



## **Servicios cognitivos en entornos FIWARE**

**Autor: Dario Alandes Codina**

**Tutor: Dr. Carlos Enrique Palau Salvador**

Trabajo Fin de Máster presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Máster en Ingeniería Telecomunicación

Curso 2020-21

Valencia, 1 de diciembre de 2020



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

**TELECOM** ESCUELA  
TÉCNICA **VLC** SUPERIOR  
DE INGENIERÍA DE  
TELECOMUNICACIÓN

*A mi familia por su gran apoyo*

*Al equipo de los proyectos RETOS VALKNUT y ITEA3 COSIBAS, y en especial a Carlos Palau  
por darme la gran oportunidad de integrarme en el proyecto*



## Resumen

En este proyecto se ha realizado una plataforma para poder compartir información, concretamente datos obtenidos a partir de varias antenas dispuestas a lo largo la costa de la comunidad valenciana.

Para ello haremos uso de unos receptores AIS de bajo coste, con los cuales cubrir todos los puntos de interés a lo largo de la costa de la comunidad valencia sin llegar a tener un coste excesivamente elevado. Para el desarrollo de la plataforma se usará FIWARE.

El objetivo final de este trabajo es proporcionar una cobertura AIS de toda la costa valenciana, así como proporcionar una plataforma donde poder consultar cualquier tipo de información acerca de los barcos que hay en la misma.

Además, gracias a los servicios cognitivos ser capaces de modelar diferentes escenarios en los cuales, mediante el uso de inteligencia artificial, se analizan los datos procedentes de AIS en tiempo real junto con datos de otras fuentes para llegar a soluciones reales.



## Resum

En aquest projecte s'ha realitzat una plataforma per poder compartir informació, concretament dades obtingudes a partir de diverses antenes disposades al llarg de la costa de la Comunitat Valenciana.

Per a això farem ús d'uns receptors AIS de baix cost, amb els quals cobrir tots els punts d'interès al llarg de la costa de la comunitat valència sense arribar a tenir un cost excessivament elevat. Per al desenvolupament de la plataforma es farà servir FIWARE.

L'objectiu final d'aquest treball és proporcionar una cobertura AIS de tota la costa valenciana, així com proporcionar una plataforma on poder consultar qualsevol tipus d'informació sobre els vaixells que hi ha a la mateixa.

A més, gràcies als serveis cognitius ser capaços de modelar diferents escenaris en els quals, mitjançant l'ús d'intel·ligència artificial, s'analitzen les dades procedents d'AIS en temps real juntament amb dades d'altres fonts per arribar a solucions reals.



## **Abstract**

In this project, a platform has been created to share information, specifically data obtained from various antennas arranged along the coast of the Valencian community.

For this we will use low-cost AIS receivers, with which to cover all points of interest along the coast of the Valencian community without being excessively high cost. FIWARE will be used for the development of the platform.

The final objective of this work is to provide an AIS coverage of the entire Valencian coast, as well as to provide a platform where you can consult any type of information about the boats on it.

In addition, thanks to cognitive services, being able to model different scenarios in which, through the use of artificial intelligence, data from AIS are analyzed in real time along with data from other sources to arrive at real solutions.



## Índice

Capítulo 1. Introducción y objetivos.....	3
1.1 Introducción .....	3
1.2 Objetivos .....	4
Capítulo 2. Estado del arte .....	5
2.1 Servicios Cognitivos .....	5
2.1.1 Microsoft Azure .....	5
2.1.2 Amazon AWS .....	5
2.2 Estándar AIS .....	6
2.2.1 Funcionamiento.....	7
2.3 Receptor AIS de bajo coste .....	8
2.3.1 Introducción .....	8
2.3.2 Diseño hardware.....	8
2.3.3 Diseño software.....	9
2.4 Data market .....	10
2.4.1 MarineTraffic .....	10
2.4.2 Vesseltracker .....	11
2.5 FIWARE .....	12
2.5.1 FIWARE Orion Context Broker.....	13
2.6 Mongo DB.....	14
2.7 Docker .....	14
Capítulo 3. Desarrollo .....	16
3.1 Modificación del receptor AIS .....	16
3.1.1 Hardware .....	16
3.1.2 Software .....	16
3.2 Relación de confianza con AISHUB.....	18
3.3 Plan de cobertura de la Comunidad Valenciana.....	19
Capítulo 4. Implementación.....	21
4.1 Arquitectura.....	21
4.2 Servidor y base de datos.....	22
4.3 Agentes.....	22
4.3.1 Agente.py .....	22
4.4 CKAN .....	25
4.4.1 Instalación .....	25
4.5 Escenarios .....	27



4.5.1	Mejora de la predicción de ETA .....	27
4.5.2	Optimización de operaciones logísticas .....	27
4.5.3	Cuidado medioambiental.....	28
Capítulo 5.	Conclusiones y futuros desarrollos.....	29
5.1	Conclusiones .....	29
5.2	Futuros desarrollos .....	30
Capítulo 6.	Bibliografía.....	31



## Capítulo 1. Introducción y objetivos

### 1.1 Introducción

Actualmente, estamos viendo como la información se convierte cada vez más en uno de los bienes más preciados, las telecomunicaciones, forman un papel fundamental, ya que se encargan de la transmisión de esta, además de hacerlo de manera segura, fiable y autenticada. A su misma vez, las aplicaciones que se encargan de la transmisión y recepción están viendo como su número es cada vez mayor.

Si nos metemos en el marco de comunicaciones marítimas, las transmisiones que se producen aquí permiten conocer de manera exacta información de los navíos del tipo: ubicación, velocidad, destino, tamaño de este, etc.

Si analizamos de manera exhaustiva mediante el uso de servicios cognitivos esta información seremos capaces de llegar a saber el tiempo exacto que tardará y a la velocidad a la que debe ir para que cuando llegue al puerto no tenga que esperar a tener una dársena libre, permitiendo así a las empresas portuarias ahorrar mucho tiempo y dinero.

El protocolo AIS, es un protocolo de ayuda a la navegación que, a diferencia del radar, el AIS además muestra por pantalla la velocidad, posición y rumbo de los barcos de alrededor, además tiene bastante más alcance que el radar, pero, para poder identificar a un barco, este debe tener instalado y encendido el AIS.

Además, se obtendrá en un mismo servidor toda la información procedente de todos los receptores AIS desplegados por toda la costa para así poder implementar una plataforma donde poder compartir toda esta información para que todo el que quiera pueda acceder a ella sin necesidad de montarse su propio receptor AIS, además de proporcionar una plataforma donde, también, quien quiera y disponga de un receptor AIS es bienvenido a compartir la información que recibe.

La plataforma se desarrollará utilizando el ecosistema FIWARE, eso nos permitirá entre otras cosas asegurar la interoperabilidad entre diferentes receptores y fuentes de información, ya que una de las principales características de FIWARE es que gracias a sus agentes IoT transforma los protocolos nativos de cada dispositivo en otro soportado por FIWARE.

Finalmente, se combinará la plataforma FIWARE con servicios cognitivos para así poder reconocer patrones, los cuales tendrán la capacidad de procesamiento de los datos anticipando así los problemas nuevos que pueden aparecer al mismo tiempo que modelan soluciones posibles.



## 1.2 Objetivos

El objetivo de este trabajo es la creación de un data market con los datos recibidos por un receptor AIS de bajo coste, además de la cobertura de la zona costera la comunidad valenciana con estos receptores de bajo coste. Eso se llevará a cabo buscando los siguientes objetivos principales:

- Investigar sobre los servicios cognitivos y sus posibilidades
- Partiendo de desarrollos anteriores se modificará un receptor AIS de bajo coste, adaptándolo a las nuevas necesidades que surjan.
- Planificar la cobertura de la zona costera de la comunidad valenciana usando receptores AIS de bajo coste.
- Establecer una relación de confianza con otro data market, permitiendo así validar los datos recibidos por los receptores.
- Trabajar con el sistema de virtualización a base de contenedores Docker
- Instalar y configurar la plataforma FIWARE en un entorno Docker
- Estudiar el lenguaje de programación y el entorno de desarrollo Python
- Estudiar la base de datos NoSQL MongoDB implementándola como base principal del proyecto
- Recoger todos los datos de los receptores AIS que cubren la costa de la comunidad valenciana en un servidor.
- Modelar sistemas cognitivos en un entorno con FIWARE
- Implementar un data market donde poner publicar los datos recibidos por los receptores desplegados por la costa de la Comunidad Valenciana



## Capítulo 2. Estado del arte

### 2.1 Servicios Cognitivos

Los servicios cognitivos hacen referencia a aquellos servicios inteligentes que permiten que aplicaciones, wereables o incluso robots tengan capacidades “humanas”. Para ello es necesario implementar sistemas de autoaprendizaje y Machine Learning, los cuales se van a nutrir de datos almacenados. A través de algoritmos y minado de datos se obtienen patrones reconocibles con los cuales estos servicios cognitivos sean capaces de anticipar problemas y modelar las soluciones posibles.

