



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Técnicas para la simulación de un sistema autónomo de
vigilancia del Puerto de Marín*

Grado en Ingeniería Mecánica

ALUMNO: Pablo Munilla García

DIRECTORES: Norberto Fernández García
Guillermo Pérez Gobernado

CURSO ACADÉMICO: 2022-2023

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Técnicas para la simulación de un sistema autónomo de
vigilancia del Puerto de Marín*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Infantería de Marina

Universidade de Vigo

RESUMEN

Actualmente existen en el mercado distintos vehículos no tripulados de superficie (USV) remotamente controlados. Se están utilizando estos vehículos como apoyos en tareas de patrulla portuaria para prevenir acciones de guerra asimétrica, pero con las limitaciones del control remoto, que se traducen principalmente en tiempo de reacción. La implantación de un sistema de inteligencia artificial permitiría dotar a estos drones de una autonomía limitada, para ganar tiempo de reacción a la hora de llevar a cabo ciertas acciones, como aproximaciones, emisión de avisos sonoros o, incluso, la interceptación de potenciales amenazas.

El presente TFG se enfoca en el desarrollo de una serie de herramientas básicas que pueden ser empleadas en la implementación de un entorno de simulación que se podría aplicar al entrenamiento o aprendizaje por refuerzo de un sistema de vigilancia autónomo. El sistema desarrollado, diseñado para operar en el entorno concreto del Puerto de Marín, es capaz de analizar las velocidades de las embarcaciones que ingresan al puerto, detectar y alertar sobre posibles anomalías en dichas velocidades, así como de interceptar aquellas embarcaciones que presenten dichas anomalías. Además, se han desarrollado mecanismos que permiten representar gráficamente las trayectorias de aproximación a puerto e interceptación.

PALABRAS CLAVE

Dron, AIS, Velocidades, Anomalías, Embarcaciones, Simulación

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a mis dos tutores, Norberto y Guillermo, por toda la ayuda y conocimientos prestados, siempre disponibles en todo momento a pesar de las adversidades. Sin ellos, este trabajo no hubiera sido posible ya que han ejercido una impecable función a la hora de tutorizar este trabajo.

También me gustaría agradecer a todos mis compañeros de la promoción 423-153 por el incansable apoyo prestado de manera desinteresada y constante durante toda mi estancia en la Escuela Naval Militar, donde hemos vivido experiencias únicas e inolvidables.

A mi familia, por su disposición y ánimo incondicional en mis años de vida y por motivarme a seguir mejorando día a día, no importa la situación o el momento siempre dando la mejor versión de uno mismo.

Por último, quiero agradecer a todo el personal implicado en mi formación en la Escuela Naval Militar, han sido 7 años duros pero increíbles que no cambiaría por nada del mundo. En especial a mis Brigadieres que me enseñaron a no rendirme nunca. Muchas gracias.

CONTENIDO

Contenido	1
Índice de Figuras	3
Índice de Tablas.....	5
1 Introducción y objetivos	6
1.1 Contexto y motivación	6
1.2 Objetivos	9
1.3 Estructura	10
2 Estado del arte	11
2.1 Drones en el mundo militar.....	11
2.1.1 Aplicaciones y usos	11
2.1.2 Futuro.....	13
2.1.3 Ejemplos actuales de drones militares	14
2.2 Drones en la Armada Española	18
2.2.1 11ª Escuadrilla Aérea – Scan Eagle	18
2.2.2 Vigilancia portuaria	19
2.3 Inteligencia Artificial	20
2.3.1 Definición	20
2.3.2 Inteligencia Artificial en drones	21
2.4 Mecanismos de aprendizaje en Inteligencia Artificial	21
2.4.1 Aprendizaje supervisado.....	21
2.4.2 Aprendizaje no supervisado.....	23
2.4.3 Aprendizaje por refuerzo	24
2.5 Mecanismos de detección de anomalías	25
2.5.1 Monitoreo de sensores	25
2.5.2 Análisis de tráfico marítimo	25
2.5.3 Análisis de imágenes satelitales con AIS	26
2.5.4 Información meteorológica.....	26
2.5.5 Patrones de comportamiento.....	27
3 Desarrollo del TFG.....	28
3.1 Entorno de trabajo y software empleado.....	28
3.1.1 Tramas AIS	28
3.1.2 Lenguaje de programación Python	28
3.1.3 Open Street Map	29

3.1.4 Kepler.gl	29
3.2 Recopilación de datos AIS	30
3.3 Selección de datos de barcos.....	30
3.4 Selección de barcos con trayectorias de aproximación al puerto de Marín	32
3.5 Análisis estadístico de las velocidades de los barcos.....	33
3.6 Estimar las velocidades de los barcos a partir de las coordenadas GPS y el tiempo	34
3.7 Análisis estadísticos de las velocidades estimadas con el código <i>Python</i>	36
3.8 Código para detección de anomalías en las velocidades de los barcos.....	37
3.9 Código para la interceptación por el dron al barco que presenta anomalías	38
3.10 Código para convertir a formato <i>GeoJSON</i>	40
4 Prueba y validación	43
4.1 Estudio de las velocidades	43
4.2 Explicación con ejemplo.....	43
4.3 Representación y validación en <i>Kepler.gl</i>	45
5 Conclusiones y líneas futuras	47
5.1 Conclusiones	47
5.2 Líneas futuras	47
6 Bibliografía.....	49
Anexo I: Implicaciones Sociales, y/o Económicas, y/o Ambientales	54
Anexo II: Reflexiones Éticas y Sociales	55
Anexo III: Distribución de horas	56

ÍNDICE DE FIGURAS

Figura 1-1 Vehículo aéreo no tripulado (UAV) [30].....	6
Figura 1-2 Vehículo terrestre no tripulado (UGV) [66]	7
Figura 1-3 Vehículo marino no tripulado (USV) [31].....	7
Figura 1-4 Vehículo submarino no tripulado (UUV) [33]	8
Figura 1-5 Dron ucraniano atacando a un carro de combate ruso [66]	8
Figura 1-6 Dron Shahed-136 [36]	9
Figura 2-1 Soldado con un mini dron [5]	12
Figura 2-2 Esquema del sistema de enjambre de drones en un mapa [7].....	12
Figura 2-3 Dron Defender [37].....	13
Figura 2-4 Ejemplo de un sistema microondas en un dron [11]	14
Figura 2-5 Recreación digital del futuro EuroMALE [13].....	14
Figura 2-6 Dron <i>Bayraktar TB2</i> [35]	15
Figura 2-7 Dron ruso <i>Orlan-10</i> [16].....	15
Figura 2-8 Dron americano MQ-9 Reaper [46].....	16
Figura 2-9 Dron UGV ruso <i>Marker</i> [45].....	16
Figura 2-10 USVs <i>kamikaze</i> [49]	17
Figura 2-11 IAI <i>Harop</i> [21].....	17
Figura 2-12 Escudo 11ª Escuadrilla de Aeronaves [22].....	18
Figura 2-13 <i>Scan Eagle</i> despegando [23].....	19
Figura 2-14 Dron Airfox [47].....	19
Figura 2-15 USV <i>Vendaval</i> [26].....	20
Figura 2-16 Aprendizaje supervisado [33]	22
Figura 2-17 Aprendizaje no supervisado [34]	23
Figura 2-18 Aprendizaje por refuerzo [36]	24
Figura 2-19 Información en tiempo real por <i>Marine Traffic</i> [40].....	26
Figura 3-1 Coordenadas del Puerto de Marín y alrededores en <i>OpenStreetMap</i> (fuente: propia) ...	29
Figura 3-2 Representación de las trayectorias de distintos barcos en <i>kepler.gl</i> (fuente: propia)	30
Figura 3-3 Ejemplo de 20 tramas AIS en formato NMEA (fuente: propia).....	30
Figura 3-4 Código <i>Python</i> <code>parse_ais.py</code> (fuente: propia)	31
Figura 3-5 Código <i>Python</i> <code>calcula_traj.py</code> parte 1 (fuente: propia).....	32
Figura 3-6 Código <i>Python</i> <code>calcula_traj.py</code> parte 2 (fuente: propia).....	32
Figura 3-7 Ejemplo del formato CSV de un fichero <i>.traj</i> (fuente: propia).....	33
Figura 3-8 Gráfica de velocidades en <i>Excel</i> (fuente: propia)	34

Figura 3-9 <i>Boxplot</i> de velocidades en <i>Excel</i> (fuente: propia).....	34
Figura 3-10 Código <i>Python</i> estima_velo.py parte 1 (fuente: propia).....	35
Figura 3-11 Código <i>Python</i> estima_velo.py parte 2 (fuente: propia).....	35
Figura 3-12 Gráfica de velocidades estimadas en <i>Excel</i> (fuente: propia)	36
Figura 3-13 <i>Boxplot</i> de velocidades estimadas en <i>Excel</i> (fuente: propia).....	37
Figura 3-14 Código <i>Python</i> det_anom.py (fuente: propia)	37
Figura 3-15 Código <i>Python</i> intercept.py parte 1 (fuente: propia)	39
Figura 3-16 Código <i>Python</i> intercept.py parte 2 (fuente: propia)	39
Figura 3-17 Código <i>Python</i> intercept.py parte 3 (fuente: propia)	39
Figura 3-18 Código <i>Python</i> geojson_tray.py parte 1 (fuente: propia).....	41
Figura 3-19 Código <i>Python</i> geojson_tray.py parte 2 (fuente: propia).....	41
Figura 3-20 Código <i>Python</i> geojson_tray.py parte 3 (fuente: propia).....	41
Figura 3-21 Código <i>Python</i> geojson_tray.py parte 4 (fuente: propia).....	42
Figura 4-1 Ejemplo velocidades estimadas (fuente: propia)	44
Figura 4-2 Ejemplo velocidad anómala encontrada (fuente: propia)	44
Figura 4-3 Ejemplo archivo trayectoria interceptación dron (fuente: propia).....	45
Figura 4-4 Representación visual del dron saliendo del puerto (fuente: propia)	45
Figura 4-5 Representación visual del dron aproximandose al barco sospechoso (fuente: propia) ..	46
Figura 4-6 Representación visual del dron interceptando al barco sospechoso (fuente: propia)	46
Figura A-1 Gráfica distribución del tiempo del proyecto	56
Figura A-2 Diagrama Gantt distribución del tiempo del proyecto.....	56

ÍNDICE DE TABLAS

Tabla 4-1 Datos de las velocidades estimadas a partir de la posición (fuente: propia).....	43
--	----

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contexto y motivación

Nos encontramos en un mundo que cambia constantemente, uno de los cambios más importantes de los últimos años está siendo el empleo de los drones, tanto en el mundo militar como en el civil. Este trabajo se centrará principalmente en el uso de los drones en el mundo militar, asimismo, se tendrán en cuenta todas las consideraciones que repercuten en el mundo civil.

Un dron es un vehículo no tripulado que puede ser controlado de forma remota y utilizado para una variedad de tareas, como la vigilancia, fotografía aérea, entregas, inspección de infraestructura, agricultura, seguridad, etc. Pueden ser controlados de tres maneras distintas:

- Control manual: mediante un mando remoto o dispositivo móvil controlado por un operador humano.
- Control autónomo: en este modo el dron se controla mediante un software que le permite tomar decisiones autónomas basadas en la información que recibe a través de sus sensores.
- Control semi-autónomo: este modo combina los dos anteriores, es decir, se controla manualmente para el despegue y aterrizaje y para la toma de decisiones críticas, mientras que el control autónomo se utiliza para tareas de rutina [1].

Los drones pueden estar programados para volar automáticamente siguiendo una ruta preestablecida, incluir cámaras, sensores y otros equipos para recopilar datos y tener diferentes formas y tamaños, desde pequeños dispositivos personales hasta grandes sistemas industriales. El tipo de dron más conocido es el vehículo aéreo no tripulado UAV (*Unmanned Aerial Vehicle*, ver Figura 1-1).



Figura 1-1 Vehículo aéreo no tripulado (UAV) [30]

De igual manera, existen drones que se desplazan sobre la superficie terrestre o acuática en lugar de volar en el aire. A estos se les conoce como vehículos terrestres no tripulados UGV (*Unmanned Ground Vehicle*, ver Figura 1-2) o vehículos marinos no tripulados USV (*Unmanned Surface Vehicle*), concretamente el que vemos en la Figura 1-3 es el USV Sead 23, de la empresa Seadrone (con sede en Vigo), que ya ha colaborado con la Armada en varias ocasiones, entre ellas las 2 ediciones del ejercicio REPMUS y la primera edición de las Dynamic Messenger. Es un USV de control remoto con cierta capacidad semi-autónoma.



