como centralizando en áreas o actividades específicas.

Los *dashboards* generados por Superset permiten la actualización automática de todos los *charts* contenidos señalando el tiempo entre refrescos. De esta manera, se obtienen paneles que muestran constantemente información actualizada.

Para poder mostrar las distintas capacidades del sistema y sus aplicaciones de control para el COVAM, se decidió elaborar tres *dashboards*, cada uno de ellos con un enfoque distinto.

4.1 *Dashboard* de enfoque global

En este primer *dashboard* (Figura 4-1) se puede apreciar la enorme capacidad del sistema software producto de este proyecto a la hora de trabajar con grandes volúmenes de datos. Fue diseñado con el objetivo de obtener una imagen general de la situación marítima en todo el mundo que permitiese mantener una visión clara y en tiempo real de las zonas de interés para el cliente, así como obtener un histórico de los mensajes AIS a nivel global.

Para ello se agregaron a este *dashboard* los siguientes *charts*: Mapa global, *Filter Box*, Mensajes AIS totales, Embarcaciones en zonas de interés y Tabla de mensajes AIS totales. Por tanto, mediante un simple vistazo a este panel se puede conocer la situación marítima mundial en tiempo real (permitiendo filtrar la información según las dimensiones *shiptype*, *navstatus*, *destination* y MMSI), la evolución de esta situación respecto al tiempo (también filtrable con las dimensiones señaladas) y la cantidad de embarcaciones ubicadas en las distintas zonas de interés nacional. También se proporciona información detallada de cada una de las embarcaciones que se encuentran transmitiendo mensajes AIS en todo el mundo, permitiendo la identificación de una en concreto empleando la *search box* de la tabla.

Cabe destacar que el *dashboard* es interactivo, permitiendo acercarse, alejarse y moverse alrededor del mapa, así como obtener mayor información sobre los datos de los gráficos al pasar el cursor por encima de una barra o sección de estos. Además, el filtrado de datos por parte de *Filter Box* en los demás *charts* se realiza de manera instantánea.

Un ejemplo práctico del empleo de este *dashboard* puede ser el estudio de las zonas frecuentadas por pesqueros que se encuentran faenando (mediante el mapa y el gráfico de barras, empleando los filtros *shiptype* y *navstatus*) para obtener información detallada de las áreas de pesca en todo el mundo y de las embarcaciones que las emplean (mediante el uso de la tabla).