En el entorno de las navegaciones marítimas el análisis en tiempo real a través de algoritmos de inteligencia artificial y herramientas de autoaprendizaje permitirá mejorar la eficiencia global de la cadena de transporte marítima, apoyando a los actores involucrados (puertos, barcos, compañías de transporte, etc.) a tomar decisiones mucho más eficientes y colaborativas. Un caso sería ayudara las autoridades portuarias a planificar con mucha más precisión las llegadas y salidas de las embarcaciones haciendo un uso óptimo de los recursos.

Actualmente las principales plataformas líderes en inteligencia artificial que permiten implementar y entrenar modelos rápidamente son: IBM Watson, Google, Microsoft Azure, Amazon AWS, Oracle.

#### 2.1.1 *Microsoft Azure*

Microsoft Azure ofrece un conjunto de servicios de autoaprendizaje basados en Python, incluye un conjunto de servicios de inteligencia artificial entrenados previamente a los cuales se puede acceder vía API para crear aplicaciones con funciones cognitivas. También incluye herramientas de desarrollo para crear modelos personalizados de autoaprendizaje.

#### 2.1.2 *Amazon AWS*

Amazon AWS es una plataforma de computación en la nube ofrecida por Amazon, proporciona un conjunto de herramientas para la computación distribuida, además de un entorno de desarrollo para poder crear e implementar modelos de autoaprendizaje personalizados. Amazon también ofrece servicios previamente capacitados para diferentes campos, herramientas para diseñar, entrenar e implementar modelos de autoaprendizaje compatibles con los lenguajes más populares.



## 2.2 Estándar AIS

AIS es uno de los avances tecnológicos más significativos en el ámbito de la seguridad marítima de la historia moderna. La Organización Marítima Internacional adoptó en 2004 el uso mundial de esta tecnología para los buques. Una gran cantidad de países ampliaron el requisito de transporte de AIS para incluir dentro de este uso a todos los buques comerciales de un tamaño mayor a 65 pies, unos 20 metros, y a la gran mayoría de remolcadores.

La regla V / 19.2.4 del Convenio de la IMO para la seguridad de la vida humana en el mar (SOLAS) exige que todos los buques de 300 de toneladas bruto o más que realicen viajes internacionales y todos los buques de pasaje, independientemente de su tamaño, lleven AIS a bordo.

AIS transmite el mismo tipo de datos que los transpondedores de avión, tales como el nombre, tipo, posición, tamaño, rumbo, velocidad y destino de la embarcación varias veces por minuto. La información se recibe, se procesa y es aplicada por embarcaciones cercanas para ayudar con la prevención de colisiones.

La información transmitida por los buques no solo es compartida entre ellos, sino que también es recibida por las estaciones receptoras AIS terrestres y satelitales, esto permite proveer dicha información a los puertos, instalaciones, pilotos, remolcadores, personal de emergencia, propietarios y operadores de buques, y autoridades reguladoras para ayudar a operaciones marítimas seguras, protegidas y eficientes.

No solo los buques envían información, las estaciones terrestres también pueden transmitir información, por ejemplo, el Canal de Panamá utiliza el AIS para proporcionar información sobre la lluvia a lo largo del canal y el viento en las esclusas. Mostrando todos los datos en la pantalla de navegación de los buques puede resultar de gran ayuda para la tripulación de navegación para tomar decisiones de la forma más informada posible y así reducir considerablemente la posibilidad de que se produzca un siniestro marítimo.

El sistema AIS consta de un transmisor VHF, dos receptores TDMA VHF, un receptor DSC VHF y un enlace de comunicaciones electrónicas marinas estándar a los sistemas de visualización y sensores a bordo. La información de posición y tiempo se deriva normalmente de un receptor GPS integral o externo. Otra información transmitida por el AIS se obtiene electrónicamente del equipo a bordo a través de conexiones de datos marinas estándar.

En aguas costeras, las autoridades costeras pueden establecer estaciones AIS automatizadas para monitorear el movimiento de embarcaciones a través del área. Las estaciones costeras también pueden utilizar los canales AIS para las transmisiones de tierra a barco, para enviar información sobre mareas, NTM y condiciones meteorológicas localizadas. Las estaciones costeras pueden

usar el AIS para monitorear el movimiento de cargas peligrosas y controlar las operaciones de pesca comercial en sus aguas. El AIS también se puede utilizar para operaciones SAR que permitan a las autoridades SAR utilizar información AIS para evaluar la disponibilidad de otras embarcaciones en las proximidades del incidente.

### 2.2.1 Funcionamiento

Según el tipo de buque se clasifican en dos tipos de AIS:

- Clase A: obligatorio para los buques con 300 de tonelaje bruto o más que se dediquen a viajes internacionales y todos los buques de pasaje.
- Clase B: proporciona funcionalidad limitada, destinado a embarcaciones no SOLAS, principalmente embarcaciones de recreo.

Opera principalmente en dos canales VHF:

- AIS A: funciona en 161.975 MHz – Canal 87B (Simplex)
- AIS B: funciona en 162.025 MHz – Canal 88B (Duplex, para barco a tierra)

El medio de transmisión de la información es compartido, por ello es necesario usar un método para evitar solapamientos, AIS utiliza para intercambiar la información SOTDMA, gracias a ello, las embarcaciones conocen, en un minuto, la posición, el rumbo y la velocidad de todos los barcos dentro del alcance del VHF.

El alcance de la banda VHF marítima tiene un alcance aproximado de 25 millas náuticas, siempre y cuando la potencia de emisión sea lo suficientemente alta y la propagación se produzca en línea recta para poder distinguir una señal de voz del ruido. Con AIS la distancia aumenta hasta las 60 millas náuticas, ya que no hace falta comprender palabras ni voz, solo hay que comprender mensajes digitales.

En la siguiente tabla se definen los intervalos de envío:

Tipo de Barco	Intervalo de Información
Barco anclado	3 min
Barco en movimiento de 0 a 14 nudos	12 s
Barco en movimiento de 0 a 14 nudos con cambio de derrotero	4 s
Barco en movimiento de 14 a 23 nudos	6 s
Barco en movimiento de 14 a 23 nudos con cambio de derrotero	2 s
Barco en movimiento a más de 23 nudos	3 s
Barco en movimiento a más de 23 nudos con cambio de derrotero	2 s

Tabla 1 - Frecuencia de envío

## 2.3 Receptor AIS de bajo coste

### 2.3.1 Introducción

En desarrollos anteriores, se diseñó un sistema de bajo coste para recibir tramas AIS, donde a partir de un prototipo hardware basado en Raspberry Pi, un decodificador AIS y una antena. El coste total de estos componentes no supera los 70 euros, mientras que en el mercado los precios suelen oscilar entre los 300 euros, los más baratos y simples, y hasta más de 1000 euros. También desarrollo una aplicación para la gestión de tramas AIS en entorno Python.

Todo en conjunto se ha probado en condiciones reales, cumpliendo requerimientos y siguiendo la metodología de Marine Traffic logrando un decodificador con mínimas pérdidas de información.

### 2.3.2 Diseño hardware

Como elemento principal de computación se optó por una Raspberry Pi modelo 3, ya que se trata de un terminal muy versátil y con una gran libertad a la hora de programar, sumando a esto el bajo coste que supone teniendo en cuenta que los terminales de AIS suelen ser mucho más caros.

Comparación a nivel físico con un receptor estándar:

- En el transpondedor estándar el equipo necesario viene integrado
- La sensibilidad en la recepción es prácticamente idéntica, rondando los -107 dBm.
- En receptores estándar el consumo va de 0.5 W a 6 W, si tiene función de transmisión asciende hasta los 40W. La Raspberry Pi 3 llega hasta los 1.8 W
- En cuanto a la tasa de transmisión estamos hablando de igualdad para todos, 38400 baudios.
- La conexión antena-terminal se realiza mediante una conexión USB, fácil conectividad y estándar muy popular, a cambio aumentaría el ruido.

Para conectar la antena con el receptor se ha optado por uno coaxial con conector BNC macho para la conexión, importante que el cable no sea de una longitud mayor a 5 metros ni inferior a 3 metros, ya que eso supondría unas pérdidas y un ruido que puede hacer que las tramas no lleguen en condiciones de ser tratadas. Las especificaciones de la antena se muestran en la tabla 3.

Parámetros	Valor
Banda de frecuencia (MHz)	136-175
Ancho de banda (MHz)	5
Ganancia (dBi)	3
Impedancia ( $\Omega$ )	50
Polarización	Lineal
Tipo de conector	SMA
Longitud (cm)	150

Tabla 2 - Especificaciones técnicas de la antena

Como receptor de tramas AIS se ha optado por el receptor “dAISy AIS Receiver”, ya que su sencillez y bajo coste junto con su eficiencia y tamaño lo hacen especialmente eficaz a la hora de desarrollar un receptor AIS de bajo coste. El consumo estimado será de unos 100mW y no requiere de ningún tipo de instalación. El receptor posee sistemas de leds, con el cual podemos saber en qué estado está el receptor sin tener que acceder por ningún menú al receptor.