Figura 1-2 Vehículo terrestre no tripulado (UGV) [66]



Figura 1-3 Vehículo marino no tripulado (USV) [31]

Además, existen los vehículos submarinos no tripulados UUV (*Unmanned Underwater Vehicle*, ver Figura 1-4). Dichos drones también pueden ser utilizados para cumplir las funciones ya mencionadas anteriormente e incluso poseen características distintas tales como el movimiento mediante ruedas, orugas o propulsión acuática y la capacidad de navegar en terrenos difíciles o profundidades insospechadas. A menudo, estos drones incluyen de igual forma cámaras, sensores y otros equipos para recopilar datos.



Figura 1-4 Vehículo submarino no tripulado (UUV) [33]

Los drones pueden tener varios beneficios sociales, pueden ser utilizados para entregar paquetes y suministros médicos en áreas remotas o de difícil acceso, monitoreo y vigilancia de áreas naturales y culturales, control del tráfico y de multitudes, apoyo en situaciones de emergencia y rescate, con fines educativos y de investigación, para entretenimiento ya sea con fotografía o vídeos, etc [2].

En cuanto a los drones en el mundo militar, han revolucionado la manera en que se planean las estrategias y además el cómo ejecutarlas. Uno de los libros más importantes sobre este tema es *“Army of None: Autonomus Weapons and the Future of War”* [3] escrito por Paul Scharre que se centra en el uso de tecnologías autónomas en la guerra. El libro examina las implicaciones éticas y legales de los sistemas de armas autónomas, como los drones y discute cómo estas tecnologías están cambiando la forma en que se llevan a cabo las operaciones militares.

Los drones han tenido un impacto positivo significativo en el mundo militar, permitiendo una mayor eficiencia y seguridad en misiones de reconocimiento, vigilancia y ataque. Además, los drones permiten una mayor capacidad de recopilación de información y una reducción en la exposición de los soldados a situaciones peligrosas.

Algo que tenemos a la orden del día es el uso de los drones en la guerra de Ucrania (véase Figura 1-5), estos han proporcionado grandes ventajas ya que han permitido a las fuerzas armadas ucranianas monitorear y recopilar información sobre las actividades de las fuerzas rusas en el este de Ucrania. Esto ha sido útil para mejorar la toma de decisiones y la planificación de operaciones militares. Igualmente, los drones han servido de gran ayuda al ejército ruso ya que los han utilizado para corregir sus ataques con artillería y realizar ataques en enjambre con los UAV Geran-2/Shahed-136 (ver Figura 1-6).

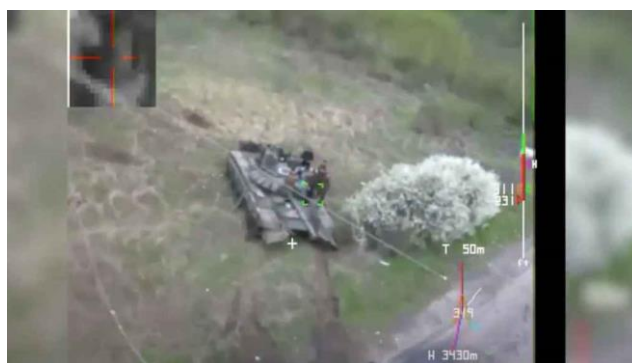


Figura 1-5 Dron ucraniano atacando a un carro de combate ruso [66]



Figura 1-6 Dron Shahed-136 [36]

Asimismo, otra guerra muy reciente en la que el impacto de los drones tuvo connotaciones decisivas ha sido en la Segunda guerra de Nagorno-Karabaj (27 sept 2020 - 10 nov 2020), tanto para la recopilación de información y reconocimiento, como en la conducción de ataques aéreos. Los drones permitieron a los combatientes obtener una ventaja táctica al proporcionar información en tiempo real y la capacidad de realizar ataques precisos y sigilosos.

Pasamos también a hablar de los drones autónomos, estos son objeto de gran interés debido a su capacidad para realizar tareas de manera eficiente y minimizando la intervención de un operador humano. Los drones autónomos ofrecen mayor eficiencia en la realización de misiones, reducción de costes, menor riesgo para las personas al poder realizar tareas en entornos peligrosos o inaccesibles, mejor seguimiento y sin necesidad de un operador humano que se encuentre próximo controlándolo, mayor supervivencia en entornos de guerra electrónica, al minimizar la necesidad de señales de control, y un menor tiempo de respuesta ante eventos.

Por último, los drones han permitido realizar misiones que antes requerían aviones o helicópteros de manera más económica, lo que ha reducido los costes de operación y mantenimiento de la flota militar. En resumen, la evolución de los drones en el mundo militar ha tenido un impacto positivo en la seguridad, eficiencia, capacidad de respuesta, inteligencia y reducción de costes de las operaciones militares.

1.2 Objetivos

El objetivo principal es desarrollar algunas técnicas básicas que puedan servir como primer paso a la hora de implementar un entorno simulado y controlado que pueda servir para entrenar, por ejemplo, mediante técnicas como el aprendizaje por refuerzo, una inteligencia artificial para el control de un dron de vigilancia de entornos portuarios. Este tipo de entornos simulados tienen una ventaja: se puede realizar el entrenamiento sin disponer de los medios físicos (el dron) y sin riesgo de accidentes reales, pero a cambio tienen una desventaja: para que el aprendizaje sea significativo es necesario que el entorno simulado sea realista.

En este trabajo se pretenden dar los primeros pasos para la implementación de ese entorno simulado, en un escenario muy acotado: el entorno geográfico se limita al Puerto de Marín, sólo habrá una embarcación "objetivo" y el dron (no hay terceros buques implicados) y, la tarea del dron se limitará a la interceptación de embarcaciones que presenten anomalías en su velocidad.

Los objetivos específicos que se van a perseguir en este trabajo para poder cumplir el objetivo principal son los siguientes:

- Análisis de datos reales de las derrotas y velocidades de los barcos en la zona.

- Estudio estadístico de dichas velocidades para modelar comportamientos.
- Detección de anomalías de las velocidades de los barcos.
- Definición de trayectorias de interceptación y seguimiento de barcos con comportamiento anómalo.
- Representación del mapa de la Ría de Pontevedra y del Puerto de Marín.
- Simulación de la derrota del barco sospechoso y la trayectoria de interceptación del dron.

1.3 Estructura

A continuación se exponen los diferentes capítulos que componen la estructura de esta memoria:

- Capítulo 1: este capítulo consta de la introducción del trabajo, en primer lugar con un contexto centrado principalmente en el mundo de los drones, asimismo se indica tanto el objetivo principal como los subobjetivos en los que este se apoyará. Por último, se plantea la organización de la memoria.
- Capítulo 2: en él se encuentra el estado del arte que principalmente revisa tanto los conceptos teóricos de importancia para la comprensión del trabajo, como las distintas herramientas, programas, tecnologías y aplicaciones necesarias para el desarrollo de dicho trabajo.
- Capítulo 3: se trata del desarrollo del TFG en el cual se describen los pasos que se han ido siguiendo en la realización del trabajo, explicando así el cumplimiento de los objetivos.
- Capítulo 4: se describen detalladamente los datos disponibles y cómo se ha llegado hasta los resultados obtenidos, a su vez se comprueba y se argumenta la validación de dichos resultados, adjuntando sus respectivas pruebas.
- Capítulo 5: este capítulo incluye las conclusiones obtenidas durante la realización del trabajo, así como las líneas futuras en su aplicación, nombrando los posibles aspectos a mejorar y las ventajas que pueden llegar a proporcionar.
- Finalmente, se adjunta la bibliografía con todas las fuentes y referencias que se han ido utilizando en el desarrollo de este trabajo, y los anexos de apoyo para el mismo.

2 ESTADO DEL ARTE

En este capítulo se trata de estudiar en primer lugar la importancia de los drones en el mundo militar, así como sus aplicaciones y usos más destacados. Asimismo se habla del futuro de los drones en el mundo militar y de los ejemplos más actuales de los drones militares. Posteriormente, se pone el foco en su uso dentro de la Armada española particularizando en la 11ª Escuadrilla Aérea y la vigilancia portuaria.

A continuación, se define la Inteligencia Artificial y se explica sus múltiples aplicaciones en drones para estudiar las ventajas que proporciona. Se exponen los mecanismos de entrenamiento de Inteligencia Artificial como son el aprendizaje supervisado, el no supervisado y el aprendizaje por refuerzo. Finalmente, se estudian los distintos tipos de detección de anomalías, explicando detalladamente los más relevantes para el desarrollo de este trabajo.

2.1 Drones en el mundo militar

2.1.1 Aplicaciones y usos

Los drones tienen una amplia variedad de aplicaciones y usos en el mundo militar. Estos drones se utilizan principalmente para misiones de reconocimiento, vigilancia e inteligencia, ataque y defensa.

Reconocimiento: los drones se utilizan para recopilar información sobre el terreno, las tropas y los objetivos en tiempo real. Debido a que los drones poseen una gran potencia para obtener imágenes detalladas, permiten a los militares tener una visión completa y específica de la situación antes de iniciar una operación [4]. Los drones también pueden equiparse con cámaras termográficas y de infrarrojos para detectar objetivos en la oscuridad o en la niebla.

Vigilancia e inteligencia: los drones también se utilizan para vigilar zonas de frontera, instalaciones militares, terrenos complicados y para monitorear la actividad en zonas de conflicto. Estos drones pueden operar durante largos períodos de tiempo y proporcionar información en tiempo real a las fuerzas militares en el campo. La importancia, utilidad y empleo de estos drones va en un aumento imparable y se han convertido en un elemento indispensable para este tipo de labores [5]. En estas misiones ISR (*Intelligence, Surveillance & Reconnaissance*) se están empleando drones cada vez más pequeños (ver Figura 2-1) lo que dificulta que sean detectados por el enemigo y proporciona una gran ventaja a la hora de la obtención de información.



Figura 2-1 Soldado con un mini dron [5]

Ataque: algunos drones están equipados con armamento y se utilizan para realizar ataques aéreos contra objetivos enemigos. Estos drones pueden lanzar misiles guiados y realizar ataques con precisión, reduciendo el riesgo para los soldados en el campo. En este tipo de operaciones los drones son de mayor tamaño ya que cuentan con capacidad de portar misiles o pequeñas ametralladoras que puedan causar graves daños en el enemigo. También son capaces de transportar algunos tipos de bombas y lanzarlas en pleno vuelo contra vehículos ligeros y carros enemigos. Asimismo, está aumentando el uso de los llamados drones “kamikaze” que se lanzan contra un objetivo al que se pretende dañar y pasar a quedar inutilizado [6], otro modo de empleo también conocido es el de usarlos a modo de enjambre (ver Figura 2-2), es decir, un gran número de drones con un objetivo conjunto con el fin de dirigir ofensivas aéreas.

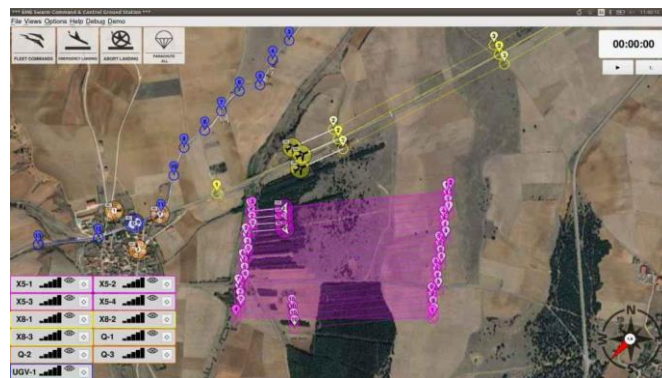


Figura 2-2 Esquema del sistema de enjambre de drones en un mapa [7].

Defensa: los drones también se utilizan para la defensa aérea, detectando y derribando a otros drones enemigos. Los drones de defensa pueden equiparse con sistemas para interceptar y derribar drones enemigos antes de que puedan realizar un ataque. Además, se pueden utilizar los ya mencionados enjambres para proporcionar la defensa de un área determinada. Algunas fuentes aseguran incluso que “*un avispero de drones sería la mejor defensa de Taiwán ante un repentino ataque chino*” [7].

Además de estos usos, los drones también se utilizan para misiones humanitarias, como la entrega de suministros médicos y alimentos en áreas remotas o afectadas por desastres naturales. También se utilizan en la búsqueda y rescate, y en la monitorización de incendios forestales, que es de gran ayuda para la Unidad Militar de Emergencias (UME), esta unidad está adquiriendo cada vez más drones para mejorar sus capacidades y proporcionar una mayor protección en las operaciones [8].

En conclusión, los drones tienen una amplia variedad de usos y aplicaciones en el mundo militar y han cambiado la forma en que las fuerzas militares realizan misiones. A medida que la tecnología avanza, se espera que los drones jueguen un papel cada vez más importante en el futuro de las operaciones militares.

2.1.2 Futuro

El futuro de los drones militares se caracterizará por un aumento en su capacidad tecnológica, una mayor autonomía y una integración cada vez más estrecha con otros sistemas militares.