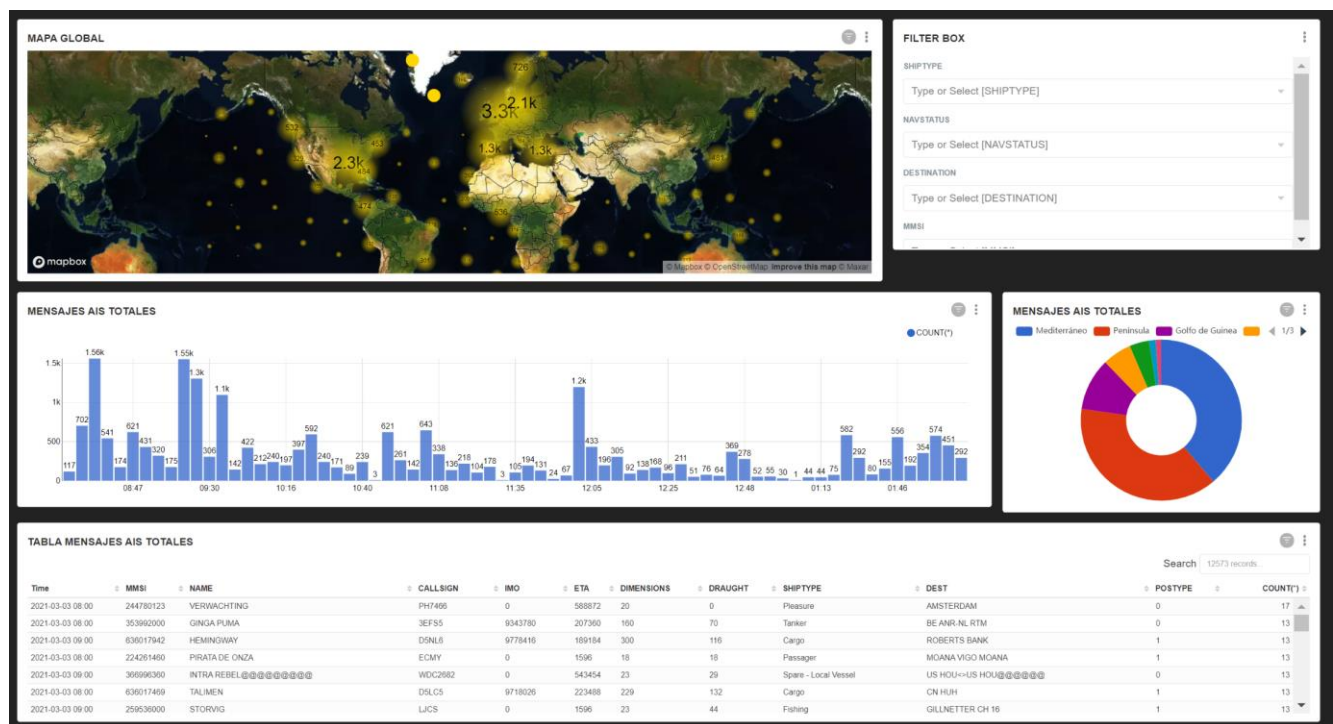


Figura 4-1 Dashboard de enfoque global

4.2 Dashboard de posibles actividades ilícitas o de interés

El segundo *dashboard* diseñado (Figura 4-2) cumple con el objetivo de mostrar las posibles aplicaciones del sistema al mundo militar y, más específicamente, a las funciones del COVAM. Se trata de un *dashboard* de menor tamaño y contenido que el anterior pero que persigue un objetivo mucho más concreto.

Fruto de interacciones con el COVAM, se obtuvieron una serie de escenarios y situaciones que interesa detectar. Para ello se indican ciertos parámetros cinemáticos o de información general de las embarcaciones que permiten detectar comportamientos anómalos en la mar y así facilitar la detección de posibles actividades ilícitas. De entre estas propuestas se seleccionaron dos, explicadas en el apartado 3.4.4, para añadir a este *dashboard*: la detección de posibles transbordos de pesca no autorizado en la mar y la detección de posibles actividades de *bunkering* no autorizado en la mar.

Otro de los objetivos perseguidos con este *dashboard* es el de mostrar la posibilidad de monitorear y la capacidad de seguimiento de una embarcación específica únicamente empleando los mensajes AIS recibidos de esta. Si bien este método de rastreo se ve limitado por el número de transmisiones emitidas por las embarcaciones, es posible perfeccionarlo complementándolo con la información cinemática obtenida de la cola *ship_info* como puede ser la SOG o el ROT.

Por tanto, los *charts* agregados a este *dashboard* fueron los siguientes: Mapa Estrecho de Gibraltar, Posible *bunkering* en la mar, Posible transbordo de pesca en la mar, Derrota del J.S. Elcano y Tipo de buques según zonas de interés. Este último *chart* se añadió para mantener la información sobre las zonas de interés nacional actualizada.

Al igual que en el anterior, todos los elementos de este *dashboard* son interactivos pudiéndose mover alrededor de los mapas y obteniendo etiquetas y leyendas al situar el cursor sobre el *Sunburst Chart*.