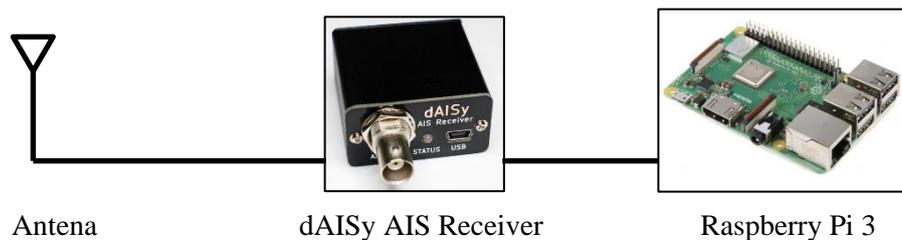


Figura 1 – Esquema receptor AIS

### 2.3.3 *Diseño software*

El software se diseñó para interpretar las 5 tramas más importantes para el objetivo del proyecto que se tenía entre manos.

Se partió del sistema operativo raspbian para el desarrollo, siendo este descargable desde la página web oficial, <https://www.raspberrypi.org/downloads/raspberry-pi-os/>, completamente gratuito, fácil de instalar y de configurar.

Para el lenguaje principal para el desarrollo software se optó por el uso de Python. Python se trata de un lenguaje de programación dinámico, y al ser un lenguaje interpretado no tiene la necesidad de ser procesado mediante un compilador.

El módulo decodificación consta de un bucle para encontrar el puerto USB donde la antena esté conectada. El programa trata cada tipo de trama por separado, usando una función para cada una, esto se planeó de esta manera ya que la trama de tipo 5 tiene algunas cosas diferentes al resto. Concretamente hay 3 funciones una para las tramas de tipo 5, otra para las tramas de tipo 4 y otra para las tramas de tipo 1, 2 y 3. Una vez decodificadas se almacenan en la base de datos, con la información lista para representarse en un mapa

Para la interfaz gráfica se optó por la librería wxPython, buscando que la interfaz sea sencilla y simple a la vez que facilite a la interpretación de los datos. La interfaz tendría 5 menús: el primero crea los mapas, el segundo para futuro, el tercero captura tramas automáticamente, los dos últimos puedes introducir tramas manualmente.

El diseño de la base de datos se optó por SQLite ya que así la gran cantidad de datos y su diversidad no es ningún problema, además es muy fácil de instalar y de integrar.



Para conectarse a la base de datos desde Python, primero hay que instalar la librería correspondiente, después hay que indicar donde se encuentra almacena y mediante la función correcta se establece la conexión con la base de datos.

Finalmente, la representación y geolocalización de los barcos es la parte más visible de la misma, donde se pueden ver los barcos donde están posicionados y que ruta están siguiendo. Para visualizar el mapa solo hace falta tener las coordenados donde se quiere centrar el mapa y este ya nos genera una vista de satélite, de una forma muy parecida Google Maps, y añadir cada barco para que este esté como un marcador, se puede interactuar con él para ver sus parámetros.

## 2.4 Data market

Actualmente existen varios data market en funcionamiento, todos ellos ofrecen varios servicios en función de las necesidades del cliente, entre ellos están MarineTraffic, VesselTracker y AISHub. En este apartado se mostrarán dichos data market viendo cual es la manera en que funcionan.

### 2.4.1 *MarineTraffic*

MarineTraffic es el proveedor líder mundial de rastreo de barcos e inteligencia marítima. Tienen como núcleo de trabajo la monitorización de los movimientos de los barcos. Cuentan con una red costera de estaciones receptoras AIS, además de varios receptores satelitales, que nutren sus bases de datos. Aplicando algoritmos y fuentes de datos complementarias proporcionan información útil sobre la actividad del transporte marítimo.

El alcance de MarineTraffic es global, tienen las oficinas principales en Reino Unido, Grecia y Singapur, pero cuentan con presencia en los principales centros marítimos del mundo proporcionándoles acceso directo a los mercados con los cuales trabajan.

En cuanto a sus servicios tienen una gran variedad, el primero sería el servicio online, donde puedes seleccionar un plan: estándar, profesional o global, donde cada uno ofrece unas mayores coberturas que el anterior. En la figura 3 se pueden ver los detalles de cada plan.

También ofrecen la posibilidad de contratar un simple servicio, estos son muy diversos, desde las condiciones climáticas que hay a lo largo de la ruta hasta el seguimiento vía satélite de un barco. En la figura 2 se pueden ver algunos de los servicios ofrecidos por MarineTraffic.

<p><b>Vessel Particulars</b> →</p> <p>Management, size and capacity details for the commercial fleet</p> <p><b>30.00EUR</b> per month, billed monthly</p> <p>TRY 7-DAYS FREE</p>	<p><b>Nautical Charts</b> →</p> <p>Navigable water charts, bathymetry, landmarks and sea hazards</p> <p><b>8.00EUR</b> per month, billed monthly</p> <p>TRY 7-DAYS FREE</p>	<p><b>Voyage Planner Pro</b> →</p> <p>Access all voyage planner features</p> <p><b>16.00EUR</b> per month, billed monthly</p> <p>TRY 7-DAYS FREE</p>	<p><b>Advanced Live Map</b> →</p> <p>Filters for the vessels shown on the Live Map, auto-refresh and full screen view</p> <p><b>14.00EUR</b> per month, billed monthly</p> <p>TRY 7-DAYS FREE</p>
<p><b>Vessel Events</b> →</p> <p>LITE PLUS</p> <p>Vessel voyage events with 30 days past activity</p> <p><b>35.00EUR</b> per month, billed monthly</p> <p>TRY 7-DAYS FREE</p>	<p><b>Port Congestion</b> →</p> <p>Figures and trends of port congestion for 1 year in the past</p> <p><b>55.00EUR</b> per month, billed monthly</p> <p>TRY 7-DAYS FREE</p>	<p><b>Weather Maps</b> →</p> <p>Measured and forecast weather information, globally.</p> <p><b>15.00EUR</b> per month, billed monthly</p> <p>TRY 7-DAYS FREE</p>	<p><b>Advanced Density Maps</b> →</p> <p>Visualise global shipping patterns, by vessel type</p> <p><b>6.00EUR</b> per month, billed monthly</p> <p>PURCHASE NOW</p>

Figura 2 - Servicios ofrecidos por MarineTraffic

También ofrecen la posibilidad de acceder a estos datos vía API, no solo vía online, con lo cual sería mucho más fácil la integración con una posible aplicación o servicio que se tenga. A través de la API se puede acceder a posiciones de los barcos, eventos de estos, como información detallada de arribo o zarpe en puerto, información específica del barco, como por ejemplo fotos y características, información relativa al trayecto del barco, como las distancias, las rutas, el tiempo estimado de llegada, etc. además también se puede acceder a los datos de la cuenta para bien administrar la flota de barcos, el saldo de la cuenta, etc.

Finalmente ofrecen la posibilidad de acceder al histórico de todos los barcos en una zona y tiempo determinado, para ello habría que contactar con ellos a través del formulario web que tiene a disposición de quien lo desee.

#### 2.4.2 Vesseltracker

Vesseltracker es un proveedor líder de servicios de información marítima y movimientos de embarcaciones AIS, fundado en 2006 por un especialista en tecnología de la información y un gerente de operaciones portuarias.

Algunas de las características de vesseltracker son:

- Entrega de datos históricos y en tiempo real personalizados, incluidos informes y análisis sobre movimientos de embarcaciones, eventos portuarios y proyectos personalizados para que pueda utilizar nuestros datos para aplicaciones internas, análisis y / o sistemas de informes.
- Base de datos completa de embarcaciones con especificaciones, características, equipo, propiedad e información de gestión, con todos los cambios registrados.



- Base de datos de más de 1.000.000 de imágenes de buques y puertos.
- Alertas sobre el estado de la embarcación y las regiones personalizables por correo electrónico, SMS y teléfono, además de varias listas de embarcaciones con informes de mediodía por correo electrónico para ahorrar tiempo y hacer que el sistema funcione para usted.
- Información actualizada y sistemas de alerta sobre embarcaciones previstas, llegadas y salidas para un solo puerto o una lista de puertos.
- Vistas de mapas y capas que incluyen el satélite Google, Google Earth, cartas náuticas e información meteorológica y de piratería global y local.
- Cálculo de distancia y ETA para permitir la predicción precisa de cuándo llegará un barco.

Al igual que Marinetrffic, vesseltracker ofrece también servicios online y servicios vía API. Para los servicios online existen 4 planes, el primero sería el gratuito, donde se podría acceder a las fotos y novedades del barco o puerto y un mapa de densidad de rutas además de tener limitado el acceso a los datos AIS así como al histórico, solo estarían disponibles para barcos de pasajeros, los otros tres tipos de planes van creciendo en funcionalidades según se pague, llegando a tener acceso a todos los barcos, ya sea por vía terrestre o mediante uso de los satélites, además de todo el histórico.