En términos de capacidad tecnológica, los drones militares seguirán mejorando con el tiempo, esto incluirá una mayor capacidad de vuelo autónomo, una mejora en las capacidades de inteligencia artificial y una mayor capacidad de comunicación. Además, se espera que los drones militares sean cada vez más capaces de mantenerse en el aire durante períodos más prolongados, lo que los hace más útiles para misiones militares [9].

La integración con otros sistemas militares será una parte importante del futuro de los drones militares. Esto incluirá la integración con sistemas de armas, sistemas de defensa aérea y sistemas de información y seguimiento. Este tipo de integración permitirá una mayor eficacia en el uso de los drones militares en misiones, así como una mejor protección para las fuerzas militares en el campo.

El uso de drones militares seguirá aumentando en misiones de inteligencia, vigilancia y reconocimiento, así como en misiones de ataque selectivo. Sin embargo, también es probable que se utilicen drones no tripulados en misiones humanitarias y de ayuda [10]. Esto permitirá a las fuerzas militares llegar a lugares inaccesibles o peligrosos de manera más segura y eficiente.

La competencia internacional por el desarrollo y la producción de drones militares continuará siendo fuerte, con países tratando de mejorar sus capacidades militares mediante la adquisición de tecnología avanzada de drones. Esto significará una mayor inversión en la investigación y el desarrollo de drones militares, lo que conduce a un futuro más seguro y más avanzado para las fuerzas militares de todo el mundo.

Dada la previsible importancia de los drones en los conflictos futuros es de esperar el desarrollo de tecnologías que permitan denegar el uso de este tipo de sistemas al adversario. Actualmente, la Armada cuenta con el sistema Drone Defender (ver Figura 2-3) que funciona emitiendo señales de radio en la misma frecuencia utilizada por el control remoto del dron, lo que interfiere con la capacidad del dron para comunicarse con su operador. Cabe destacar que este tipo de sistemas tendrían menor impacto para contrarrestar la amenaza de drones autónomos, pues estos tienen una menor dependencia del control humano.



Figura 2-3 Drone Defender [37]

Además, algunos ejércitos como el estadounidense están trabajando en el desarrollo de sistemas microondas aplicables a drones militares que puedan ser utilizados para derribar otros drones y así poder evitar la amenaza de los drones enemigos [11], este sistema estaría colocado en la parte inferior de los drones (ver Figura 2-4) y no sólo estaría capacitado para derribar un dron sino que tendría la capacidad de derribar a todo un enjambre de drones.



Figura 2-4 Ejemplo de un sistema microondas en un dron [11]

Otra forma de hablar del futuro de los drones en el ámbito militar sería centrándonos en el dron EuroMALE (*Medium Altitude Long Endurance*) [12] como podemos ver en la Figura 2-5, es un dron de vigilancia y reconocimiento militar desarrollado por empresas europeas, liderado por Airbus Defence and Space. Fue diseñado para cumplir con las necesidades de las fuerzas armadas europeas en términos de capacidades de vigilancia y reconocimiento a larga distancia. El EuroMALE cuenta con una amplia gama de sensores y tecnologías avanzadas, como cámaras de alta resolución y radar, que le permiten realizar misiones de vigilancia y reconocimiento de manera eficiente y precisa. Es uno de los drones militares más avanzados y versátiles que estará disponible en el mercado.



Figura 2-5 Recreación digital del futuro EuroMALE [13]

Su principal finalidad será la de realizar misiones ISTAR (*Intelligence, Surveillance, Target Acquisition and Reconnaissance*), para ello deberá ser capaz de volar a cotas medias y durante periodos de 24 horas. El Ministerio de Defensa español ya ha presupuestado una inversión de unos 1.900 millones de euros para el desarrollo de este dron [13].

Los países implicados en este proyecto son: España, Alemania, Francia e Italia. Esta aeronave no tripulada tendrá 26 metros de envergadura, 16 metros de longitud y 6 metros de altura. El primer vuelo está previsto para 2025 y se prevé la producción de estos drones en 2034. El EuroMALE será capaz de alcanzar los 500km/h y podrá alcanzar cerca de los 14.000 metros de altura [14].

2.1.3 Ejemplos actuales de drones militares

El principal dron militar de Ucrania es el *Bayraktar TB2* (ver Figura 2-6), es un dron de fabricación turca desarrollado por Baykar Makina. Es un UAV de alta tecnología utilizado para recopilar información y realizar misiones de combate. Ucrania comenzó la guerra con una flota de 50 de estos drones [15].

Entre sus características destacan un alcance de vuelo de hasta 200 km, una capacidad de carga de armamento de hasta 55 kg y la capacidad de operar de manera autónoma o controlado remotamente,

Tiene unos 6.5 metros de longitud y 12 metros de envergadura y puede volar hasta los 8.200 metros de altitud con una autonomía de 27 horas [16]. Además, cuenta con un sistema avanzado de navegación y una tecnología de transmisión de datos en tiempo real. El *Bayraktar TB2* ha sido ampliamente utilizado por las fuerzas armadas turcas en su conflicto en Siria y también ha sido exportado a otros países, incluido Azerbaiyán donde fue empleado con éxito en la guerra de Nagorno-Karabaj, además de Ucrania como ya se ha mencionado anteriormente.



Figura 2-6 Dron *Bayraktar TB2* [35]

En cuanto al bando ruso, utiliza principalmente el dron *Orlan-10* (ver Figura 2-7), es un dron diseñado para uso militar y civil, de alta tecnología que se utiliza para recopilar información sobre el terreno y realizar tareas de vigilancia y reconocimiento. Posee una capacidad de vuelo de hasta 10 horas y puede operar en una amplia gama de condiciones climáticas. La envergadura es de 3 metros, mientras que el fuselaje alcanza los 2 metros [17]. Además, cuenta con un sistema de navegación GPS y capacidad para alojar sistemas destinados a provocar interferencias en los sistemas del enemigo. En resumen, el *Orlan-10* es un dron versátil y efectivo que se utiliza para una variedad de aplicaciones.



Figura 2-7 Dron ruso *Orlan-10* [16]

Además, han sido especialmente eficaces los Geran-2/Sahed-136, ya mencionados anteriormente, que han sido utilizados como drones kamikazes en enjambres y han impactado en infraestructuras críticas de casi toda Ucrania (sobre todo Kiev y Odessa). Actualmente se sospecha que Rusia ha adquirido o está a punto de adquirir el Arash-2, con mayor radio de acción y mayor capacidad destructiva.

Por otro lado, el dron *MQ-9 Reaper* (ver Figura 2-8), también conocido como *Predator*, ha desempeñado un papel fundamental como parte del ejército americano en Afganistán, tanto con labores de reconocimiento como de ataques a objetivos.



Figura 2-8 Dron americano MQ-9 Reaper [46]

En cuanto a los UGVs, pueden ser utilizados para una variedad de tareas militares, incluyendo reconocimiento, vigilancia, adquisición de objetivos y eliminación de explosivos. Los UGVs son una herramienta valiosa para las fuerzas militares, ya que permiten a los soldados realizar misiones peligrosas desde una distancia segura.

En el campo de batalla, los UGVs pueden ser equipados con una variedad de sensores, cámaras y armas para realizar diferentes tareas. Por ejemplo, los UGVs equipados con cámaras y sensores pueden ser utilizados para detectar y localizar objetivos enemigos, mientras que los UGVs equipados con explosivos pueden ser utilizados para eliminar artefactos explosivos improvisados (IED) en zonas peligrosas [18].

Además, los UGVs pueden ser utilizados para apoyar a las fuerzas militares en tareas logísticas, como el transporte de suministros y equipo, lo que puede mejorar la eficiencia y la seguridad de las operaciones militares.

El ejército ruso está usando este tipo de drones en la actual guerra contra Ucrania, en concreto el UGV *Marker* (ver Figura 2-9) que podría ser capaz de detectar y atacar automáticamente carros de combate *Abrams* y *Leopard* gracias al catálogo electrónico del sistema de control con imágenes de vehículos enemigos [19].



Figura 2-9 Dron UGV ruso Marker [45]

Por otra parte, se ha informado que Ucrania ha utilizado vehículos submarinos no tripulados (UUV) para la detección y eliminación de minas marinas. Estos vehículos pueden ser controlados a distancia y equipados con cámaras, sensores y brazos robóticos para inspeccionar y desactivar minas en áreas peligrosas sin poner en riesgo a los equipos de desactivación de explosivos.

De hecho, estos drones de superficie están cobrando bastante importancia en la batalla del Dnieper, con los pequeños asentamientos que tienen ambos ejércitos, y las pequeñas lanchas que patrullan la zona, tener USV con capacidades se está convirtiendo en algo extremadamente importante.

Asimismo, se han confirmado distintos ataques sorpresa contra la Flota del Mar Negro mediante el uso de USVs (ver Figura 2-10) como drones suicida, que realizaron ataques *kamikaze* contra los barcos rusos. Ucrania, al no tener flota que pueda contrarrestar a la rusa está usando estos drones para obtener información y para atacar buques rusos incluyendo la Almirante *Makarov* [20].



Figura 2-10 USVs *kamikaze* [49]

Finalmente, destacar el concepto de munición merodeadora (*loitering munitions*) que son drones de ataque que tienen la capacidad de permanecer en el aire durante períodos prolongados de tiempo (*loitering*) antes de atacar un objetivo enemigo. Estos drones de ataque son a menudo llamados "drones *kamikaze*" debido a su capacidad de volar hacia un objetivo y explotar en contacto.

Las municiones merodeadoras son capaces de operar en una variedad de entornos, incluyendo áreas urbanas, montañas y entornos marinos. Estos drones de ataque son relativamente pequeños y portátiles, lo que los hace fáciles de desplegar y operar [21].

Una de las más conocidas es el IAI *Harop* (ver Figura 2-11), un tipo de munición merodeadora desarrollado por la compañía israelí Israel Aerospace Industries (IAI). El *Harop* es capaz de detectar objetivos mediante un radar y cámaras de alta resolución. También tiene un sistema de identificación de objetivos que le permite distinguir entre objetivos amigos y enemigos. Una vez que se identifica un objetivo enemigo, el *Harop* se lanza a sí mismo hacia el objetivo y explota en contacto [22].



Figura 2-11 IAI *Harop* [21]

El *Harop* ha sido utilizado por el ejército israelí en conflictos como la guerra de Gaza en 2014, así como por otros países en operaciones militares. Ha demostrado ser eficaz en la identificación y destrucción de objetivos, especialmente en áreas urbanas y en entornos donde las fuerzas enemigas se

esconden en estructuras o están en movimiento constante. Sin embargo, también ha habido preocupaciones sobre su precisión y la posibilidad de daños colaterales.

2.2 Drones en la Armada Española

2.2.1 11ª Escuadrilla Aérea – Scan Eagle

La 11ª Escuadrilla Aérea (ver Figura 2-12) se encarga de los sistemas aéreos no tripulados, tiene asignados los *Scan Eagle* (ver Figura 2-13) los cuales están controlados de forma remota, así como los blancos aéreos teledirigidos denominados *SCRAB-II*. El *Scan Eagle* está diseñado y fabricado por la compañía norteamericana INSITU y puede ser operado desde tierra o desde buques en la mar [23]. Esta unidad es la más moderna de la Flotilla de Aeronaves (FLOAN).



Figura 2-12 Escudo 11ª Escuadrilla de Aeronaves [22]

Estos sistemas son los primeros de su tipo en servicio en la Armada, teniendo como principales cometidos la vigilancia, la obtención de información y la obtención de imágenes en tiempo real en zonas de interés. Otro cometido de estos sistemas es el adiestramiento de las unidades navales en el tiro antiaéreo, para ello se generan unidades aéreas que se embarcan en los buques de la Armada Española para participar en los distintos ejercicios programados, así como en numerosas operaciones [24].

Los cometidos de esta Escuadrilla con el *Scan Eagle* son los de ISR (Identificación, Vigilancia y Reconocimiento de objetivos de interés), la vigilancia marítima, el apoyo a las unidades de Operaciones Especiales, las operaciones de búsqueda y rescate y el apoyo a entidades civiles (control de fronteras, vigilancia de ferrocarriles, detección de incendios, etc).

La Escuadrilla se encuentra ubicada en las Instalaciones de la Flotilla de Aeronaves en Torregorda (Cádiz). Esta ubicación permite un acceso fácil y rápido a las áreas de adiestramiento y al Estrecho de Gibraltar.

La plantilla actual de la 11ª Escuadrilla está formada por 68 profesionales.

- 1 Comandante Jefe de Escuadrilla (Piloto/DUO).
- 8 Oficiales (8 Pilotos/DUO).
- 24 Suboficiales (7 Pilotos/DUO).
- 9 Cabos 1º.
- 26 Cabos y Marineros.