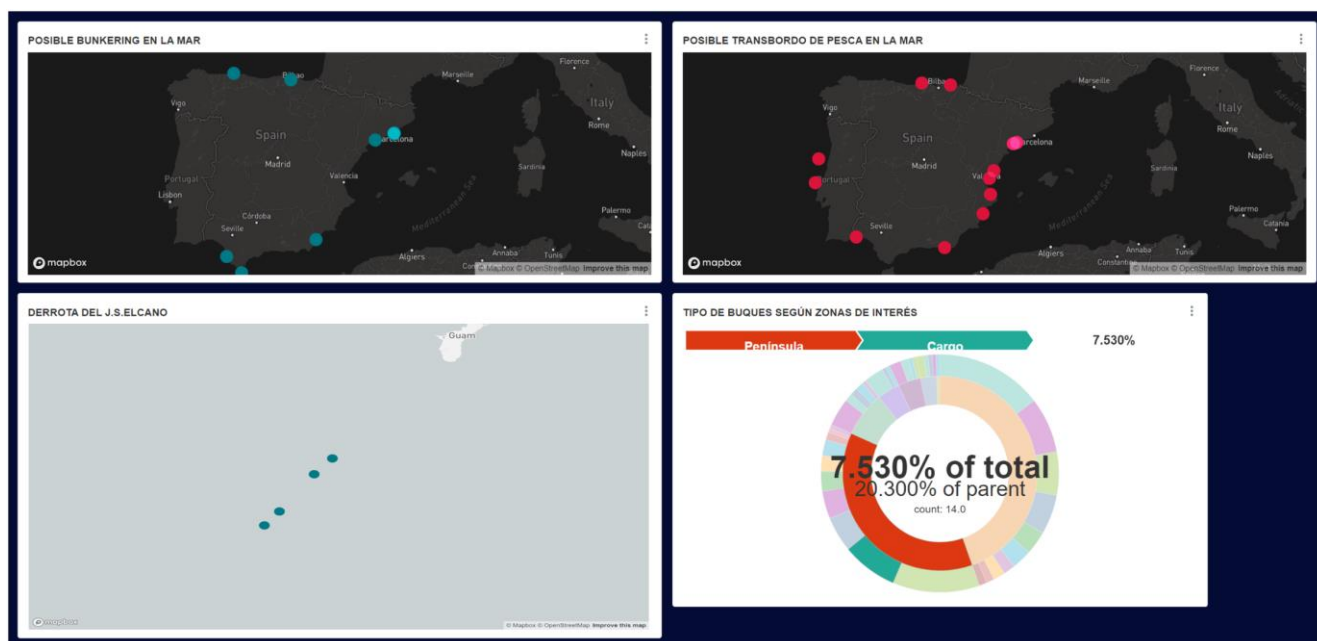


Figura 4-2 Dashboard de posibles actividades ilícitas o de interés

4.3 Dashboard del Estrecho de Gibraltar

El último *dashboard* que se diseñó (Figura 4-3) tiene como objetivo mostrar la capacidad de monitorización y análisis en profundidad de un área determinada por parte del sistema de estudio del presente trabajo.

La zona seleccionada fue el Estrecho de Gibraltar por su gran importancia en el entorno marítimo al tratarse de la entrada de las rutas mercantes al Mediterráneo. El análisis, vigilancia y seguridad del tráfico a través del Estrecho ha sido uno de los cometidos constantes de la Armada Española desde tiempos remotos por la gran ventaja geoestratégica que proporciona.

Con este *dashboard* se pretende obtener información en tiempo real de la situación marítima en esta zona y, para ello, los *charts* que se agregaron fueron: Mapa Estrecho de Gibraltar, Gráfico de tráfico de buques de gran tamaño, *Filter Box*, y el conjunto de *charts* de alerta de embarcaciones a la deriva compuesto de la tabla Embarcaciones a la deriva y el *Big Number A* la deriva.

Por tanto, este *dashboard* permite mantener la situación del Estrecho clara, visualizándola en el mapa, un histórico horario de los buques de gran tamaño (según las consideraciones del RIPAM) que lo han cruzado y un método de señalización de posibles embarcaciones que se encuentren a la deriva en esta zona proporcionando además su MMSI y posición geográfica. Además, es posible la especialización del análisis según el *shiptype*, *navstatus*, *destination* y MMSI.

Este *dashboard*, como los otros dos, permite interactuar con cada uno de los *charts* que lo conforman, pudiendo el cliente acercarse, alejarse y moverse alrededor del mapa, filtrar los datos instantáneamente con el *Filter Box* y mostrando etiquetas con valores específicos en el gráfico de barras.

En la tabla Embarcaciones a la deriva de la Figura 4-3 no se muestra ningún dato debido a que en el momento de captura de pantalla del *dashboard* no había ninguna embarcación a la deriva en la zona del Estrecho de Gibraltar.