Para la API se tendría una funcionalidad muy parecida: seguimiento global de los barcos, aviso de cualquier tipo de evento que se produzca en el barco, llegadas y salidas de puerto, información característica de los barcos e información sobre el tiempo climatológico, así como el tiempo esperado de llegada a destino.

## 2.5 FIWARE

FIWARE es una plataforma abierta que define un conjunto universal de estándares para la gestión de datos de contexto que facilitan el desarrollo de soluciones inteligentes para diferentes dominios como Smart Cities, Smart Industry, Smart Agrifood y Smart Energy. La plataforma proporciona especificaciones de API públicas y protocolos interoperables para la creación de nuevos servicios y aplicaciones de internet.

FIWARE permite portar los datos para diferentes aplicaciones, también aporta un conjunto de habilitadores genéricos que permiten acceder a información relevante mediante la API NGSI, lo que viene a significar que habilita una forma rápida y eficiente de recolección, publicación, intercambio y proceso de grandes volúmenes de datos.

Además, FIWARE, también proporciona soporte para una infraestructura de hosting en la nube mediante habilitadores genéricos, accesibles mediante REST APIs. Esta arquitectura en la nube incluye servicios como IaaS (Infracstructure as a Service), servicios de almacenamiento de datos y metadatos en la nube, y servicios de gestión y monitorización de aplicaciones y dispositivos.

El habilitador genérico de procesamiento de datos se divide en dos bloques funcionales. Como primera parte está el bloque de procesamiento por lotes, este aporta infraestructuras bajo demanda que soportan herramientas de análisis de Big Data. Y en la otra parte está el bloque de procesamiento de flujos, este proporciona una interfaz modular para realizar el procesamiento de los datos en tiempo real, utilizando Apache Storm.

En cualquier solución inteligente existe la necesidad de recopilar y gestionar la información del contexto, procesando esa información e informando a los actores externos, permitiéndoles actuar y por lo tanto alterar o enriquecer el contexto actual. El componente FIWARE Orion Context Broker es el componente central de cualquier plataforma "Powered by FIWARE". Permite al sistema realizar actualizaciones y acceder al estado actual del contexto.

### **2.5.1 FIWARE Orion Context Broker**

El Context Broker es el que se encarga de la gestión de la comunicación entre las aplicaciones y los dispositivos con el fin de separar las aplicaciones IoT de las instalaciones de los dispositivos. Para la comunicación entre los dispositivos y las aplicaciones, el Context Broker utiliza el protocolo NGSIv2, lo cual permite generar y recibir notificaciones de eventos contextuales.

Las entidades (*entities*) son las unidades destinadas a almacenar los datos, cada una se identifica mediante un identificador único (*id*) y pertenece a un tipo (*type*), varias entidades pueden formar parte a un mismo tipo. Las entidades, además, pueden tener atributos (atributes), estos atributos describen la información que contiene la entidad mediante un valor (*value*) y un tipo de datos (*type*), opcionalmente pueden contener metadatos (*metadata*) introducidos por el usuario.

Cabe destacar que el valor que se guarda es el último valor de cada atributo de las entidades. Los datos se almacenan en una base de datos NoSQL, como podría ser MongoDB.

Estas entidades pueden ser clasificadas según un sistema jerárquico, siendo el nivel principal un *Fiware-Service*, pudiendo contener niveles inferiores, formando un árbol, el camino hacia los cuales se llama *Fiware-ServicePath*.

Se pueden diferenciar dos roles para Orion, el rol de productor, los cuales se encargan de la creación y actualización de los datos de las entidades, y el rol de los consumidores, los cuales se suscriben a Orion para recibir notificaciones de los datos de las entidades a los que, previamente, se han suscrito. En la figura 4 se puede observar cómo sería esto.

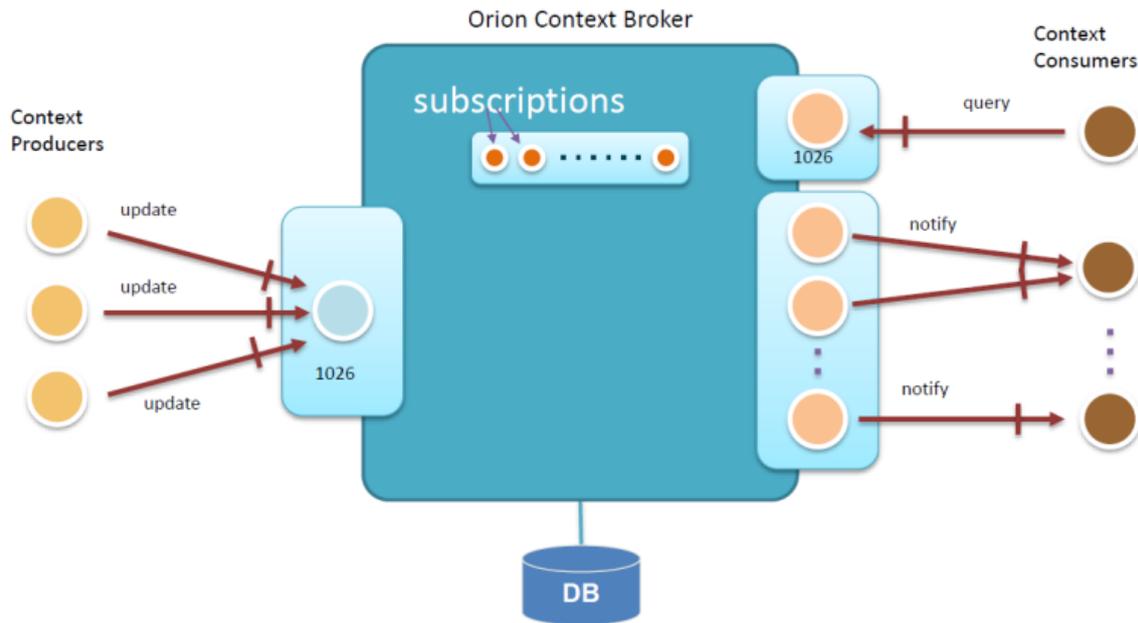


Figura 3 Esquema Orion Context Broker

## 2.6 Mongo DB

MongoDB es una base de datos gratuita y de código abierto de tipo NoSQL, orientada a documentos. El almacenamiento de datos en MongoDB es mediante el uso de documentos dinámicos en formato BSON, gracias a esto la estructura de datos puede variar con el tiempo y entre diferentes documentos.

Aunque la consola de MongoDB este escrita en Javascript, esto permite al usuario una fácil gestión de la base de datos mediante funciones en dicho lenguaje, existen múltiples librerías de clientes en muchos lenguajes de programación como Python, Java, C++, etc. [X]

## 2.7 Docker

Docker es un Proyecto de código abierto utilizado principalmente para permitir al usuario la automatización del despliegue de aplicaciones dentro de contenedores de procesos. Para ello un pequeño subconjunto de un sistema de archivos, los necesarios para que la aplicación se ejecute correctamente, se integran en un contenedor ligero para su posterior implementación.

Como los contenedores no ejecutan imágenes de un sistema operativo, sino que ejecutan uno o más procesos específicos no tienen una dirección IP propia, la comunicación entre contenedores se realiza mediante la escucha en puertos específicos que se han de “exponer”. En las últimas versiones de Docker ya se permite la asignación de una IP propia e interfaces de red mapeadas.

Compose, es una herramienta para definir y ejecutar aplicaciones Docker de varios contenedores. Con Compose, usa un archivo YAML para configurar los servicios de su aplicación. Luego, con

un solo comando, crea e inicia todos los servicios desde su configuración. Funciona en todos los entornos: producción, desarrollo, pruebas, etc.

Para crear un archivo YAML basta con indicar que contenedores queremos ejecutar, opcionalmente podemos decir donde se almacenará la información, además de puertos expuestos para llegar hasta los servicios de los contenedores, así como las conexiones entre los puertos y alguna otra configuración extra.

```
version: "3.8"
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ./code
      - logvolume01:/var/log
    links:
      - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

Figura 4. Ejemplo de archivo YAML de Docker Compose

## Capítulo 3. Desarrollo

### 3.1 Modificación del receptor AIS

#### 3.1.1 Hardware

Para la realización de este proyecto, se reutilizará todo el hardware mencionado en el apartado 2.2. Lo primero a realizar es la comprobación de que todo el hardware funciona correctamente,



Figura 5 Antena receptora

para ello colocamos la antena en lugar adecuado para la correcta recepción de la señal de los barcos. Como el proyecto mayoritariamente ha sido desarrollado en la escuela técnica superior de ingenieros de telecomunicación, en la segunda planta, la antena se tuvo que colocar en la ventana lo más cerca posible.

Para comprobar que reciba las señales que emitían los barcos, esta, se conectó al receptor dAISy AIS, tras encenderlo todo, al cabo de unos momentos el receptor dAISy AIS comenzó a parpadear con la luz led verde, indicando que se recibían las tramas desde los barcos.

En cuanto a las Raspberry Pi, formatea con el sistema operativo raspbian y se comprueba que esta recibe las tramas correctamente, para ello tan solo es necesario escuchar los que entra por el puerto donde está conectada la antena. En la figura 6 se pueden ver las tramas AIS recibidas.