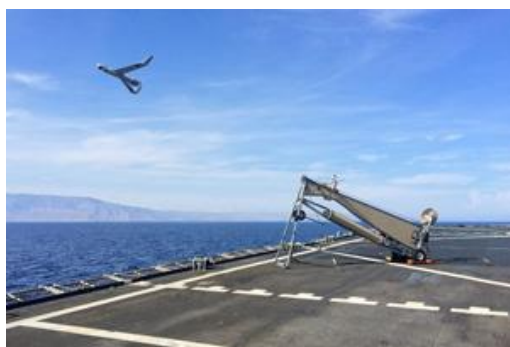


Figura 2-13 Scan Eagle despegando [23]

Las principales operaciones y ejercicios en los que ha participado esta Escuadrilla han sido:

- 2015, 2016 y 2017: Operación Atalanta.
- 2017 – 2019: Operación “Inherent Resolve” en Irak.
- 2020 - Actualidad: Operación Atalanta.

También participa en ejercicios nacionales e internacionales, como “Dinamic Mariner”, “Gruflex” o “Unified Vision” [23].

Recientemente, la Armada acaba de comprar 3 unidades del UAV Airfox (ver Figura 2-14), de la empresa Marine Instruments, que se trata de un UAV de pequeño tamaño, pero toda su superficie alar son paneles solares que le permiten volar con buenas condiciones solares más de 12 horas. Todo apunta a que quedarán en manos de la FMCM (Fuerza de Medidas Contra Minas).



Figura 2-14 Dron Airfox [47]

2.2.2 Vigilancia portuaria

Las aplicaciones de los drones de la Armada Española en vigilancia portuaria pueden incluir:

- Mejora de la seguridad marítima: pueden ser utilizados para mejorar la seguridad marítima en puertos y zonas costeras, detectando la presencia de peligros potenciales como objetos flotantes y restos de naufragios.
- Protección de los recursos naturales: los drones pueden ser utilizados para vigilar la actividad humana en zonas costeras y proteger los recursos naturales, como áreas marinas protegidas [25].
- Inspecciones de estructuras portuarias: los drones pueden ser utilizados para realizar inspecciones visuales de puentes, muelles y otras estructuras portuarias para determinar su estado y detectar posibles problemas.

- Gestión del tráfico marítimo: pueden ser utilizados para apoyar la planificación y la gestión del tráfico marítimo en puertos y zonas costeras, monitoreando la ubicación de barcos y ayudando a evitar colisiones.

Destacar también la labor del USV *Vendaval* (ver Figura 2-15), que opera en el puerto de Ceuta para la Autoridad Portuaria y la Armada Española realizando labores de protección de puertos tanto civiles como militares y que ha llevado a cabo distintos simulacros de manera exitosa [26].



Figura 2-15 USV *Vendaval* [26]

Finalmente, hacer hincapié en la diferencia entre el mundo de los UAV y el resto de UXV, ya que no existe legislación para este tipo de vehículos en casi ningún país, por ejemplo, el USV *Vendaval*, aunque es semi-autónomo, no puede operar solo debido a que no está legislado, por lo que siempre tiene que llevar un patrón de la embarcación, aunque no realice ninguna función específica.

2.3 Inteligencia Artificial

2.3.1 Definición

La Inteligencia Artificial (IA) es una rama de la informática y la ingeniería que se esfuerza por desarrollar sistemas y algoritmos que permitan a las máquinas realizar tareas que requieren inteligencia humana. Estas tareas incluyen la capacidad de aprender, razonar, planificar, entender lenguaje natural, reconocer imágenes y objetos, tomar decisiones y solucionar problemas [27].

La IA se basa en el estudio de la inteligencia humana y su representación en forma de modelos matemáticos y algoritmos computacionales. Estos modelos se entrenan con grandes cantidades de datos y, a medida que se procesan más datos, se van perfeccionando y mejorando su capacidad para realizar tareas específicas.

Hay varios ámbitos de trabajo dentro de la IA, incluyendo el aprendizaje automático, el procesamiento del lenguaje natural y la robótica. El procesamiento del lenguaje natural se centra en la capacidad de las máquinas para comprender y generar lenguaje humano. La robótica combina la IA con el hardware físico para crear robots que puedan realizar tareas complejas en un entorno físico.

La IA tiene un impacto significativo en muchas industrias, incluyendo la banca, la seguridad nacional, la salud, la automoción y el *retail*. Por ejemplo, los sistemas de IA están siendo utilizados para detectar fraudes bancarios, mejorar la eficiencia energética en las fábricas, y ayudar a los médicos a diagnosticar enfermedades [27].

El objetivo de la IA es crear sistemas que puedan aprender y mejorar a lo largo del tiempo, sin la necesidad de programación explícita. Además, la IA está transformando la forma en que las empresas operan y se relacionan con sus clientes. Las tecnologías de IA están ayudando a las empresas a automatizar tareas repetitivas, a mejorar la eficiencia y a tomar decisiones más informadas. Asimismo, están permitiendo a las empresas ofrecer una experiencia más personalizada y empática a sus clientes, lo que a su vez mejora la satisfacción del cliente y fomenta la lealtad.

En conclusión, la IA es una tecnología en constante evolución que está transformando la forma en que vivimos y trabajamos. Aunque existen preocupaciones acerca de sus efectos negativos, también hay muchas oportunidades para que la IA mejore la vida humana y fomente una sociedad más justa y equitativa. Es importante abordar las preocupaciones éticas y de privacidad a medida que se desarrolla y aplica la IA, para asegurarse de que sus beneficios y avances sean utilizados de manera responsable y sostenible [28].

2.3.2 Inteligencia Artificial en drones

La Inteligencia Artificial en drones se refiere al uso de técnicas de inteligencia artificial para mejorar la eficiencia, efectividad y seguridad de los drones. La IA está transformando la forma en que operan los drones, y su adopción está creciendo rápidamente en una amplia variedad de aplicaciones. Desde la exploración y la vigilancia hasta la entrega y la agricultura, los drones con IA están revolucionando la forma en que se llevan a cabo muchas tareas [29].

Uno de los mayores avances en la tecnología de IA en drones ha sido el desarrollo de algoritmos de aprendizaje profundo, también conocido como *Deep Learning* [30]. Estos algoritmos permiten a los drones analizar y procesar grandes cantidades de datos, lo que les facilita tomar decisiones informadas y reaccionar a su entorno de manera autónoma. Por ejemplo, los drones equipados con sensores pueden utilizar algoritmos de aprendizaje profundo para detectar y evitar obstáculos en su camino, lo que mejora su seguridad y eficiencia.

Otra aplicación de la IA en drones es la navegación autónoma. Los drones pueden ser programados para seguir rutas específicas sin intervención humana. Esto se logra mediante el uso de algoritmos de IA para planificar y optimizar rutas, y para monitorear el entorno en tiempo real para asegurarse de que los drones estén en el curso correcto [31]. La IA es la clave para dar el salto al dron completamente autónomo.

Además, los drones equipados con cámaras pueden utilizar algoritmos de IA para analizar y clasificar imágenes y vídeos en tiempo real. Esta tecnología se utiliza en una amplia variedad de aplicaciones, incluyendo la vigilancia y la topografía. Los drones con IA también pueden ser programados para seguir y monitorear objetivos específicos, como personas o vehículos [32].

En el futuro, es probable que la IA en drones se utilice cada vez más en una amplia variedad de aplicaciones. Además, es probable que la tecnología de IA en drones se integre cada vez más con otras tecnologías, como la robótica y el Internet de las cosas, para ofrecer soluciones más eficientes y avanzadas [33].

En resumen, la IA en drones está revolucionando la forma en que se llevan a cabo muchas tareas y su adopción está creciendo rápidamente. A medida que la tecnología continúa evolucionando, es probable que veamos un aumento en el uso de drones con IA en una amplia variedad de aplicaciones. Sin embargo, también es importante abordar los desafíos y preocupaciones relacionados con la seguridad, la privacidad y la regulación para garantizar un futuro seguro y sostenible para esta tecnología.

2.4 Mecanismos de aprendizaje en Inteligencia Artificial

2.4.1 Aprendizaje supervisado

El aprendizaje supervisado es uno de los mecanismos más utilizados en el campo de la Inteligencia Artificial. En este tipo de aprendizaje, el modelo es entrenado en base a datos etiquetados, es decir, datos que tienen una etiqueta o valor objetivo asociado, luego se evalúa el desempeño del modelo con un conjunto de datos de prueba no visto previamente, y se ajustan los parámetros internos del modelo hasta que alcance un nivel de precisión deseado (ver Figura 2-16) [34]. El objetivo es que el modelo pueda predecir el valor objetivo para nuevos datos a partir de los patrones aprendidos durante el entrenamiento.

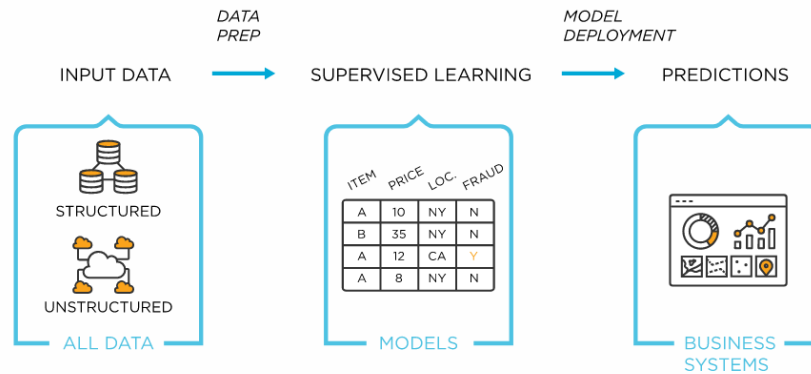


Figura 2-16 Aprendizaje supervisado [33]

A continuación, se describen algunas de las características y ventajas más relevantes del aprendizaje supervisado:

- Buena precisión: debido a que el modelo es entrenado con datos etiquetados, es más probable que produzca resultados precisos en comparación con otros tipos de aprendizaje.
- Gran variedad de aplicaciones: el aprendizaje supervisado se utiliza en una amplia gama de aplicaciones, desde la clasificación de imágenes hasta la predicción de valores numéricos.
- Fácil de implementar: la mayoría de las bibliotecas de aprendizaje automático incluyen algoritmos de aprendizaje supervisado, lo que facilita su implementación y uso.

Los pasos básicos del aprendizaje supervisado son los siguientes:

1. Preparación de los datos: se recopilan y limpian los datos relevantes para el problema específico y se dividen en un conjunto de entrenamiento y un conjunto de prueba.
2. Selección de un modelo: se selecciona un modelo de aprendizaje supervisado adecuado para el problema y se definen sus parámetros.
3. Entrenamiento del modelo: se utiliza el conjunto de entrenamiento para entrenar el modelo, ajustando sus parámetros internos para minimizar una función de pérdida.
4. Evaluación del modelo: se evalúa el desempeño del modelo en el conjunto de prueba, utilizando una métrica apropiada como la precisión.
5. Ajuste fino del modelo: si el desempeño en el conjunto de prueba no es satisfactorio, se pueden realizar ajustes adicionales en los parámetros o en el modelo en sí para mejorar su rendimiento.
6. Implementación y uso: finalmente, se implementa el modelo entrenado en un sistema real y se utiliza para hacer predicciones en situaciones futuras.

Optimizar y volver a entrenar el modelo: la evolución de datos es una parte natural del aprendizaje máquina (*Machine Learning*). Por lo tanto, los modelos se deberán volver a entrenar periódicamente con datos actualizados para garantizar la precisión [34].

Los dos tipos principales de aprendizaje supervisado son:

- Regresión: en este tipo de aprendizaje supervisado, el objetivo es predecir una variable continua o valor numérico, como el precio de una casa o la probabilidad de un evento. Los modelos de regresión más comunes incluyen la regresión lineal y la regresión logística.
- Clasificación: en este tipo de aprendizaje supervisado, el objetivo es predecir una variable discreta o categórica, como la etiqueta de una imagen o la clase a la que pertenece un objeto. Los modelos de clasificación más comunes incluyen el árbol de decisión, la máquina de vectores de soporte y la red neuronal.

En resumen, el aprendizaje supervisado es una herramienta poderosa en el desarrollo de sistemas de inteligencia artificial y es ampliamente utilizado en una diversa gama de aplicaciones prácticas.

2.4.2 Aprendizaje no supervisado

El aprendizaje no supervisado en IA es un enfoque en el que un modelo de inteligencia artificial es entrenado con un conjunto de datos sin etiquetas, es decir, sin una salida objetivo específica. En lugar de aprender a realizar una tarea específica, el objetivo del aprendizaje no supervisado es descubrir patrones o estructuras ocultas en los datos [35].

En este proceso, el modelo es alimentado con datos sin etiquetar y utiliza un algoritmo para detectar patrones y características en los datos. Algunos de los algoritmos más comunes de aprendizaje no supervisado incluyen el agrupamiento y la reducción de dimensionalidad (ver Figura 2-17):

- Algoritmos de agrupamiento (*clustering*): los algoritmos de agrupamiento se utilizan para estructurar los datos en grupos similares.
- Reducción de dimensionalidad: la reducción de dimensionalidad es una técnica utilizada en el aprendizaje no supervisado para comprimir los datos y hacerlos más fáciles de analizar.