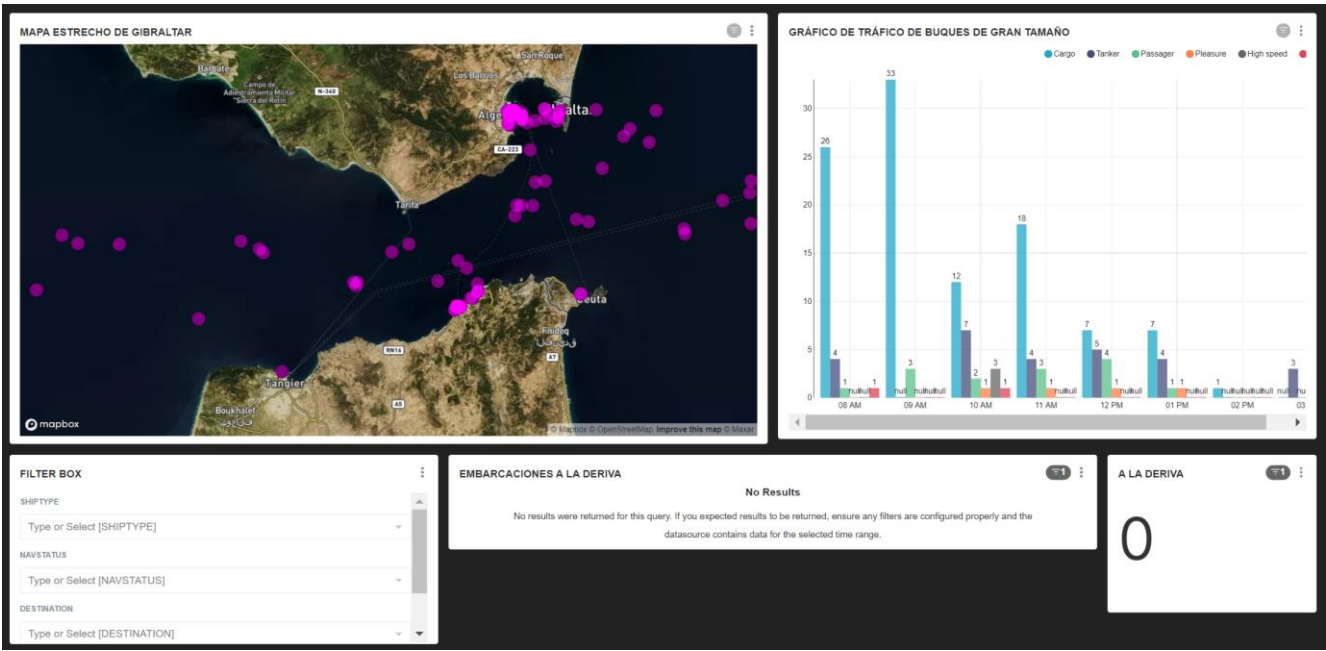


Figura 4-3 *Dashboard* del Estrecho de Gibraltar

5 CONCLUSIONES Y LÍNEAS FUTURAS

La Cuarta Revolución Industrial ha obligado a las Fuerzas Armadas a adaptarse y comenzar la transformación digital de sus unidades y sistemas. Debido al importante recurso estratégico en que se ha convertido el análisis y gestión de datos frente a la frenética evolución de la información, la Armada se ha visto necesitada de herramientas *Big Data* que faciliten el conocimiento de un entorno tan turbulento como es el entorno marítimo. Para ello, con este TFG se ha pretendido contribuir a esta evolución, ya en curso, que conlleva la Armada 4.0.

La principal conclusión extraída, por tanto, es que el conocimiento e implantación de las nuevas tecnologías de análisis de datos, sin duda, proporciona una enorme ventaja a la organización que las emplee. Más concretamente, se ha demostrado que la cantidad de información de la que se dispone sobre la situación marítima a nivel mundial es inimaginable, solo hace falta aprender a obtenerla y gestionarla de manera eficaz y ordenada. Teniendo en cuenta que el presente trabajo únicamente ha incluido información obtenida del sistema AIS, es importante poner en perspectiva la inmensa cantidad de opciones y puertas que puede abrir la extrapolación de este concepto a un centro de gestión de información tan potente como es el COVAM, que recibe información de múltiples fuentes de datos marítimos como pueden ser las sonoboyas, radares de navegación, sonares de búsqueda, organizaciones de tráfico marítimo y los sistemas INMARSAT.

5.1 Revisión de los objetivos

Finalizado el proyecto, se puede concluir que el objetivo principal de desplegar y configurar un sistema software capaz de almacenar volúmenes de datos *Big Data*, mediante la ingesta de flujos de datos provenientes de mensajes AIS en este caso, para permitir un análisis de estos en tiempo real obteniendo así resultados a consultas con mínimas latencias, se ha cumplido satisfactoriamente. También se ha realizado con éxito el diseño de tres *dashboards* que muestran, de manera visual y sencilla, una imagen global del entorno marítimo, así como información detallada de ciertas actividades y zonas de interés, acorde con los indicadores de interés para el COVAM.

Adicionalmente, se ha realizado una revisión detallada de herramientas *Big Data* consideradas de mayor relevancia para este TFG. El motor analítico seleccionado fue Apache Kylin, por las potentes capacidades de consulta y almacenamiento y por proporcionar un método efectivo de análisis de datos en tiempo real. Sin embargo, hubo que generar dos cubos *Real Time OLAP* para hacer posible este análisis, proceso que requirió gran dedicación y tiempo. La herramienta de visualización de datos escogida fue Apache Superset, con esta herramienta se diseñaron los tres *dashboards* mostrados en el Capítulo 4. Esta herramienta fue seleccionada por las diversas funciones y capacidades que permite, así como los múltiples métodos de visualización de los que dispone. Es importante resaltar que la curva de aprendizaje requerida por esta herramienta es mayor de lo que se esperaba.

A título personal, el desarrollo de este trabajo me ha sido de gran utilidad para familiarizarme con las herramientas de tecnología *Big Data* y profundizar más en mis conocimientos, hasta el momento limitados, sobre conceptos como Inteligencia Artificial, *Big Data* y *Data Analytics*, cuya importancia en la Armada solo acaba de empezar.

5.2 Líneas Futuras

Una vez demostrada la posibilidad de implementar un sistema de procesamiento, análisis y presentación de grandes volúmenes de datos en tiempo real en el ámbito militar y marítimo, quedan abiertas una infinidad de oportunidades mediante la explotación de este y de la información que ahora se encuentra disponible.