```
!AIVDM,1,1,,A,13aEOK?P00PD2wVMdLDRhgvL289?,0*26
!AIVDM,1,1,,B,16S`2cPP00a3UF6EKT@2:?vOr052,0*00
!AIVDM,2,1,9,B,53nFBv015J<thHp6220H4heHTf222222222221?50:454o<`9Q51UDp,0*09
!AIVDM,2,2,9,B,8888888888888880,2*2E
```

Figura 6. Tramas AIS recibidas

#### 3.1.2 Software

Tras comprobar que el hardware funciona correctamente, se procede a la instalación del software.

Como se busca un tratamiento global de las tramas y no solo de las principales se optará por la instalación de rPiAIS, siempre teniendo en cuenta que lo que se está buscando es obtener una cobertura total de la costa de la comunidad valencia, por ello se busca un software ligero y que permita la retransmisión de los datos, no hace falta que se haga mucho más en las estaciones receptoras.



rPiAIS es un software desarrollado por AISHub, se trata de una implementación de AIS Dispatcher para dispositivos Raspberry Pi. Admite una amplia gama de receptores AIS conectados a través de un cable serie, USB o Ethernet y proporciona:

- Recepción y procesamiento de todos los mensajes AIS estándar
- Transmisión de datos AIS sin procesar a múltiples destinos
- Monitoreo avanzado del estado del dispositivo, cobertura y estadísticas de mensajes
- Mapa electrónico integrado con todos los barcos recibidos

Como principal punto a favor a la hora de optar por rPiAIS es el procesamiento de todos los mensajes AIS estándar, ya que así no se descartaría ninguna trama AIS. Además, ofrece la posibilidad de transmitir las tramas recibidas sin procesarlas, ya que así ocupan un menor ancho de banda, siendo más eficiente su transmisión.

Como *input*, rPiAIS soporta 3 opciones:

- **Serial input:** los datos vendrían mediante receptores AIS conectados por puerto serie o bien por USB, mediante el botón *View devices* se puede seleccionar por que entrada, de las que tiene la Raspberry Pi, estaría conectado el receptor AIS.
- **TCP Client/Server:** rPiAIS crearía una conexión TCP para recibir los datos de los receptores AIS, se puede especificar en caso de que sea necesario la dirección IP del host y el puerto.
- **UDP Server:** rPiAIS se quedaría a la escucha, en el puerto que se le indique, la llegada de paquetes UDP con la información de los receptores AIS.

Como *output*, rPiAIS soporta reenvío de información, el reenvío de esta información utiliza el protocolo UDP, por tanto, solo hay que indicar que host y que puerto tiene que recibir la información retransmitida.

En cuanto a los *settings* cabe destacar 3 opciones:

- Como primera opción tendríamos el posicionamiento de la estación receptora, esto se especifica introduciendo la latitud y longitud.
- *Downsampling time*, esta opción permitiría reducir el tráfico que se reenvía mediante la eliminación de tramas de un mismo barco, si este reporta en un intervalo de tiempo, máximo 1 minuto, varias veces la su posición, rPiAIS solo reenviaría una vez el reporte.
- *Duplicates removal*, activando esta opción los mensajes duplicados que se fueran a enviar serían eliminados.



### 3.2 Relación de confianza con AISHUB

AISHub ofrece la posibilidad de establecer una relación de confianza, donde por nuestra parte se le envían la información de nuestro receptor y a cambio ellos nos ofrecen toda la información que tienen, ya sea mediante API o mediante servicio online, eso sí, su información viene retrasada 2 horas.

Para establecer esta relación de confianza se tiene que acceder a su página web, <https://www.aishub.net/>, donde habrá un botón de “JOIN US”, esto llevará a un formulario que, tras rellenarlo y cumplir los requisitos mínimos, proporcionará un conexión TCP dedicada donde se tendrá acceso a las tramas AIS sin decodificar, que ellos tengan acceso, global, también se proporcionará una cuenta personal para usar su API y recibir directamente en formato XML o JSON los datos, estos si vendrán decodificados. Además, se podrá monitorizar la información que se envía desde nuestra parte y también se podrá observar dicha información desde VesselFinder. Si por algún motivo AISHub no recibe nada por parte de nuestro receptor AIS durante más de 6 horas, automáticamente recibiremos un correo notificando la incidencia, si pasan 48 horas sin recibir nada, se suspenderá todo a la espera de volver a recibir más información.

Los términos de servicio de AISHub son pocos y claros:

- Cada usuario, como mínimo se ha de proporcionar información de una estación de recepción.
- Cada usuario tendrá acceso total a la información agregada de todas las fuentes de AISHub.
- No hay restricciones a la hora de usar los datos, se es libre de usarla, tanto como publicarla gratuitamente como para usos comerciales.
- AISHub no se hace responsable de pérdidas causadas por fallos técnicos o información defectuosa.

### 3.3 Plan de cobertura de la Comunidad Valenciana

Como primera idea de propuso cubrir toda la zona costera de la comunidad valenciana usando estos receptores AIS de bajo coste, teniendo en cuenta que las tienen aproximadamente 40 km de alcance. Asumiendo que la costa de Comunidad Valencia tiene alrededor de 400 km de costa harían falta entre 10 y 20 receptores para una cobertura óptima de toda la costa. Sin embargo, estos puntos “óptimos” puede que sea más complicado instalar un receptor AIS, ya que estos necesitan de corriente eléctrica y conexión a internet, sin conexión a internet pueden recibir la información de los barcos, pero no harían nada con ella.