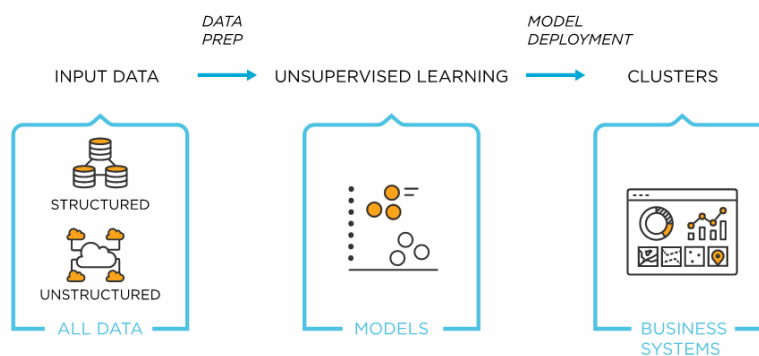


Figura 2-17 Aprendizaje no supervisado [34]

Ejemplos de tareas que pueden ser realizadas con aprendizaje no supervisado incluyen la segmentación de clientes en grupos, la detección de anomalías en datos de seguridad, y la compresión de datos de imágenes [35].

Las características del aprendizaje no supervisado son las siguientes:

- Sin dirección objetivo: a diferencia del aprendizaje supervisado, en el aprendizaje no supervisado no hay una dirección objetivo clara o una característica a aprender.
- Visualización de datos: la visualización de datos ayuda a representar gráficamente los patrones descubiertos en los datos.
- Exploración de datos: el aprendizaje no supervisado es una herramienta valiosa para la exploración de datos y la identificación de patrones y relaciones ocultas en los datos.

Los pasos básicos del aprendizaje no supervisado son idénticos a los del aprendizaje supervisado pero sin la parte del entrenamiento.

Los tipos principales de aprendizaje no supervisado son:

- Descubrimiento de asociaciones: consiste en identificar patrones o relaciones entre variables en un conjunto de datos. La idea es encontrar variables que tienden a aparecer juntas o que están relacionadas de alguna manera.
- Descubrimiento de reglas: consiste en encontrar todas las reglas de asociación significativas en un conjunto de datos. Estas reglas pueden ayudar a entender la relación entre diferentes variables y pueden ser utilizadas para hacer predicciones o tomar decisiones informadas [36].
- Análisis de componentes principales: se utiliza para reducir la dimensionalidad de un conjunto de datos. La idea es encontrar las variables que más contribuyen a la variabilidad en el conjunto de datos y utilizarlas para construir nuevas variables que expliquen la mayor cantidad de variabilidad posible.

2.4.3 Aprendizaje por refuerzo

El aprendizaje por refuerzo es un tipo de aprendizaje automático en el que un agente actúa en un entorno para maximizar una recompensa acumulada. La idea es que el agente aprende a realizar acciones que maximizan la recompensa a lo largo del tiempo, a partir de la retroalimentación de la recompensa que recibe por cada acción que realiza (ver Figura 2-18). Este proceso se basa en la idea de que el aprendizaje se produce a través de la experimentación y la repetición [37].

Este tipo de aprendizaje consta de dos componentes:

- Agente: será el modelo que queremos entrenar para que aprenda a tomar decisiones.
- Ambiente: será el entorno donde el agente interactúa. El ambiente contiene las limitaciones y reglas posibles en cada momento.

Entre ellos existe una relación que viene determinada por:

- Acción: las posibles acciones que puede tomar en un momento determinado el Agente.
- Estado: son los indicadores del ambiente de cómo están los diversos elementos que lo componen en ese momento.
- Recompensas: a raíz de cada acción tomada por el Agente, podremos obtener un premio o una penalización que orientarán al Agente sobre si lo está haciendo bien o mal.

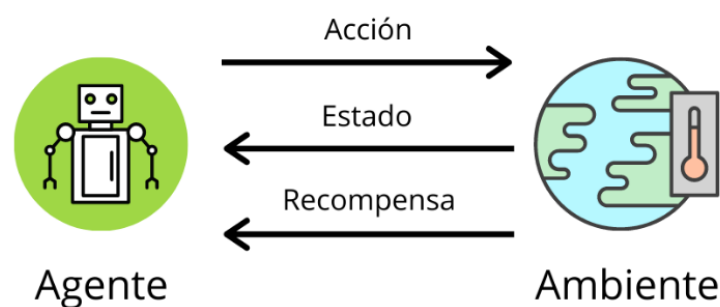


Figura 2-18 Aprendizaje por refuerzo [36]

La clave del entrenamiento de este tipo de máquinas reside en la experiencia, no trabaja con datos etiquetados como hemos visto en otras técnicas de aprendizaje máquina. Para que la experiencia funcione, es necesario que exista una retroalimentación fluida entre el agente y el entorno.

2.5 Mecanismos de detección de anomalías

2.5.1 Monitoreo de sensores

El monitoreo de sensores puede ser utilizado para detectar anomalías en los movimientos de los barcos en el ambiente marítimo. Los sistemas de monitoreo de sensores pueden recopilar datos de varios sensores, incluyendo sistemas de monitoreo satelital y de radar, y utilizarlos para detectar patrones inusuales en el movimiento de los barcos [38].

Para llevar a cabo este monitoreo, se utilizan diferentes sensores que miden el movimiento, la velocidad y la dirección del barco en tiempo real. Estos sensores pueden estar integrados en el equipo de navegación del barco o instalados en dispositivos de monitoreo externos.

La información recolectada por los sensores se procesa y analiza para detectar posibles anomalías en los movimientos del barco. Por ejemplo, si el barco está experimentando movimientos bruscos o inusuales, esto puede indicar que está navegando en condiciones de mar difíciles o que ha sufrido algún daño en su estructura.

El monitoreo de sensores también puede ayudar a detectar comportamientos anormales de los operadores del barco, como maniobras bruscas o exceso de velocidad, lo que puede ser indicativo de negligencia o conducta imprudente. También pueden ser utilizados para gestionar y controlar las embarcaciones en puerto y alertar al propietario en caso de una salida no programada [39] .

En resumen, el monitoreo de sensores es una herramienta valiosa para detectar anomalías en el movimiento de los barcos en el ambiente marítimo. La recopilación de datos de múltiples sensores y su análisis en tiempo real puede ayudar a prevenir actividades ilegales o peligrosas

2.5.2 Análisis de tráfico marítimo

El análisis de datos de tráfico marítimo es un proceso que consiste en extraer y analizar información sobre la actividad de los barcos en los océanos y mares del mundo. Se utiliza para entender mejor el comportamiento del tráfico marítimo y detectar patrones inusuales o anormales en la actividad de los barcos.

Para realizar el análisis, se utilizan datos de seguimiento de embarcaciones, registros de puertos y otros datos de navegación. Estos datos se procesan y se transforman en información útil para monitorear la actividad marítima y detectar patrones inusuales mediante técnicas de fusión de datos e inteligencia artificial [40].

Las técnicas utilizadas incluyen análisis de rutas, análisis de velocidad, análisis de tiempo de espera y análisis de tráfico en tiempo real. El análisis de rutas permite detectar desviaciones o ingresos en zonas restringidas. El análisis de velocidad detecta barcos que viajan demasiado rápido o lento en relación con otros barcos en su área. El análisis de tiempo de espera permite detectar barcos que esperan demasiado tiempo en un puerto y el análisis de tráfico en tiempo real permite detectar barcos que navegan en direcciones opuestas a los otros barcos en su área.

El análisis de datos de tráfico marítimo es importante para la seguridad marítima y la gestión del tráfico marítimo, ya que permite a los operadores marítimos detectar patrones inusuales o anormales en la actividad de los barcos y tomar medidas para prevenir incidentes. Una de las páginas web más utilizada para este tipo de análisis es *Marine Traffic* (ver Figura 2-19), que recopila información en tiempo real de todos los barcos del mundo [41].

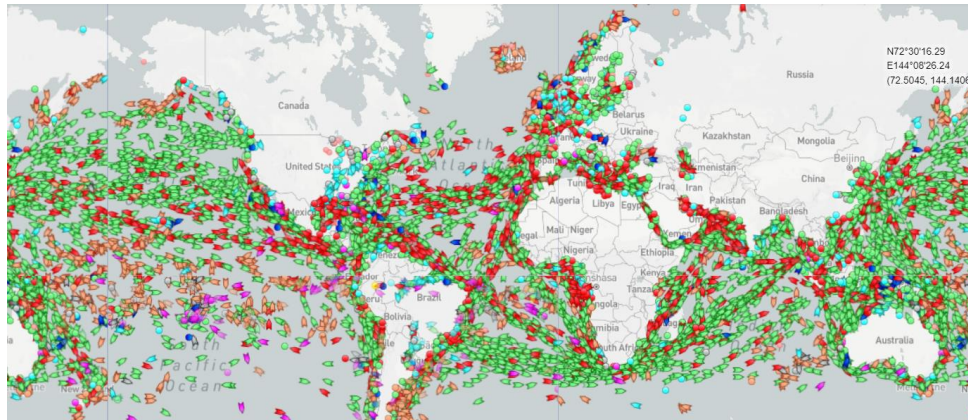


Figura 2-19 Información en tiempo real por *Marine Traffic* [40]

2.5.3 Análisis de imágenes satelitales con AIS

El análisis de imágenes satelitales para el control de barcos con AIS (*Automatic Identification System* o Sistema de Identificación Automática) es una técnica comúnmente utilizada para mejorar la seguridad en la navegación y monitorear el tráfico marítimo.

El AIS es un sistema de comunicaciones por radio que transmite información de identificación, posición, velocidad y dirección de los barcos a través de una señal VHF. Al utilizar imágenes satelitales para el control de barcos con AIS, se pueden monitorear las señales AIS para detectar la ubicación y el movimiento de los barcos, y luego se comparan con las imágenes satelitales para confirmar su ubicación y verificar que estén navegando en el lugar que deben estar [42].

Este tipo de análisis de imágenes satelitales con AIS puede ser especialmente útil para detectar barcos que pueden estar infringiendo las normas de navegación o que estén operando en áreas protegidas o restringidas.

Además, el análisis de imágenes satelitales para el control de barcos con AIS también puede ser útil para rastrear el tráfico marítimo y ayudar a las autoridades a responder rápidamente en caso de emergencias marítimas, como accidentes o derrames de petróleo.

En este mecanismo para la detección de anomalías en ambiente marítimo se capturan automáticamente numerosos datos de distintas fuentes que deben ser filtrados y procesados para que posteriormente puedan servir de utilidad. El fin es obtener una imagen completa del movimiento de los barcos a nivel mundial.

2.5.4 Información meteorológica

La información meteorológica es esencial para la detección de anomalías en el movimiento de los barcos, ya que las condiciones climáticas pueden afectar significativamente la navegación y el comportamiento de los barcos en el mar.

Por ejemplo, los fuertes vientos, las grandes olas y las corrientes marinas pueden causar movimientos bruscos e inusuales en los barcos, lo que puede indicar que el barco está navegando en condiciones peligrosas o que ha sufrido algún tipo de daño.

La información meteorológica también puede ser utilizada para prever el comportamiento futuro del mar y de las condiciones climáticas. Esto puede ayudar a las autoridades marítimas a tomar decisiones informadas sobre la navegación, como la ruta de navegación, la velocidad del barco, y la necesidad de precaución en caso de condiciones climáticas adversas [43].

Además, la información meteorológica puede ser utilizada en conjunto con la información de sensores y análisis de imágenes satelitales para detectar anomalías en el movimiento de los barcos. La

combinación de estas herramientas puede proporcionar una visión más completa y precisa del comportamiento de los barcos en el mar.

Una fuente confiable para obtener información meteorológica para la detección de anomalías en el movimiento de los barcos es la Agencia Estatal de Meteorología (AEMET), que es la agencia nacional de meteorología de España y proporciona información meteorológica y climatológica para toda España y sus áreas marítimas. La AEMET es responsable de la emisión de alertas y avisos meteorológicos, así como de la generación de pronósticos del tiempo y de condiciones marítimas [44].

2.5.5 Patrones de comportamiento

El análisis de patrones de comportamiento de los barcos es una técnica útil para detectar anomalías en el movimiento de los barcos. Esto implica la recopilación de datos de posicionamiento de los barcos durante un período de tiempo y el análisis de los patrones de movimiento típicos de los barcos en una determinada área.

El análisis de patrones de comportamiento se puede realizar utilizando diversas técnicas, como análisis estadísticos, técnicas de aprendizaje automático y algoritmos de detección de anomalías. Estas técnicas pueden ayudar a identificar patrones de movimiento atípicos de los barcos, que pueden ser indicativos de comportamientos inusuales o sospechosos.

Por ejemplo, el análisis de patrones de comportamiento puede detectar si un barco está moviéndose de manera errática o en patrones que no son comunes para el área. También puede identificar si un barco está desviándose de su ruta planificada o si está navegando a velocidades inusuales. Estas anomalías pueden indicar actividades ilegales, como el contrabando, la pesca ilegal o el tráfico de personas.