Como líneas futuras para la mejora y desarrollo del sistema implementado se propone lo siguiente:

- Dedicar tiempo a la creación de tablas Hive de manera que el sistema pueda funcionar en su máximo exponente de desarrollo, empleando tanto los cubos *Real Time OLAP* para *streaming* como los cubos OLAP creados a partir de tablas Hive para las consultas a datos almacenados.
- Estudiar la posibilidad de emplear más fuentes de datos, aparte del sistema AIS, para la obtención de un mayor número de campos de información diferentes, así como para contrastar la información obtenida de los datos AIS (algunos de sus campos no son fiables por haberse rellenado mal o de forma incompleta los datos del buque).
- Estudiar más en profundidad las propiedades configurables de Apache Kylin para mejorar la arquitectura de los cubos incrementando su eficiencia y reduciendo las latencias de consulta.
- Investigar la posibilidad de emplear Apache Kylin sobre un almacén de datos que permita la realización de *geo-queries* directamente sin necesidad de realizar un procesamiento previo. Una opción que, en principio, parece factible, sería utilizar el TFG realizado en paralelo por el AF Javier Vila Aparicio “Plataforma *Big Data* para almacenamiento de flujos de información marítima” para desarrollar un sistema que emplee Elasticsearch como *storage engine* de Apache Kylin.

6 BIBLIOGRAFÍA

- [1] International Maritime Organization, «www.imo.org,» [En línea]. Available: <https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx>. [Último acceso: 24 Enero 2021].
- [2] Global Fishing Watch, «Global Fishing Watch ¿Qué es el AIS?,» [En línea]. Available: <https://globalfishingwatch.org/es/faqs/que-es-ais/>. [Último acceso: 24 Enero 2021].
- [3] «VesselFinder,» [En línea]. Available: <https://www.vesselfinder.com/es>. [Último acceso: 2 Marzo 2021].
- [4] Armada Española, «Entorno Colaborativo Marítimo de la Armada,» [En línea]. Available: <https://encomar.covam.es/bienvenido>. [Último acceso: 2 Marzo 2021].
- [5] C. Sanchez, E. Marmol y I. Eguía, «Puertos de España,» *Transporte XXI. El periódico del transporte y la logística en España*, 1 Abril 2020.
- [6] B. J. Copeland, «Artificial intelligence,» Britannica, [En línea]. Available: <https://www.britannica.com/technology/artificial-intelligence>. [Último acceso: 21 Enero 2021].
- [7] L. A. Hidalgo, *Inteligencia Artificial y Sistemas Expertos*, Córdoba: Servicio de Publicaciones de la Universidad de Córdoba, 1996.
- [8] E. Rich, *Artificial Intelligence*, New York: McGraw-Hill, 1984.
- [9] SAS Institute, «Inteligencia Artificial ¿Qué es y por qué es importante?,» [En línea]. Available: https://www.sas.com/es_es/insights/analytics/what-is-artificial-intelligence.html. [Último acceso: 13 Enero 2021].
- [10] J. M. Gutiérrez, «Introducción a las Redes Neuronales,» Universidad de Cantabria.
- [11] ATRIA Innovations, «Qué son las redes neuronales y sus funciones,» 22 Octubre 2019. [En línea]. Available: <https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/>. [Último acceso: 23 Enero 2021].
- [12] SIEMENS, «La Inteligencia Artificial en la industria: producción inteligente,» [En línea]. Available: <https://new.siemens.com/global/es/empresa/la-revista/industria/inteligencia-artificial-en-la-industria.html>. [Último acceso: 15 Enero 2021].