En la Comunidad Valenciana hay un total de 31 puertos, de norte a sur estos son: Vinaròs, Benicarló, Peñíscola, las Fuentes, Oropesa, Castelló, Burriana, Canet d'en Berenguer, el port de Sagunt, la Pobla de Farnals, Port Saplaya, Valencia, el Perelló, Cullera, Gandia, la Goleta, Denia, Xàbia, Nou Fontana, Moraira, Les Bassetes, Calp, Puerto Blanco, L. Campomanes, la Galera, l'Olla d'Altea, Altea, Benidorm, la Vila Joiosa, el Campello, Costa Blanca, R. C. de Regates, Alacant, Santa Pola, Tabarca, Guardamar, Torrevieja, Cabo Roig, Dehesa de Campoamor, Torre de la Horadada

Asumiendo un coste total por receptor AIS de 100 euros, el coste que supondría cubrir todos los puertos, incluyendo dársenas, embarcaderos y puertos deportivos ascendería a 3100 euros. Instalando todos estos receptores AIS se aseguraría una cobertura más que total de toda la costa de la Comunidad Valenciana.

Sin embargo, teniendo en cuenta que muchos de estos puertos están prácticamente pegados a escasos kilómetros unos de otros, se puede observar en la figura 4 se podría utilizar un solo receptor AIS para cubrir a varios puertos.

En este segundo caso, reduciendo el número de receptores AIS, el total de estos receptores AIS quedaría reducido a 22-24 receptores.

En este segundo caso, el coste total de cobertura de la costa de la Comunidad Valenciana ascendería a 2400 euros, una cifra menor en comparación a la primera opción, concretamente, un ahorro de 1900 euros. Con esta opción, la cobertura sería más que decente para toda la costa sin sobrecargar la costa de receptores AIS.

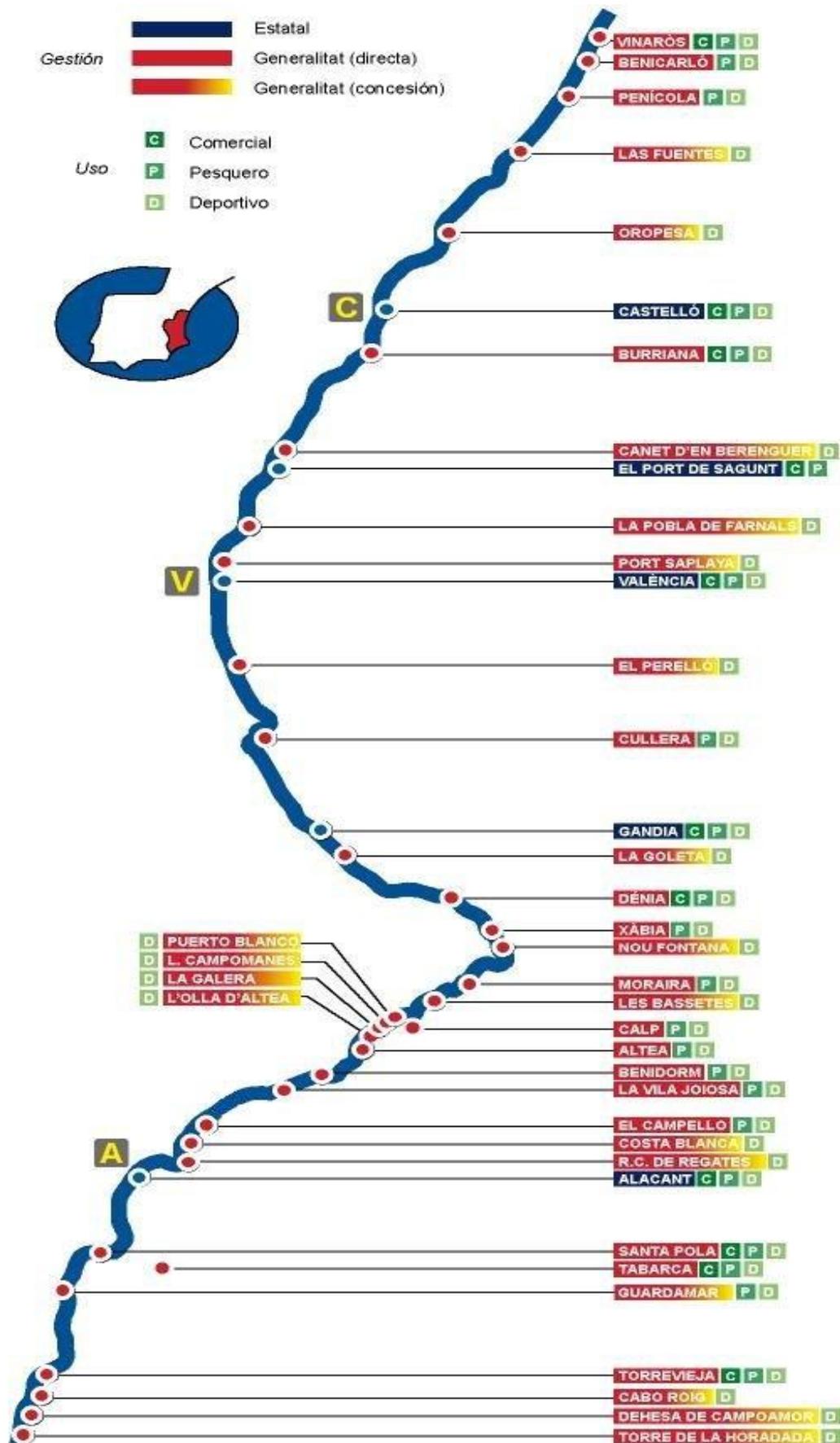


Figura 7 Puertos en la Comunidad Valenciana

## Capítulo 4. Implementación

En este capítulo se abordará todo lo relacionado con la reunión de la información y el proceso de creación de una plataforma para la publicación del contenido, el data market.

Para ello usaremos FIWARE, este nos permite, por una parte, tener siempre disponible la última posición, o actualización de una entidad, en este caso, la entidad sería un barco, mediante Orion, mencionado en el capítulo 2, además de guardar un histórico de las actualizaciones de la entidad, STH, también mencionado en el capítulo 2. Sin embargo, para más robustez, también se guardarán las tramas sin procesar en una base de datos MongoDB.

Para que la información llegue a Orion, es necesario la creación de unos agentes, encargados de recibir la información que envían los receptores AIS al servidor, estos se crearan usando Python, concretamente la librería pyngsi, la cual nos permite insertar los datos en Orion a la vez que en la base de datos MongoDB.

Finalmente, la plataforma para la publicación de datos se realizará mediante CKAN, CKAN es una herramienta para crear sitios web de datos abiertos, ayuda a administrar y publicar colecciones de datos.

### 4.1 Arquitectura

Como se ha visto en el capítulo 3, se ha planificado la instalación de 16 receptores AIS en los puertos distribuidos por toda la costa de la Comunidad Valenciana. Estos receptores funcionan en un Raspberry Pi con el sistema rPiAIS funcionando, este sistema permite la retransmisión de los datos en crudo, sin procesar utilizando el protocolo UDP. Aprovechando esta funcionalidad se reenviarán todos los datos a un servidor central donde se preparan los datos para poder ser publicados.

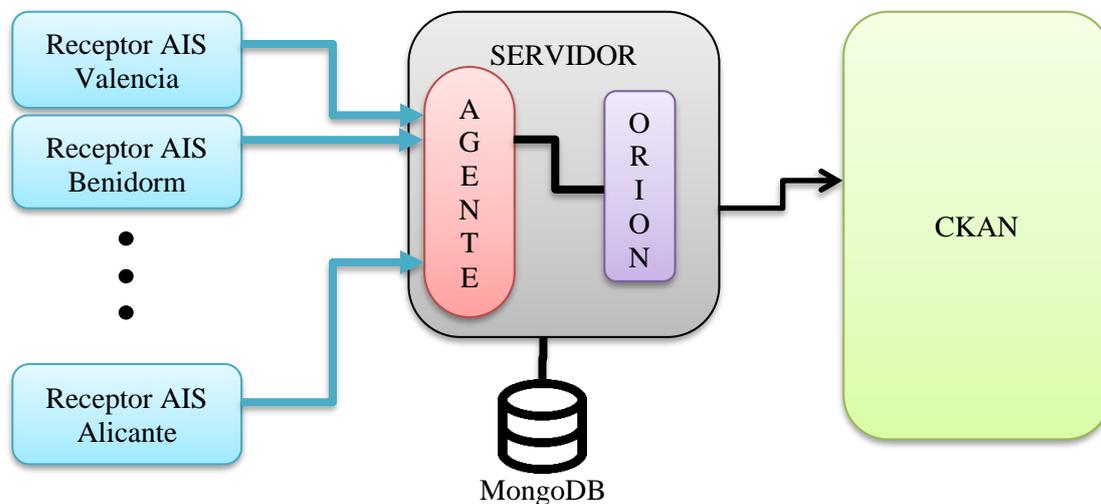


Figura 8 - Arquitectura

## 4.2 Servidor y base de datos

Para la instalación de los componentes necesarios se ha optado por la utilización de contenedores Docker, como se ha visto en el capítulo 2, estos contenedores nos proporcionan una solución rápida para poner en marcha cualquier tipo de servicio.

En este caso, el context bróker de FIWARE se obtendrá del repositorio oficial de DockerHub, donde se encuentran la gran mayoría de imágenes para utilizar.

## 4.3 Agentes

El agente es el componente que permite a los dispositivos IoT enviar sus datos y ser administrados desde un Context Broker utilizando sus propios protocolos nativos, el agente es el encargado de la transformación a protocolos aceptados por el Context Broker, además también realiza funciones de seguridad.

En este caso, Orion Context Broker solo funciona con solicitudes NGSI-v2, por tanto, el agente proporciona una interfaz para dichas solicitudes, facilitando así el uso de distintos dispositivos con distintos protocolos nativos.

El agente que se va a utilizar en este caso se trata de un módulo de Python, pyngsi, sus funciones principales son:

- Asignación de datos nativos de Python a entidades NGSI
- Escribir entidades NGSI en Fiware Orion
- Manejar los datos entrantes a través de un interfaz común
- Calcular estadísticas

### 4.3.1 *Agente.py*

El código del agente se divide en 3 partes principales, la primera de ellas sería la adquisición de datos, la segunda el tratamiento de la información y modelado de los datos para poder ser enviados correctamente a Orion, y finalmente el envío de estos.

La adquisición de datos se puede realizar de diversas maneras, mediante ficheros JSON, a través de una entrada manual como podría ser el mismo teclado, un fichero Excel, un servidor ftp, etc. para este caso en concreto la opción que se usará será la de ServerUdp, con ella el agente actuará como un servidor UDP a la espera de la información recibida a través del puerto UDP que se le asigne, en este caso quedará a la espera de recibir las tramas en el puerto 2876. Como todo estará en el mismo sistema la dirección IP sería 127.0.0.1.

Para ello se importa la clase ServerUDP y se declara el servidor, indicando dirección IP y puerto:

```
from pyngsi.server import ServerUdp
src = ServerUdp('127.0.0.1', '2876')
```

Para el envío de la información se realiza algo similar, se declara la clase y se indica la dirección IP donde está instalado Orion:

```
from pyngsi.sink import SinkOrion
sink = SinkOrion('127.0.0.1')
```

Finalmente, para el tratamiento de los datos se ha de definir un proceso y un modelo de datos:

```
def build_entity(row: Row) -> DataModel:
```

Con esto lo que se consigue es que cada vez que se reciba una trama AIS sea procesada, una a una según vayan llegando. Como las tramas que llegan están codificados con el estándar AIS, primero hay que decodificarlas, para ello se utilizara un módulo de Python desarrollado para este fin, pyais, su funcionamiento es sencillo y tan solo hay que entregarle el mensaje codificado y decirle como tiene que sacarlo, para este caso, basta con que nos saque un JSON con la información.

```
msg = row.record
decoded_json_msg = msg.decode().to_json()
msg_py = json.loads(decoded_json_msg)
```

Con estas tres líneas se obtiene un JSON con toda la información decodificada proveniente de las antenas AIS. Tras esto se procederá a crear el modelo de datos para su inserción en Orion.

```
ide = msg_py['MMSI']
m = DataModel(id=ide, type="AIS_DATA")
m.add("name", msg_py['name'])
```

```
m.add("imo",msg_py['imo'])
m.add("status",msg_py[' navigationStatus'])
m.add("speed",msg_py[' speedOverGround'])
...
m.add("latitude",msg_py[' latitude'])
m.add("longititude",msg_py[' longititude'])
...
return m
```

Para la creación del modelo de datos hay que definir un id, que servirá para identificar los datos una vez estén introducidos en Orion, para este caso se ha optado por poner como identificador el MMSI, ya que sirve también como identificador único de barco. Después se define un tipo, que será el tipo que lleven todos los datos que se envíen a través de este agente a Orion. Finalmente se añaden todos los campos necesarios para que estos sean enviados correctamente, al hacer la función *m.add*, lo que se está haciendo es identificar qué tipo de dato le estamos pasando, por ejemplo, en el caso del nombre sería un *string*, el caso del IMO un *number* y así con todos los demás, esto permitirá posteriormente saber qué tipo de dato es cada uno y ayudar así a la interoperabilidad con otros sistemas. Como último paso faltaría crear el agente:

```
agent = NgsiAgent.create_agent(src, sink, process=build_entity)
```

Como añadido, se “dockeriza” el agente, esto no es mas que crear una imagen Docker del agente para que este pueda ser desplegado y ejecutado de una manera fácil y rápida. Para ello, creamos un Dockerfile con toda la información necesaria, para este caso:

```
FROM python:3
ADD agente.py /
ADD pyngsi/* /pyngsi/
RUN pip install pyais
RUN pip install pyngsi
CMD ["python", "/agente.py"]
```

Tras esto se ejecuta el comando *docker build* creando así la imagen Docker lista para su uso.



## 4.4 CKAN

CKAN es una plataforma de código abierta diseñada para almacenar y distribuir datos contenidos en bases de datos, también admite la distribución de datos contenidos en hojas de cálculo. Hay muchos portales basados en esta aplicación, así como gobiernos que la utilizan para mostrar su información, países como España están actualmente utilizando CKAN. <https://datos.gob.es/es/tecnologia>

Existe una extensión de FIWARE para poder integrar CKAN dentro del ecosistema de FIWARE, con ello se permite la publicación, gestión y visualización de los datos, además de mejorar el control del acceso y permitir la monetización de los datos. Concretamente las extensiones que se utilizan son las siguientes:

- OAuth2 para autenticar a los usuarios de CKAN.
- Private Datasets, esta extensión permite dar acceso a un conjunto de datos predeterminados a unos usuarios específicos.
- Vista NGSI permite la publicación de las consultas hechas al Context Broker.
- BAE Publisher, esta extensión es la que integra CKAN con el ecosistema FIWARE permitiendo la creación de productos y ofertas para un conjunto de datos.
- Vista de WireCloud permite crear visualizaciones de los datos.
- Data Requests, esta extensión permite pedir datos que no estén publicados, creando así una demanda de datos.

### 4.4.1 Instalación

Para la instalación de CKAN hay varias maneras, la primera será la típica instalación por paquetes, teniendo que además configurar todos los demás requisitos para que funcione, además, este método solo funciona en Ubuntu 18.04 o mayor. Para funcionar necesitaría tener instalado Ubuntu con Python, además de NGINX actuando como proxy, uWSGI con DataPusher y servidor web, PostgreSQL para almacenar los datos y Solr, Jetty y Redis para la búsqueda. En este caso hay que configurar todo paso a paso.

Para este trabajo se optó por la segunda opción, usar Docker, concretamente usando Docker Compose, con lo que nos ahorramos mucho, como por ejemplo la necesidad de tener específicamente Ubuntu instalado, además para la configuración de los componentes se hace todo a través de un fichero de entorno con todas las variables necesarias. En la figura 9 se puede observar el contenido de dicho fichero.

Para instalar CKAN, lo primero que hay que hacer es descargar todos los archivos a través de <https://github.com/ckan/ckan/tree/2.9>, una vez hecho esto se modifica el fichero .env según nuestro caso.

```
# Image: ckan
CKAN_SITE_ID=default
CKAN_SITE_URL=http://localhost:5000
CKAN_PORT=5000
#
# Image: db
POSTGRES_PASSWORD=ckan
POSTGRES_PORT=5432
DATASTORE_READONLY_PASSWORD=datapusher
```

Figura 9 - Fichero .env

Tras descargar y establecer el fichero .env, se procede a crear la imagen necesaria, para eso se ejecuta el comando: `docker-compose up -d --build`

Tras la ejecución del comando, se crearán varios contenedores Docker:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b7332bf2040f	docker_ckan	"/ckan-entrypoint.sh..."	49 minutes ago	Up 49 minutes	0.0.0.0:5000->5000/tcp	ckan
52c4c73ee44f	redis:latest	"docker-entrypoint.s..."	49 minutes ago	Up 49 minutes	6379/tcp	redis
ca0d6b3840df	docker_solr	"docker-entrypoint.s..."	49 minutes ago	Up 49 minutes	8983/tcp	solr
cb2956ef7bd4	clementmouchet/datapusher	"python datapusher/m..."	49 minutes ago	Up 49 minutes	0.0.0.0:8800->8800/tcp	datapusher
dd00bae2b435	docker_database	"docker-entrypoint.s..."	49 minutes ago	Up 49 minutes (healthy)	5432/tcp	database

El contenedor **ckan** contiene CKAN con extensiones estándar, **database** contiene la base de datos de CKAN, que luego también ejecuta la base de datos del almacén de datos de CKAN, **redis** contiene una imagen de Redis prediseñada, **solr** contiene una imagen SolR prediseñada configurada para CKAN, finalmente **datapusher** contiene una imagen prediseñada de CKAN Datapusher.

La base de datos y el usuario se crean al crear el contenedor database, sin embargo, se necesitan un par de configuraciones adicionales para que funcione correctamente, para ello ejecutamos el comando:

```
PS C:\Users\Dario\Desktop\ckan\ckan-2.9\contrib\docker> docker exec ckan /usr/local/bin/ckan-paster --
plugin=ckan datastore set-permissions -c /etc/ckan/production.ini | docker exec -i db psql -U ckan
```

Finalmente, reiniciamos CKAN para que los cambios funcionen y ya estaría todo listo para comenzar a mostrar los datos a través de la web. En la figura 10 se puede observar la interfaz de CKAN.

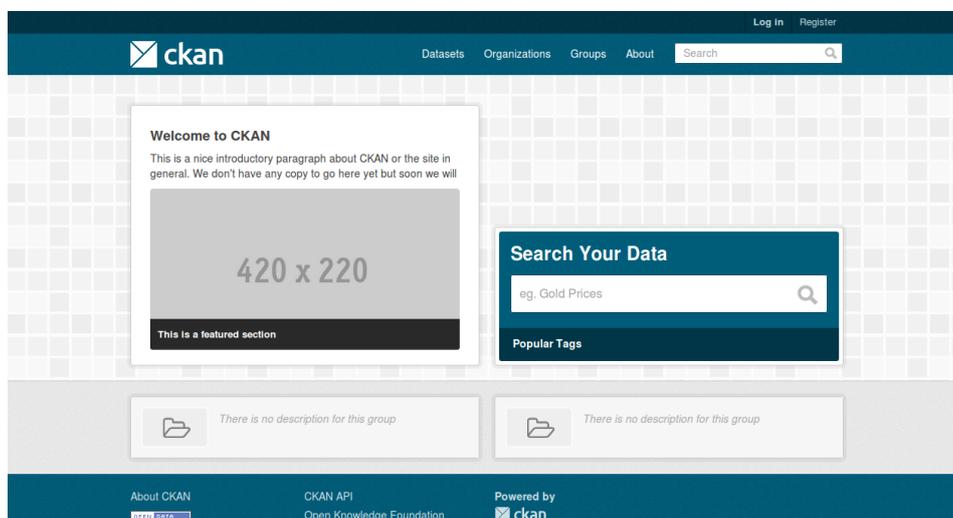


Figura 10 Pantalla CKAN

## 4.5 Escenarios

Y para finalizar describiremos diferentes escenarios en los que los servicios cognitivos puedan llegar a tener un uso, el primero sería el más fácil de ver, mejorar las estimaciones de tiempo de llegada de los barcos, ya que estos son puestos manualmente por la tripulación y raramente se actualiza, un segundo caso sería la optimización del atraque, carga y descarga de los barcos proporcionando datos sobre el estado de las grúas el patio, etc. y como tercer caso estaría el cuidado medioambiental del puerto, lo que viene a ser cuanto “contamina” o cuan contaminante son los trabajos realizados en los puertos.

### 4.5.1 Mejora de la predicción de ETA

En la cadena logística, el tiempo estimado de llegada es algo muy importante, ya sea el transporte por aire, mar o tierra, en este caso nos centraremos en el caso del transporte por mar, ya que tenemos acceso a las tramas AIS emitidas por los barcos.

Para aumentar la precisión de la predicción y la accesibilidad se puede reducir a realizar la predicción cuando el barco este cerca del puerto, así no es necesario que los puertos tengan acceso a datos globales de AIS, ya que muchos puertos pequeños y medianos no tienen las capacidades para acceder a estos servicios y solo disponen de datos AIS en vivo locales. Al mismo tiempo, como fuente de información añadida estaría la información de las boyas, estas proporcionan información sobre las olas, el viento, la temperatura, las corrientes, etc.

Los impactos más directos sobre la aplicación en este escenario serían los costes de mantenimiento de los activos y los datos, menos discusiones al saber de forma más precisa cuando va a llegar un barco, además, también gracias a la precisión se puede mejorar la planificación de atraque haciendo más eficientes las operaciones del terminal ayudando a reducir las emisiones del puerto.

### 4.5.2 Optimización de operaciones logísticas

Antes nos centramos en el tiempo de llegada, ahora, y como otra parte fundamental en la cadena logística nos centramos en todo lo que sería el poner la mercancía en el puerto y volver a cargar el barco con nueva mercancía.

Una vez ya se sabe cuándo va a llegar el barco el siguiente paso es optimizar el atraque y la carga y descarga de contenedores. Para poder realizar los análisis de forma correcta se requiere la participación de varios agentes, entre ellos los transportistas, las compañías navieras, las autoridades portuarias, los operadores de terminal, etc.



Para la planificación de atraques, se optimizará la distribución de los barcos la acercase a la terminal, así como la asignación la posición que deben ocupar en el puerto, también las grúas y el trabajo de estas. Para ello es necesario disponer de la información de los recursos disponibles del puerto según se vaya acercando la fecha de atraque del barco.

La otra operación logística que puede ser optimizada mediante los servicios cognitivos sería la planificación de carga y descarga de los contenedores, ya que la gran cantidad de contenedores puede llegar a suponer problemas no solo en el puerto, sino también en las entradas a la ciudad donde los camiones pueden llegar a colapsar una entrada a la ciudad ya que están conectadas directamente. Para llevar a cabo esta optimización sería necesario reducir el cuanto camino tienen que recorrer los camiones sin llevar carga además de la posición de los contenedores en los patios de los puertos.

#### **4.5.3 Cuidado medioambiental**

Como ultimo escenario estaría el cuidado medioambiental, usando servicios cognitivos junto con la información proveniente de AIS más información que aporten los puertos, del tipo, consumo de la terminal para la estancia de cada barco, los residuos que recogen de dichos barcos, la cantidad de tráfico que pasan por los muelles, etc., otra información que se puede llegar a sacar es el consumo de los barcos en las dependencias de los puertos, desde que llegan hasta que salen y con ello, haciendo uso de búsquedas avanzadas, llegar a saber las emisiones.

Finalmente, mediante el uso de fórmulas y algoritmos se puede llegar a tener un índice de cuanto contaminante es la actividad el puerto, y gracias a los servicios cognitivos ser capaces de analizar dicho índice y llegar a modelar soluciones que hagan posible reducir la contaminación producida por el puerto.

## Capítulo 5. Conclusiones y futuros desarrollos

### 5.1 Conclusiones

En el trabajo final de máster se ha desarrollado una plataforma para compartir la información de todos, o la gran mayoría, de los barcos que navegan por la costa de la comunidad valenciana, siempre y cuando estos estén usando el sistema AIS.

Para el desarrollo de este proyecto se observó que un compañero de la misma universidad hizo como trabajo final de grado, un receptor de tramas AIS de bajo coste, utilizando componentes baratos pero que cumplen a la perfección su cometido, recibir y decodificar tramas AIS.

Además, para realización del resto de este trabajo se ha tenido que investigar el estado actual del mercado en cuanto a barcos se refiere, y que cobertura tiene la costa valenciana. Se ha visto que hay ciertas zonas que no tienen cobertura de recepción AIS ya que, al no estar cerca de ningún puerto importante, no hay ningún receptor AIS.

Por otra parte, se ha profundizado en el uso de la aplicación Docker, se trata de una aplicación que a simple vista no parece que sea algo verdaderamente útil, pero tras su uso intensivo durante la realización de este trabajo he podido observar que una cosa tan crucial como es añadir y quitar, encender y apagar partes del sistema se vuelve una tarea mucho más fácil utilizando Docker ya que tienes todos los componentes “independientes”. O si simplemente quería empezar de cero basta con eliminar todo y en un comando tenía todo ejecutándose otra vez todo limpio.

El uso del ecosistema FIWARE me ha parecido algo realmente innovador, ya que permite reunir muchos datos desde diferentes fuentes con sus diferentes protocolos en un mismo lugar sin que haya ningún tipo de conflicto, ya que gracias a los agentes IoT todos los datos tienen un tratamiento personalizado, consiguiendo así la interoperabilidad entre sistemas IoT.

Finalmente, los servicios cognitivos parecen ser la clave para modernizar muchas aplicaciones y muchos servicios, ya que facilitan el uso de la inteligencia artificial y de las herramientas de autoaprendizaje, en este trabajo se han descrito muy brevemente tres posibles escenarios donde estos servicios cognitivos pueden llegar a tener utilidad.



## 5.2 Futuros desarrollos

Como futuros desarrollos entraría un infinito de posibilidades, ya que solo se ha diseñado un prototipo situado en el laboratorio de la UPV, donde ni la cobertura ni el lugar serían los óptimos. Como primer desarrollo pondría el despliegue de más receptores por la costa de la comunidad valenciana, permitiendo así uno de los propósitos de este trabajo, cubrir la zona costera con receptores AIS de bajo coste.

Uno de los otros puntos que se podrían mejorar es la funcionalidad de los receptores, ya que como bien indican, son receptores, solo reciben la información y no informan de nada vía AIS, es decir, no emiten señales de radio, con un poco más de trabajo e investigación se podría desarrollar de alguna forma que estos receptores sean en realidad transpondedores, y que confirmaran su situación.

Finalmente, la publicación de la plataforma sería también un buen desarrollo futuro, ya que esta no ha llegado a salir en ningún momento públicamente, para hacer eso se necesitaría un servidor público donde alojar la plataforma además de aumentar la seguridad de esta, ya que, aunque se haya asegurado algo, no todo está perfectamente seguro.



## Capítulo 6. Bibliografía

- [1] Azure Machine Learning. <https://azure.microsoft.com/es-es/services/machine-learning/>
- [2] Servicios Cognitivos Azure. <https://azure.microsoft.com/es-es/services/cognitive-services/>
- [3] AWS Machine Learning <https://aws.amazon.com/es/machine-learning/ai-services/>
- [4] AIS : Automatic Identification System = Sistema de Identificación Automática : teoría y práctica : el sistema de navegación y seguridad del futuroHirche, Rüdiger | Madrid : Tutor, D.L. 2010.
- [5] J. Song, K. Oh, I. Kim and J. Lee, "Application of maritime AIS (Automatic Identification System) to ADS-B (Automatic Dependent Surveillance — Broadcast) transceiver," ICCAS 2010, Gyeonggi-do, 2010, pp. 2233-2237, doi: 10.1109/ICCAS.2010.5669842.
- [6] Delás Castellá, H. (2018). Desarrollo de un terminal AIS lowcost para seguimiento de barcos. <http://hdl.handle.net/10251/108859>
- [7] F. Liang, W. Yu, D. An, Q. Yang, X. Fu and W. Zhao, "A Survey on Big Data Market: Pricing, Trading and Protection," in IEEE Access, vol. 6, pp. 15132-15154, 2018, doi: 10.1109/ACCESS.2018.2806881.
- [8] Dong Yang, Lingxiao Wu, Shuaian Wang, Haiying Jia & Kevin X. Li (2019) How big data enriches maritime research – a critical review of Automatic Identification System (AIS) data applications, Transport Reviews, 39:6, 755-773, DOI: 10.1080/01441647.2019.1649315
- [9] Pagina web oficial de Marinetraffic <https://www.marinetraffic.com/> [Online]
- [10] Pagina web oficial de Vesseltracker <https://www.vesseltracker.com/> [Online]
- [10] Documentación oficial de FIWARE Orion: <https://fiware-orion.readthedocs.io>
- [12] MongoDB Inc, What is MongoDB? <https://www.mongodb.com/what-is-mongodb> [Online]
- [13] Docker Inc., What is Docker, <https://www.docker.com/what-docker> [Online]
- [14] Docker Inc., What is a container, <https://www.docker.com/what-container> [Online]
- [15] Pagina web oficial de AISHub <https://www.aishub.net/> [Online]
- [16] <http://politicaterritorial.gva.es/es/web/puertos/puertos-cv/puertos-gv-p>
- [17] Documentación de pyngsi <https://pypi.org/project/pyngsi/>
- [18] <https://pixel-ports.github.io/pyngsi-tutorial.html>
- [19] <https://github.com/M0r13n/pyais>
- [20] Documentación oficial docker compose <https://docs.docker.com/compose/>
- [18] Documentación oficial fiware-ckan <https://fiware-ckan-extensions.readthedocs.io/en/latest/installation-administration-guide.html>
- [19] Pagina web oficial CKAN <https://ckan.org/>
- [20] Documentación oficial CKAN <https://docs.ckan.org/en/2.9/user-guide.html>