Las fuentes de datos para el análisis de patrones de comportamiento incluyen sistemas de seguimiento de barcos, como el AIS, que proporciona información sobre la ubicación, velocidad y dirección de los barcos en tiempo real. Otros sistemas de vigilancia, como los radares y las cámaras de seguridad, también pueden proporcionar datos de posicionamiento y movimiento de los barcos para su análisis [45].

El análisis de patrones de comportamiento es una técnica prometedora para la detección de anomalías en el movimiento de los barcos y puede ser útil para mejorar la seguridad en el mar y combatir actividades ilegales en las zonas marítimas.

3 DESARROLLO DEL TFG

En el presente capítulo se expone principalmente tanto el entorno de trabajo como el software empleado, detallando las diferentes herramientas utilizadas en el desarrollo de este proyecto. En primer lugar, se empieza con la definición de las tramas AIS de los barcos y cómo han sido estudiadas y utilizadas. En segundo lugar, se define y explica el lenguaje de programación *Python* el cual ha sido utilizado para el desarrollo de todos los códigos empleados.

A continuación, se habla sobre el programa *Open Street Map* y su principal uso en este trabajo, así como también la plataforma *Kepler.gl* que sirve de gran utilidad en el aspecto visual del trabajo. Por último, se describe de manera detallada el desarrollo del trabajo, explicando concretamente los pasos seguidos en la elaboración de los códigos y los análisis realizados con los datos obtenidos.

3.1 Entorno de trabajo y software empleado

3.1.1 Tramas AIS

Las tramas AIS son mensajes estandarizados utilizados por los barcos y las estaciones terrestres para intercambiar información de navegación y otros datos importantes relacionados con la seguridad marítima. Estas tramas contienen información como la posición del barco, su velocidad, rumbo, identificación y otros datos relevantes para la navegación y el control del tráfico marítimo. El sistema AIS se utiliza para mejorar la seguridad y la eficiencia del tráfico marítimo y para prevenir colisiones en el mar [46].

Estas tramas se han utilizado en el desarrollo de este trabajo para poder realizar tanto los distintos análisis estadísticos de las velocidades de los barcos, como los distintos *scripts* que estiman las velocidades a partir de las coordenadas GPS y el tiempo transcurrido y las anomalías en dichas velocidades.

3.1.2 Lenguaje de programación *Python*

Python [47] es un lenguaje de programación de alto nivel, interpretado y orientado a objetos. Fue creado en 1991 por *Guido van Rossum* y actualmente es uno de los lenguajes de programación más populares. *Python* es conocido por su sintaxis clara y sencilla, lo que lo hace fácil de aprender y de leer, además de contar con una gran cantidad de bibliotecas y herramientas disponibles para diversos usos, como el procesamiento de datos, la creación de aplicaciones web o la inteligencia artificial, entre otros.

Python es utilizado en muchas áreas, desde el desarrollo de aplicaciones de escritorio hasta la automatización de tareas y el análisis de datos. Es un lenguaje multiplataforma y se ejecuta en diversos sistemas operativos como *Windows*, *macOS* y *Linux* [47]. Gracias a este lenguaje de programación se han podido desarrollar distintos códigos que serán explicados más adelante.

3.1.3 Open Street Map

OpenStreetMap (OSM) [48] es un proyecto colaborativo en línea que tiene como objetivo crear un mapa mundial libre, editable y de uso público. OSM es similar a otros servicios de mapas en línea como Google Maps y Bing Maps, pero difiere en que cualquiera puede contribuir al mapa agregando, editando y verificando información geográfica, como carreteras, edificios, ríos, entre otros elementos.

El mapa base de OSM es creado por una comunidad de voluntarios de todo el mundo que utilizan herramientas en línea o aplicaciones específicas de edición de mapas. La información geográfica se almacena en una base de datos geoespacial y se puede descargar en diferentes formatos, lo que permite a los usuarios crear sus propias aplicaciones basadas en mapas [49].

La principal ventaja de OSM es su licencia de datos abierta, lo que significa que cualquiera puede usar los datos de OSM de forma gratuita y sin restricciones legales. Además, OSM permite a los usuarios crear mapas personalizados y adaptarlos a sus necesidades específicas, como mapas de ciclovías, de senderismo, de transporte público, entre otros.

En este proyecto OSM se ha utilizado para definir las coordenadas en las que se encuadra el Puerto de Marín (ver Figura 3-1) sobre el que se ha hecho el estudio de los datos de los barcos.

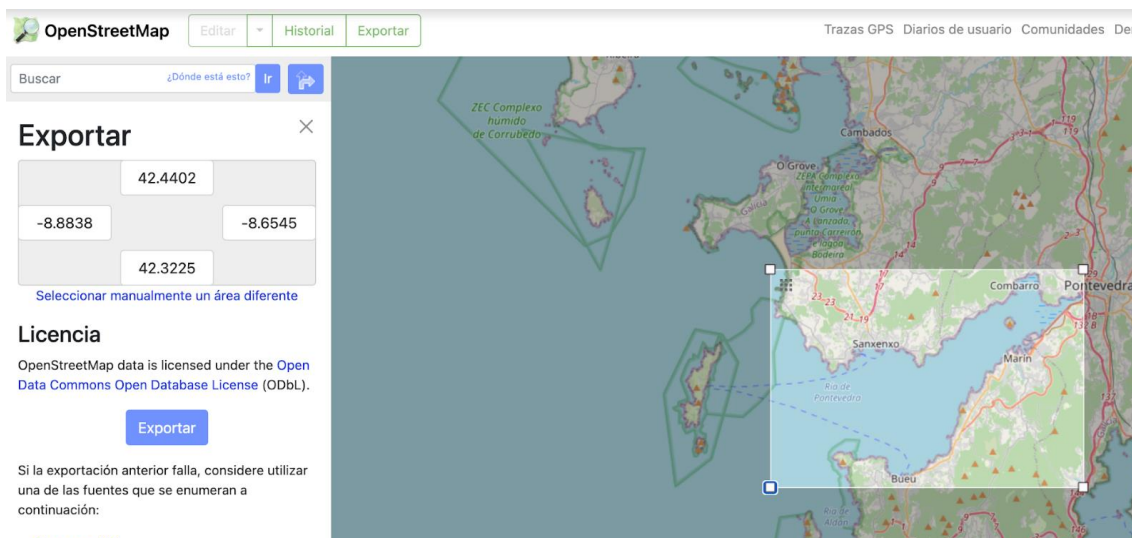


Figura 3-1 Coordenadas del Puerto de Marín y alrededores en *OpenStreetMap* (fuente: propia)

3.1.4 Kepler.gl

Kepler [50] es una plataforma de visualización de datos geoespaciales. Permite a los usuarios cargar, explorar y compartir datos geoespaciales en un formato fácil de usar y altamente personalizable. La plataforma se basa en *React.js* y utiliza la biblioteca de visualización de mapas *Mapbox GL* [51] para generar visualizaciones interactivas en 3D. *Kepler.gl* se ha utilizado en una variedad de aplicaciones, desde la visualización de patrones de tráfico hasta la identificación de patrones de uso de bicicletas compartidas en ciudades.

Es una herramienta poderosa para aquellos que trabajan con datos geoespaciales y desean crear visualizaciones y análisis dinámicos y personalizados [52]. Esta plataforma se ha utilizado en el desarrollo del trabajo para representar los movimientos de los barcos que se adentran en el Puerto de Marín. Se han introducido una serie de ficheros en formato *GeoJSON* [53] con los distintos datos de aproximación al puerto para que puedan ser representados como vemos en la Figura 3-2.

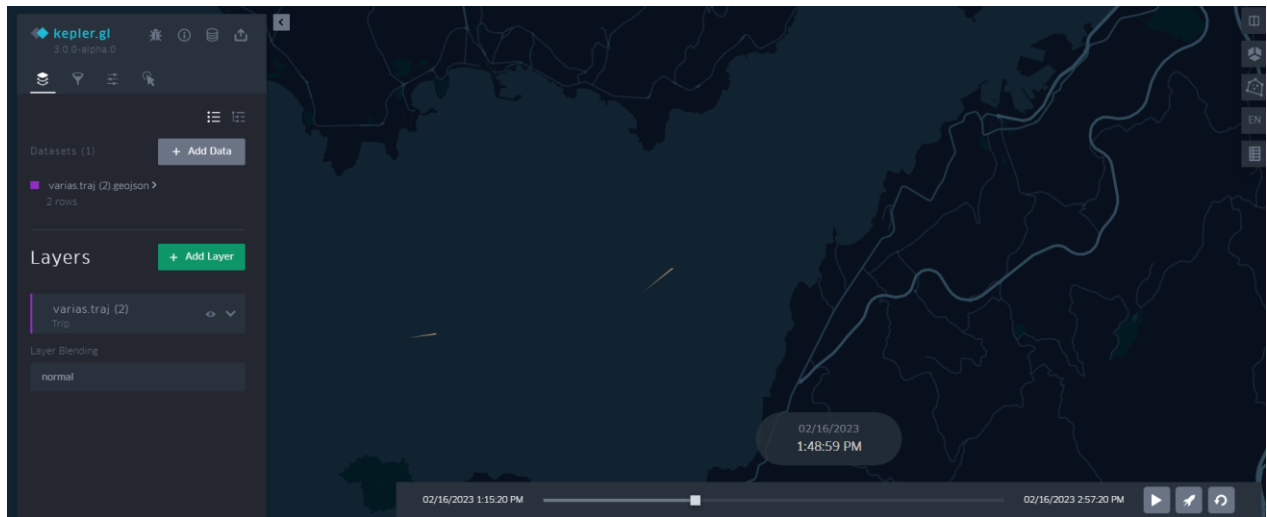


Figura 3-2 Representación de las trayectorias de distintos barcos en *kepler.gl* (fuente: propia)

3.2 Recopilación de datos AIS

En primer lugar, se han recopilado una serie de tramas AIS en formato NMEA (*National Marine Electronics Association*) de distintos barcos para llevar a cabo el estudio de los datos que se necesitan para el desarrollo del trabajo. Se ha partido para ello de ficheros conteniendo tramas AIS que representan la captura de una parte del tráfico marítimo internacional entre los días 1 de enero y 23 de febrero de 2023 (ver Figura 3-3).

```
!ABVDM,1,1,,A,13mvGf0000NDNb2UdQd0V`1J089N,0*5A,1675378757
!AIVDM,1,1,,1,13b`Nd0P000LJM>NSs>>0wvN0000,0*79,1675378757
!AIVDM,1,1,,A,H4hMOp1<5HDB09V0M84<D000000,2*62,1675378757
!AIVDM,2,1,4,A,559:qCh2B<@MP=5c621<44n1@E0ET4>222222011`=,0*48,1675378757
!AIVDM,2,2,4,A,5573o08l0CPC@H88888888888880,2*7F,1675378757
!BSVDM,1,1,,B,33@o`bW000Pq4DvOLInLna8P03rk,0*44,1675378757
!AIVDM,2,1,1,,5,53b`Nd4296S=08<j221M;:0ADp`6liTpr222220N2052400Ht0000000000,0*0E,1675378757
!AIVDM,2,2,1,,0,00000000008,2*6C,1675378757
!AIVDM,2,1,6,B,55N;5R02756iL@CO;3Hl<dTphET0000000000000`K,0*5F,1675378757
!AIVDM,2,2,6,B,284000:1RDj2Cqh0000000000000,2*10,1675378757
!BSVDM,1,1,,A,B3@onHh000;Dfs` :Hom;gwhUKP06,0*4E,1675378757
!BSVDM,2,1,5,A,5340qI`00000ucG?GD04qALE900000000000001I<H;66w?AT7hCU5iDT1C`,0*67,1675378757
!BSVDM,2,2,5,A,0UCRiDTh000,2*5D,1675378757
!AIVDM,1,1,5,A,H4eGN54N3=>6@8o00000000H@440,0*65,1675378757
!AIVDM,1,1,,1,13bLtJ7P000HPvPN@h0dE?vP0000,0*0D,1675378757
!AIVDM,2,1,5,A,59NSB0T2A4qh7QU7:20L5>0<E88E9E>2222221@:@A,0*34,1675378757
!AIVDM,2,2,5,A,7;4KaN>DmC2CPH88888888888880,2*78,1675378757
!BSVDM,1,1,,B,139Gw@0P@60kbVhO;Wd@kC?J0HJb,0*6C,1675378757
!AIVDM,2,1,1,,5,53bLtJ420c1M0E`R221=C;R0iD88E9B1<DpTu:0N2043321P044m0EQDQC`8,0*3B,1675378757
!AIVDM,2,2,1,,8,888888888888,2*6C,1675378757
```

Figura 3-3 Ejemplo de 20 tramas AIS en formato NMEA (fuente: propia)

Estos datos han sido filtrados por distintos códigos en *Python*, que a continuación se explicarán, para poder enfocarnos en aquellos que más nos interesan.