- [13] V. L. Cuquerella Gamboa, «Armada Española,» [En línea]. Available: <https://armada.defensa.gob.es/archivo/rgm/2018/08/rgm080918cap07.pdf>. [Último acceso: 15 Enero 2021].
- [14] Iberdrola, «Cuarta Revolución Industrial,» [En línea]. Available: <https://www.iberdrola.com/innovacion/cuarta-revolucion-industrial>. [Último acceso: 19 Enero 2021].
- [15] A. P. Esañon, «Los cambios que trae la industria 4.0,» 29 Noviembre 2019. [En línea]. Available: <https://blog.wearedrew.co/los-cambios-que-trae-la-industria-4.0>. [Último acceso: 20 Enero 2021].
- [16] M. Perod, «La Inteligencia Artificial, más presente de lo que imaginamos,» [En línea]. Available: <https://www.muyinteresante.es/tecnologia/inteligencia-artificial/articulo/la-inteligencia-artificial-mas-presente-de-lo-que-imaginamos-661521457387#:~:text=Se%20conoce%20como%20deepfakeporn%2C%20y,Delegvine%2C%20Taylor%20Switf%20entre%20otras..> [Último acceso: 30 Enero 2021].
- [17] G. Seisdedos Domínguez, «Smart Cities. La transformación digital de las ciudades,» Centro de Innovación del Sector Público de PwC e IE Buisness School.
- [18] V. Mayer-Schonberger y K. Cukier, Big Data, La revolución de los datos masivos, Turner Publicaciones S.L., 2013.
- [19] Instituto de Ingenieria del Conocimiento UAM, «Las 7 V del Big data: Características más importantes,» [En línea]. Available: <https://www.iic.uam.es/innovacion/big-data-caracteristicas-mas-importantes-7-v/>. [Último acceso: 17 Enero 2021].
- [20] M. Navarro, «Almacenamiento: cuando la nube lo cambió todo,» 7 Noviembre 2019. [En línea]. Available: <https://revistabyte.es/tema-de-portada-byte-ti/almacenamiento-la-nube/>. [Último acceso: 29 Enero 2021].
- [21] PowerData, «Big Data: ¿En qué consiste? Su importancia, desafíos y gobernabilidad,» [En línea]. Available: <https://www.powerdata.es/big-data>. [Último acceso: 19 Enero 2021].
- [22] Instituto de Ingenieria del Conocimiento UAM, «Infografía Big Data: las 7 V,» [En línea]. Available: <https://www.iic.uam.es/innovacion/big-data-infografia-7-v/>. [Último acceso: 18 Enero 2021].
- [23] Master's in Data Science, «What is Data Analytics?,» [En línea]. Available: <https://www.mastersindatascience.org/resources/what-is-data-analytics/>. [Último acceso: 20 Enero 2021].
- [24] «Zeus,» [En línea]. Available: <https://zeus.vision/>. [Último acceso: 20 Enero 2021].
- [25] INDRA, «Indra y la Armada utilizan la inteligencia artificial para mejorar el mantenimiento de buques de nueva generación,» 17 Agosto 2020. [En línea]. Available: <https://www.indracompany.com/es/noticia/indra-armada-utilizan-inteligencia-artificial-mejorar-mantenimiento-buques-generacion>. [Último acceso: 21 Enero 2021].
- [26] infodefensa.com, «Indra y la Armada prueban un cerebro artificial que mejora el mantenimiento de los buques,» 17 Septiembre 2020. [En línea]. Available: <https://www.infodefensa.com/es/2020/09/17/noticia-indra-armada-prueban-exito-cerebro-artificial-mejora-mantenimiento-buques.html>. [Último acceso: 22 Enero 2021].

- [27] J. A. Carrillo Ruiz, «Big Data en los entornos de Defensa y Seguridad,» CESEDEN, 2013. [En línea]. Available: http://www.ieee.es/Galerias/fichero/docs_investig/DIEEEINV03-2013_Big_Data_Entornos_DefensaSeguridad_CarrilloRuiz.pdf. [Último acceso: 21 Enero 2021].
- [28] J. Díaz García, «Selección e implantación de una herramienta de diseño de dashboards de aplicación al COVAM,» 2020. [En línea]. Available: <http://calderon.cud.uvigo.es/handle/123456789/348>. [Último acceso: 21 Enero 2021].
- [29] R. González Guitart, «Plataforma Big Data para el análisis de flujos de información marítima,» 2020. [En línea]. Available: <http://calderon.cud.uvigo.es/handle/123456789/398>. [Último acceso: 21 Enero 2021].
- [30] «Grafana,» [En línea]. Available: <https://grafana.com/>. [Último acceso: 21 Enero 2021].
- [31] Y. Dendane, F. Petrillo, H. Mcheick y S. B. Ali, «Quality model for evaluating and choosing a stream,» 25 Enero 2019. [En línea]. Available: https://www.researchgate.net/publication/330700749_A_quality_model_for_evaluating_and_choosing_a_stream_processing_framework_architecture. [Último acceso: 22 Enero 2021].
- [32] «kafka.apache.org,» Apache Kafka, [En línea]. Available: <https://kafka.apache.org/>. [Último acceso: 4 Marzo 2021].
- [33] Galaktika-soft, «OLAP server comparison,» [En línea]. Available: <https://galaktika-soft.com/blog/olap-servers-comparison.html>. [Último acceso: 29 Enero 2021].
- [34] tic.PORTAL, «Open source (código abierto),» Septiembre 2015. [En línea]. Available: <https://www.ticportal.es/glosario-tic/open-source-codigo-abierto#:~:text=La%20ventaja%20principal%20del%20software,variado%20de%20multitud%20de%20herramientas>. [Último acceso: 25 Enero 2021].
- [35] «American National Standard Institute,» [En línea]. Available: <https://www.ansi.org/>. [Último acceso: 29 Enero 2021].
- [36] «druid.apache.org,» Apache Druid, [En línea]. Available: <https://druid.apache.org/>. [Último acceso: 27 Enero 2021].
- [37] «Metamarkets,» [En línea]. Available: <https://metamarkets.com/>. [Último acceso: 25 1 2021].
- [38] «Aprende Big Data,» 2 Enero 2021. [En línea]. Available: <https://aprenderbigdata.com/apache-druid/>. [Último acceso: 25 Enero 2021].
- [39] Amazon Web Services, «Almacenamiento en columnas,» [En línea]. Available: https://docs.aws.amazon.com/es_es/redshift/latest/dg/c_columnar_storage_disk_mem_mgmt.html. [Último acceso: 25 Enero 2021].
- [40] «zookeeper.apache.org,» Apache Zookeeper, [En línea]. Available: <https://zookeeper.apache.org/>. [Último acceso: 28 Enero 2021].
- [41] «calcite.apache.org,» Apache Calcite, [En línea]. Available: https://calcite.apache.org/docs/druid_adapter.html. [Último acceso: 27 Enero 2021].
- [42] «kylin.apache.org,» Apache Kylin, [En línea]. Available: kylin.apache.org. [Último acceso: 26 Enero 2021].

- [43] «hadoop.apache.org,» Apache Hadoop, [En línea]. Available: <https://hadoop.apache.org/>. [Último acceso: 3 Marzo 2021].
- [44] Tecnología+Informática, «Sistemas OLAP: Análisis empresarial. Cubos y tipos de OLAP,» 17 Julio 2020. [En línea]. Available: <https://www.tecnologia-informatica.com/sistemas-olap-cubos/>. [Último acceso: 25 Enero 2021].
- [45] «Cubo OLAP Cinema,» [En línea]. Available: <https://sites.google.com/site/cuboolapcinema/project-definition>. [Último acceso: 25 Enero 2021].
- [46] «hive.apache.org,» Apache Hive, [En línea]. Available: <https://hive.apache.org/>. [Último acceso: 3 Marzo 2021].
- [47] D. Pinal, «Big Data – Buzz Words: What is MapReduce – Day 7 of 21,» 9 Octubre 2013. [En línea]. Available: <https://blog.sqlauthority.com/2013/10/09/big-data-buzz-words-what-is-mapreduce-day-7-of-21/>. [Último acceso: 28 Enero 2021].
- [48] «spark.apache.org,» Apache Spark, [En línea]. Available: <https://spark.apache.org/>. [Último acceso: 20 Febrero 2021].
- [49] «hbase.apache.org,» Apache HBase, [En línea]. Available: <https://hbase.apache.org/>. [Último acceso: 20 Febrero 2021].
- [50] Microsoft, «Descripción de un esquema de estrella e importancia para Power BI,» 9 9 2019. [En línea]. Available: <https://docs.microsoft.com/es-es/power-bi/guidance/star-schema>. [Último acceso: 1 Febrero 2021].
- [51] Inteligencia de Negocios, «Modelo Esquema en Estrella,» 27 Agosto 2013. [En línea]. Available: <http://intenegro.blogspot.com/2013/08/modelo-esquema-en-estrella.html>. [Último acceso: 1 Febrero 2021].
- [52] «pinot.apache.org,» Apache Pinot, [En línea]. Available: <https://pinot.apache.org/>. [Último acceso: 27 Enero 2021].
- [53] R. Leventov, «Comparación de los sistemas OLAP de código abierto para Big Data: ClickHouse, Druid y Pinot,» 1 Febrero 2018. [En línea]. Available: <https://leventov.medium.com/comparison-of-the-open-source-olap-systems-for-big-data-clickhouse-druid-and-pinot-8e042a5ed1c7>. [Último acceso: 28 Enero 2021].
- [54] «helix.apache.org,» Apache Helix, [En línea]. Available: <https://helix.apache.org/>. [Último acceso: 27 Enero 2021].
- [55] «clickhouse.tech,» ClickHouse, [En línea]. Available: <https://clickhouse.tech/>. [Último acceso: 27 Enero 2021].
- [56] «yandex.com,» Yandex, [En línea]. Available: <https://yandex.com/>. [Último acceso: 25 Enero 2021].
- [57] J. Xu, «MarkLogic,» [En línea]. Available: <https://markmail.org/message/mf6gfdwfwqwtbtv6#query:+page:1+mid:sp7ek7x5pawjlx6+sate:results>. [Último acceso: 30 Enero 2021].
- [58] ProgrammerSought, «Comparación del motor OLAP Kylin y Clickhouse,» [En línea]. Available: <https://www.programmersought.com/article/59366417257/>. [Último acceso: 30 Enero 2021].