3.3 Selección de datos de barcos

Posteriormente, se ha desarrollado un programa en *Python* (**parse_ais.py**) que selecciona los datos AIS que queremos, tal como se muestra en la Figura 3-4.

```

1  ✓ import sys
2
3  from pyais.stream import FileReaderStream
4
5  ✓ for msg in FileReaderStream(sys.argv[1]):
6
7  ✓     try:
8         decoded = msg.decode().asdict()
9         # Comprobamos que los mensajes son del tipo adecuado
10        ✓     if "msg_type" in decoded and decoded["msg_type"] in [1,2,3,18,19]:
11
12            lat = decoded["lat"]
13            lon = decoded["lon"]
14            speed = decoded["speed"]
15            # Zona del puerto de Marín
16        ✓     if -8.8838 < lon and lon < -8.6545 and 42.3225 < lat and lat < 42.4402:
17            # Barco en movimiento
18        ✓     if speed > 1.0:
19            # Computa el sello temporal de los mensajes AIS que nos interesan
20            sentence_str = str(msg)
21        ✓     if sentence_str.endswith(">"):
22            # Imprime el objeto # AISentence<!BSVDM,1,1,,A,B3m8vaP00H9?lJa3kCR9Swb5oP06,0*6F,1672527385>
23            # print(sentence_str)
24            tokens = sentence_str.split(",")
25            # Extraer el sello temporal que aparece al final del mensaje
26            timestamp = int(tokens[-1][:-1])
27            decoded["time"] = timestamp
28            # Imprime la informacion de interes
29            print(decoded["mmsi"], lon, lat, 0, timestamp, speed, decoded["course"], decoded["heading"], sep=",")
30
31        ✓     except Exception:
32            continue

```

Figura 3-4 Código Python parse_ais.py (fuente: propia)

Este es un *script* que lee los mensajes AIS del archivo y filtra los mensajes de ciertos tipos y dentro de una determinada área geográfica. Para los mensajes filtrados, el *script* extrae e imprime información específica como el número MMSI (*Maritime Mobile Service Identity* o Identidad de Servicio Móvil Marítimo), longitud, latitud, altitud (en este caso siempre es 0 ya que los barcos se encuentran a nivel del mar), marca de tiempo, velocidad, rumbo sobre fondo y rumbo de superficie.

El *script* utiliza la biblioteca *pyais* para leer y decodificar los mensajes AIS del archivo especificado, y la biblioteca *sys* para obtener el nombre del archivo de los argumentos de la línea de comandos.

La clase *FileReaderStream* de *pyais.stream* se utiliza para leer el archivo como un flujo continuo, devolviendo objetos de mensajes AIS uno a uno en un bucle *for*. Cada mensaje se decodifica utilizando el método *decode* y se convierte en un diccionario *Python* utilizando el método *asdict*.

El *script* comprueba que el mensaje decodificado sea de uno de los tipos AIS especificados (1, 2, 3, 18, 19), que ofrecen información de posicionamiento, y dentro de un área geográfica especificada (un área rectangular alrededor del Puerto de Marín en España).

Si el mensaje pasa el filtro, el *script* extrae la información relevante (MMSI, longitud, latitud, altitud, velocidad, rumbo sobre fondo, rumbo de superficie) del diccionario y calcula la marca de tiempo a partir del final de la cadena de mensaje. Luego se imprime la marca de tiempo y otra información en la consola.

El bloque *try-except* se utiliza para capturar cualquier situación excepcional o error que pueda ocurrir durante la decodificación o la impresión, y la instrucción *continue* se utiliza para omitir cualquier mensaje que no se pueda decodificar o imprimir.

En general, este *script* proporciona un ejemplo sencillo de cómo filtrar y extraer información de los mensajes AIS utilizando *Python* y la biblioteca *pyais*. Con este código conseguimos tener datos de 54 barcos diferentes en un fichero de tipo CSV que posteriormente va a ser procesado nuevamente por un código *Python* distinto.

3.4 Selección de barcos con trayectorias de aproximación al puerto de Marín

A continuación se expone el siguiente *script* en *Python* (**calcula_traj.py**) que selecciona los datos de los barcos que se aproximan hacia el puerto de Marín (ver Figura 3-5 y 3-6), se ha centrado el estudio en estos ya que son los más representativos de posibles trayectorias de ataque hacia el puerto.

```

1  import csv
2  import sys
3
4  import geopy.distance
5
6
7  MARIN = (42.397067, -8.69289)
8
9  MIN_PUNTOS = 8
10
11 # Construimos un diccionario de trayectorias organizado por MMSI
12 trayectorias = {}
13 with open(sys.argv[1], "r") as fichero:
14     reader = csv.reader(fichero, delimiter=",")
15     for i, linea in enumerate(reader):
16         (mmsi, lon, lat, alt, time, speed, course, heading) = linea
17         # Distancia de la posicion al puerto de Marín
18         dist = geopy.distance.geodesic(MARIN, (float(lat), float(lon))).km
19         if not mmsi in trayectorias:
20             trayectorias[mmsi] = []
21         trayectorias[mmsi].append((time,lon,lat,dist,linea))
22

```

Figura 3-5 Código *Python* calcula_traj.py parte 1 (fuente: propia)

```

23
24 # Procesamos las posiciones para encontrar trayectorias de aproximacion
25 for mmsi in trayectorias:
26     trayectoria = trayectorias[mmsi]
27     longitud = len(trayectoria)
28     # Queremos trayectorias largas, que tengan al menos unos cuantos puntos
29     if longitud > MIN_PUNTOS:
30         # Un valor muy grande para que la primera vez funcione la comparacion
31         dist_ant = 2000
32         aprox = True
33         # En las rutas de aproximacion todos los puntos estan cada vez mas cerca del puerto de destino
34         for pos in range(longitud):
35             (time,lon,lat,dist,linea) = trayectoria[pos]
36             if dist > dist_ant:
37                 aprox = False
38                 break
39             dist_ant = dist
40
41         if aprox:
42             print("MMSI: ", mmsi)
43
44             # Prefijo a emplear en los ficheros de salida
45             filename = sys.argv[1][:3] + str(mmsi) + ".traj"
46             with open(filename, 'w') as f:
47                 for punto in trayectoria:
48                     f.write(",".join(punto[4]))
49                     f.write("\n")
50

```

Figura 3-6 Código *Python* calcula_traj.py parte 2 (fuente: propia)

Este código en *Python* procesa un archivo CSV que contiene información de posición de barcos y busca trayectorias sólo de aproximación a un puerto determinado (el de Marín en este caso, como ya hemos mencionado). Primero, se crea un diccionario de trayectorias organizado por MMSI. Para cada posición en el archivo, se calcula la distancia al puerto y se agrega al diccionario correspondiente.

Luego, se procesan las posiciones para encontrar trayectorias de aproximación. Se busca una secuencia de posiciones que tengan un tamaño, un número de puntos, superior a un umbral predefinido, en la que la distancia al puerto disminuye en cada punto. Si se encuentra una trayectoria de este tipo, se

escribe en un archivo de texto, con la información de todas las posiciones en esa trayectoria. Nótese que se cream ficheros distintos para trayectorias distintas.

El código utiliza la biblioteca *Geopy* para calcular la distancia entre dos puntos en la superficie terrestre a través de la función *geodesic*. La constante MARIN se define como la ubicación del puerto al que se quiere llegar. El valor MIN_PUNTOS define la longitud mínima que debe tener una trayectoria para ser considerada.

Gracias a estos dos *script* (**parse_ais.py** y **calcula_traj.py**), el número de trayectorias que nos interesan para el desarrollo de este trabajo son 36, entre el 1 de enero y el 23 de febrero de 2023. En la Figura 3-7 se puede ver un ejemplo del formato CSV de un fichero *.traj* generado.

```
244020808,-8.878973,42.33663,0,1672768980,5.7,50.6,50
244020808,-8.870665,42.342032,0,1672769286,5.7,48.7,47
244020808,-8.862578,42.34743,0,1672769586,5.8,47.9,47
244020808,-8.834908,42.35774,0,1672770504,5.3,81.5,82
244020808,-8.82509,42.359343,0,1672770804,5.4,78.7,82
244020808,-8.814993,42.360817,0,1672771105,5.4,78.9,83
244020808,-8.805018,42.362097,0,1672771405,5.3,79.4,83
244020808,-8.77254,42.370428,0,1672772615,4.4,49.6,47
244020808,-8.731807,42.39265,0,1672774156,5.8,60.1,58
244020808,-8.723857,42.395628,0,1672774464,3.2,75.8,66
244020808,-8.715037,42.398797,0,1672775065,2.6,58.9,64
244020808,-8.706313,42.40144,0,1672775686,2.1,71.3,69
```

Figura 3-7 Ejemplo del formato CSV de un fichero *.traj* (fuente: propia)

3.5 Análisis estadístico de las velocidades de los barcos

Una vez obtenidos los datos de esas 36 trayectorias, se han seleccionado únicamente las velocidades y con ellas se ha hecho un análisis estadístico en *Excel* en el que se representan en una columna todos los datos de las velocidades de dichos barcos. También se han calculado la media, la mediana, el primer cuartil, el tercer cuartil, el límite inferior y el límite inferior, y con todos estos datos se ha hecho tanto una gráfica que representa los valores de velocidad para cada muestra que hay en cada fichero de trayectoria (ver Figura 3-8) como un diagrama de caja (*boxplot*) (ver Figura 3-9).

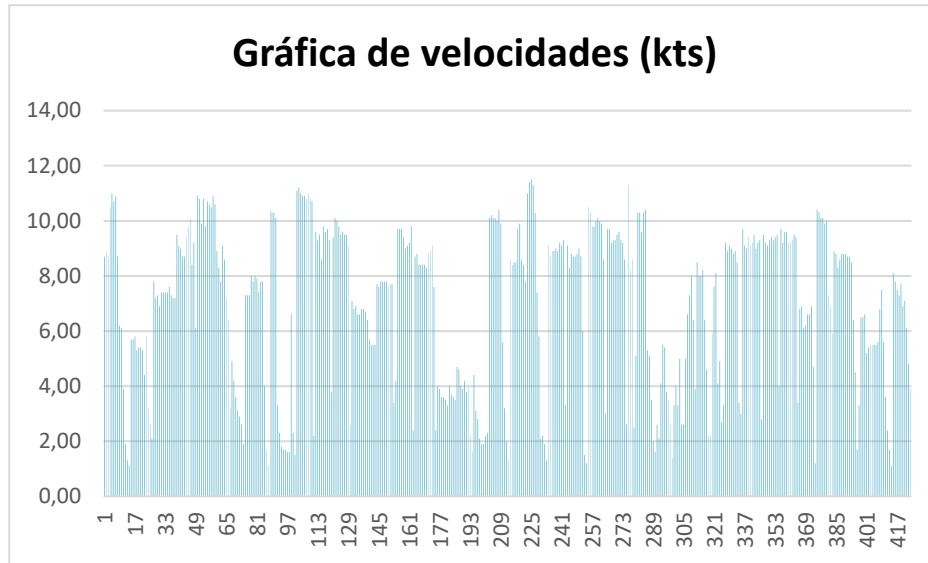


Figura 3-8 Gráfica de velocidades en Excel (fuente: propia)

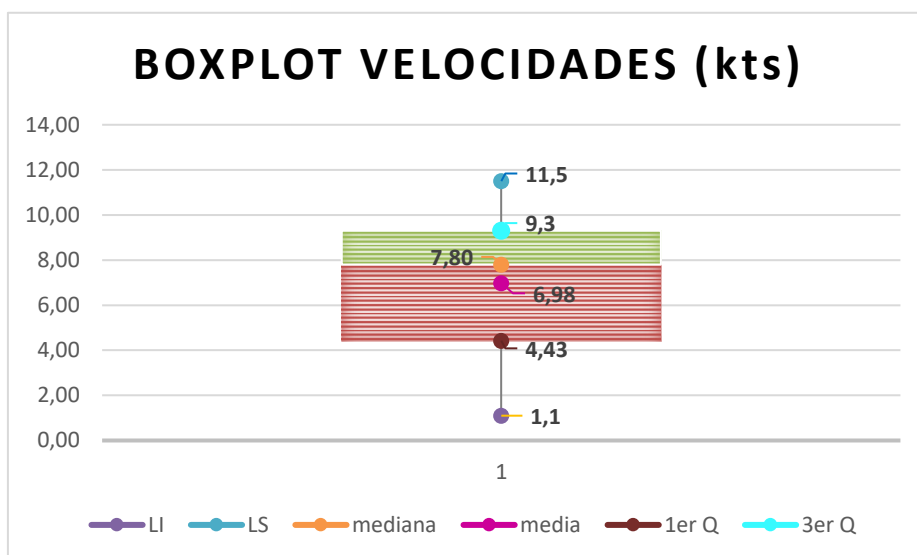


Figura 3-9 Boxplot de velocidades en Excel (fuente: propia)

Este análisis nos ha servido principalmente para calcular la velocidad media que, como vemos en el *boxplot*, tiene un valor de 6,98 kts, y definir el límite inferior, con un valor de 1,1 kts, y el superior, con un valor de 11,5 kts, de las velocidades de los barcos en la entrada al puerto de Marín que pueden servir para orientar valores anómalos de velocidades de barcos detectados.