- [59] «Cyfe,» [En línea]. Available: <https://www.cyfe.com/>. [Último acceso: 1 Febrero 2021].
- [60] dr.Metrics, «Mejores herramientas dashboards,» Metriplica, 1 Octubre 2017. [En línea]. Available: <https://www.doctormetrics.com/mejores-herramientas-para-dashboards/>. [Último acceso: 1 Febrero 2021].
- [61] 40defiebre, «Mejores herramientas dashboard de analítica web,» [En línea]. Available: <https://www.40defiebre.com/mejores-herramientas-dashboard-analitica-web>. [Último acceso: 1 Febrero 2021].
- [62] «Klipfolio,» [En línea]. Available: <https://www.klipfolio.com/>. [Último acceso: 1 Febrero 2021].
- [63] «Geckoboard,» [En línea]. Available: <https://www.geckoboard.com/>. [Último acceso: 1 Febrero 2021].
- [64] Apache Software Foundation, «Apache Superset,» [En línea]. Available: <https://superset.apache.org/>. [Último acceso: 8 Marzo 2021].
- [65] «Demos de Big Data y Machine Learning de Stratebi,» Stratebi, [En línea]. Available: <https://bigdata.stratebi.com>. [Último acceso: 1 Febrero 2021].
- [66] «Tableau,» [En línea]. Available: <https://www.tableau.com/es-es>. [Último acceso: 1 Febrero 2021].
- [67] «InetSoft,» [En línea]. Available: <https://www.inetsoft.com/>. [Último acceso: 1 Febrero 2021].
- [68] «Apache Zeppelin,» [En línea]. Available: <https://zeppelin.apache.org/>. [Último acceso: 2 Febrero 2021].
- [69] «Clicdata,» [En línea]. Available: <https://www.clicdata.com/es/>. [Último acceso: 2 Febrero 2021].
- [70] «Chartio,» [En línea]. Available: <https://chartio.com/>. [Último acceso: 2 Febrero 2021].
- [71] Action Metrics, «Dashboard tools review,» [En línea]. Available: <https://www.youtube.com/watch?v=u9siGHxcvKI..> [Último acceso: 2 Febrero 2021].
- [72] AISHub, «AIS data API,» [En línea]. Available: <https://www.aishub.net/api>. [Último acceso: 15 Febrero 2021].
- [73] «AIS Data Set,» [En línea]. Available: <https://api.vtexplorer.com/docs/ref-aistypes.html>. [Último acceso: 18 Febrero 2021].
- [74] Google, «Google Maps,» [En línea]. Available: <https://www.google.es/maps/preview>. [Último acceso: 19 Febrero 2021].
- [75] Marine Traffic, «What is the significance of the AIS Navigational Status Values?,» [En línea]. Available: <https://help.marinetraffic.com/hc/en-us/articles/203990998-What-is-the-significance-of-the-AIS-Navigational-Status-Values->. [Último acceso: 19 Febrero 2021].
- [76] Apache Kylin, «Deep dive into Kylin's Real-time OLAP,» 1 7 2019. [En línea]. Available: <http://kylin.apache.org/blog/2019/07/01/deep-dive-real-time-olap/>. [Último acceso: 26 Febrero 2021].

- [77] Apache Kylin, «Installation Guide,» [En línea]. Available: <http://kylin.apache.org/docs30/install/index.html>. [Último acceso: 26 Febrero 2021].
- [78] Apache Software Foundation Distributed Directory, «Index of /hadoop/common/hadoop-3.2.2,» 13 1 2021. [En línea]. Available: <https://www-eu.apache.org/dist/hadoop/common/hadoop-3.2.2/hadoop-3.2.2.tar.gz>. [Último acceso: 10 Febrero 2021].
- [79] Apache Software Foundation Distributed Directory, «Index of /apache/hive/hive-3.1.2,» 7 Marzo 2020. [En línea]. Available: <https://ftp.cixug.es/apache/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz>. [Último acceso: 11 Febrero 2021].
- [80] Apache Software Foundation Distributed Directory, «HBase Releases,» [En línea]. Available: <https://ftp.cixug.es/apache/hbase/2.4.0/hbase-2.4.0-bin.tar.gz>. [Último acceso: 12 Febrero 2021].
- [81] Apache Software Foundation Distributed Directory, «Index of /dist/kafka,» 19 Diciembre 2019. [En línea]. Available: https://archive.apache.org/dist/kafka/2.5.0/kafka_2.12-2.5.0.tgz. [Último acceso: 16 Febrero 2021].
- [82] Apache Software Foundation Distributed Directory, «Index of /dist/kylin,» 19 Diciembre 2019. [En línea]. Available: <https://archive.apache.org/dist/kylin/apache-kylin-3.0.0/apache-kylin-3.0.0-bin-hadoop3.tar.gz>. [Último acceso: 16 Febrero 2021].
- [83] Apache Software Foundation, «Installing Superset from Scratch,» [En línea]. Available: <https://superset.apache.org/docs/installation/installing-superset-from-scratch>. [Último acceso: 20 Febrero 2021].
- [84] «Mapbox,» [En línea]. Available: <https://account.mapbox.com/>. [Último acceso: 20 Febrero 2021].
- [85] Instituto Hidrográfico de la Marina, Reglamento Internacional para Prevenir los Abordajes en la Mar, Instituto Hidrográfico de la Marina, 1972.