3.6 Estimar las velocidades de los barcos a partir de las coordenadas GPS y el tiempo

Para extraer las estimaciones de velocidad a partir de las coordenadas GPS y los tiempos en los ficheros.traj, podemos calcular la distancia recorrida y el tiempo transcurrido entre cada par de puntos consecutivos. Con estos valores, podemos calcular la velocidad media entre los dos puntos y agregarla a una lista. Luego, podemos realizar el análisis estadístico sobre esta lista de velocidades en Excel como se ha realizado anteriormente con los datos de las velocidades proporcionados por las tramas AIS.

Para estimar las velocidades a partir de las coordenadas GPS y el tiempo se ha desarrollado el siguiente código en *Python* (**estima_velo.py**) que podemos ver en la Figura 3-10 y Figura 3-11:

```

1  import os
2  import csv
3  from math import radians, cos, sin, asin, sqrt
4
5  # función para calcular la distancia entre dos puntos GPS en la superficie de la Tierra
6  def haversine(lon1, lat1, lon2, lat2):
7      """
8      Calcula la distancia en kilómetros entre dos puntos GPS en la superficie de la Tierra.
9      """
10     # convertir grados decimales a radianes
11     lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])
12
13     # fórmula de Haversine
14     dlon = lon2 - lon1
15     dlat = lat2 - lat1
16     a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
17     c = 2 * asin(sqrt(a))
18     r = 6371 # radio medio de la Tierra en kilómetros
19     return c * r
20
21 # Definir la ruta a la carpeta "barcos_csv" en tu escritorio
22 ruta = os.path.expanduser("C:/Users/munil/OneDrive/Escritorio/barcos_csv")
23
24 # abrir el archivo velocidades.csv para escribir los resultados
25 with open('velocidades.csv', mode='w', newline='') as velocidades_file:
26     writer = csv.writer(velocidades_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)

```

Figura 3-10 Código *Python* estima_velo.py parte 1 (fuente: propia)

```

27
28 # Recorrer la lista de nombres de archivos y procesar cada archivo
29 for nombre_archivo in os.listdir(ruta):
30     # Construir la ruta completa al archivo
31     ruta_archivo = os.path.join(ruta, nombre_archivo)
32
33     # leer el archivo CSV
34     with open(ruta_archivo) as csvfile:
35         reader = csv.reader(csvfile, delimiter=',')
36         rows = list(reader)
37
38     # convertir las cadenas de texto a números
39     data = [(float(row[0]), float(row[1]), int(row[2])) for row in rows]
40
41     # Obtener el identificador del barco a partir del nombre del archivo
42     id_barco = nombre_archivo.split(".")[0]
43
44     for i in range(len(data)-1):
45         # calcular la distancia entre dos puntos consecutivos
46         distance = haversine(data[i][1], data[i][0], data[i+1][1], data[i+1][0])
47
48         # calcular el tiempo transcurrido entre dos puntos consecutivos
49         time = data[i+1][2] - data[i][2]
50
51         # calcular la velocidad en nudos náuticos
52         knots = distance * 3600 / (time * 1.852)
53
54         # escribir los resultados en el archivo CSV junto con el identificador del barco
55         writer.writerow([id_barco, f"{knots:.2f}"])

```

Figura 3-11 Código *Python* estima_velo.py parte 2 (fuente: propia)

Este código es un *script* de *Python* que procesa una serie de archivos CSV que contienen información sobre la ubicación de barcos en diferentes momentos y calcula la velocidad en nudos náuticos entre cada punto de ubicación.

El *script* comienza importando las librerías necesarias *os*, *csv* y *math*. La librería *os* se utiliza para acceder a los archivos en la carpeta "barcos_csv". La librería *csv* se utiliza para leer y escribir archivos CSV y *math* se utiliza para realizar cálculos matemáticos.

Luego, se define una función *haversine* [54] que calcula la distancia en kilómetros entre dos puntos GPS en la superficie de la Tierra utilizando la fórmula de *Haversine*. La fórmula de *Haversine* es una ecuación que se utiliza para calcular la distancia entre dos puntos en una esfera, en este caso, la Tierra.

A continuación, se define la ruta a la carpeta "barcos_csv" en el escritorio utilizando *os.path.expanduser* y se crea un archivo llamado "velocidades.csv" para escribir los resultados utilizando *csv.writer*.

Luego, se recorre la lista de nombres de archivos en la carpeta "barcos_csv" utilizando *os.listdir*. Para cada archivo, se construye la ruta completa al archivo utilizando *os.path.join*. Luego, se lee el archivo CSV utilizando *csv.reader* y se convierten las cadenas de texto a números utilizando una comprensión de lista.

A continuación, se obtiene el identificador del barco a partir del nombre del archivo utilizando *split* y se recorre la lista de ubicaciones para calcular la velocidad en nudos náuticos entre cada punto de ubicación. Para cada par de ubicaciones consecutivas, se calcula la distancia utilizando la función *haversine*, se calcula el tiempo transcurrido entre las dos ubicaciones y se calcula la velocidad en nudos náuticos utilizando la fórmula de la velocidad.

Por último, se escriben los resultados en el archivo CSV utilizando *csv.writer*. Cada fila del archivo contiene el identificador del barco y la velocidad en nudos náuticos entre dos puntos consecutivos.

En resumen, el código lee los archivos CSV de trayectorias en una carpeta, calcula la velocidad entre cada par de ubicaciones consecutivas para cada barco y escribe los resultados en un archivo llamado "velocidades.csv". Con estos nuevos datos estimados se realiza un nuevo estudio estadístico de las velocidades que se expone a continuación.

3.7 Análisis estadísticos de las velocidades estimadas con el código *Python*

Se ha realizado el correspondiente estudio estadístico en *Excel* de los datos de velocidades estimadas gracias al ya mencionado código en *Python*. En él se ha realizado el estudio y cálculo de la media, la mediana, el primer cuartil, el tercer cuartil, el límite inferior y el límite superior, y con todos estos datos se ha hecho tanto una gráfica (ver Figura 3-12) como un *boxplot* (ver Figura 3-13).

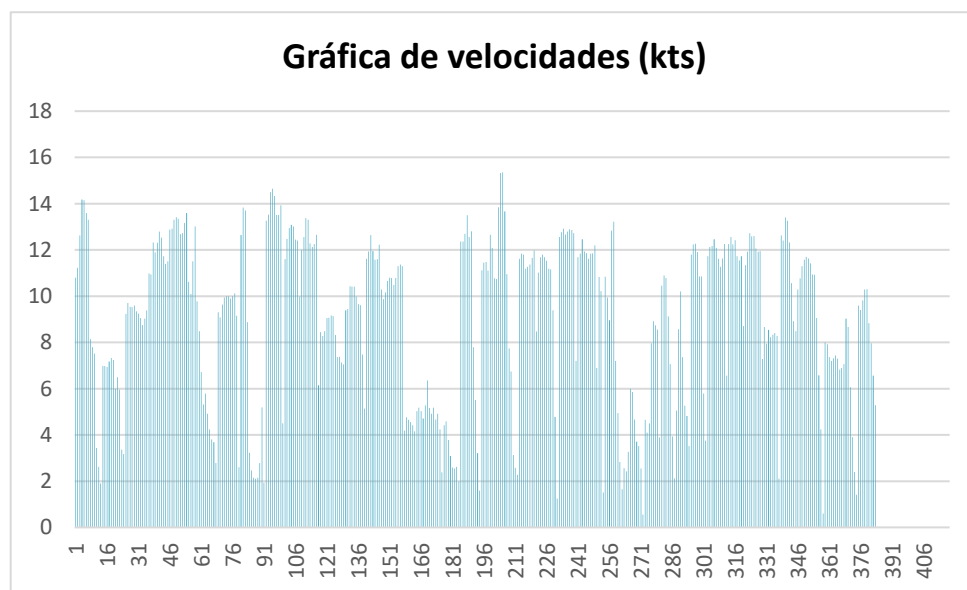


Figura 3-12 Gráfica de velocidades estimadas en *Excel* (fuente: propia)

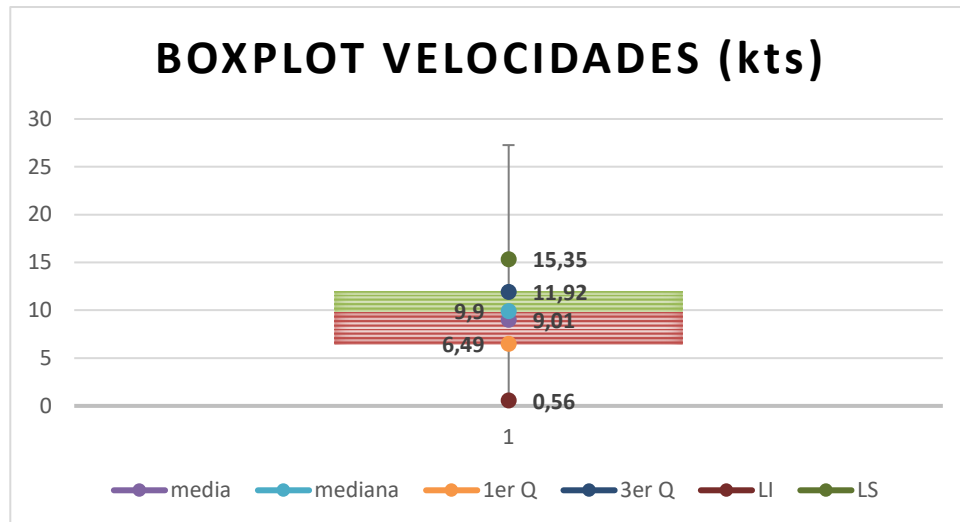


Figura 3-13 Boxplot de velocidades estimadas en Excel (fuente: propia)

Al igual que el análisis estadístico anterior, destacar los valores de la media (9,01kts), el límite inferior (0,56 kts) y el límite superior (15,35kts), que serán usados como referencia en el posterior desarrollo de la detección de anomalías en las velocidades de los barcos.

3.8 Código para detección de anomalías en las velocidades de los barcos

Una vez obtenidas las velocidades estimadas, se procede al desarrollo de un código *Python* (**det_anom.py**) específico para detectar anomalías en las velocidades de los barcos, el código es el que se puede ver en la Figura 3-14:

```

1 import csv
2 import statistics
3
4 # leer el archivo CSV
5 with open('velocidades.csv') as csvfile:
6     reader = csv.reader(csvfile, delimiter=',')
7     rows = list(reader)
8
9 # convertir las cadenas de texto a números y separar el número de barco de la velocidad
10 data = [(row[0], float(row[1])) for row in rows[1:]]
11
12 # calcular la media y la desviación estándar de las velocidades
13 velocidades = [d[1] for d in data]
14 media = statistics.mean(velocidades)
15 desviacion_estandar = statistics.stdev(velocidades)
16
17 # definir el umbral para detectar las anomalías
18 umbral = media + 2 * desviacion_estandar
19
20 # detectar las anomalías en las velocidades
21 anomalias = [(d[0], d[1]) for d in data if d[1] > umbral]
22
23 # imprimir las velocidades anómalas
24 if len(anomalias) > 0:
25     print("Velocidades anómalas encontradas:")
26     for a in anomalias:
27         print(f"Velocidad anómala {a[1]} encontrada en el barco {a[0]}.")
28 else:
29     print("No se encontraron velocidades anómalas.")

```

Figura 3-14 Código *Python* det_anom.py (fuente: propia)

Este código tiene como objetivo leer un archivo CSV que contiene la velocidad de los barcos y detectar velocidades anómalas. Para lograr esto, primero se abre el archivo CSV (*velocidades.csv*) con la función *open* del módulo *csv* y se lee el contenido con la función *reader* de este mismo módulo. La variable *rows* almacena los datos leídos del archivo.

Luego, se convierten las cadenas de texto a números y se separa el identificador del barco de la velocidad en la variable *data*. Para ello, se utiliza una comprensión de lista para recorrer las filas del archivo CSV, pero se omite la primera fila que contiene los encabezados. En la variable *data*, se almacenan los datos del barco y su velocidad correspondiente.

A continuación, se calcula la media y la desviación estándar de las velocidades mediante la función *mean* y *stdev* del módulo *statistics*. Estos valores se utilizan para definir un umbral que permitirá detectar las velocidades anómalas.

Luego, se recorre la lista de datos para encontrar velocidades que superen el umbral definido. Las velocidades anómalas se almacenan en la variable *anomalías*, que es una lista de tuplas que contiene el número de barco y la velocidad anómala.

Finalmente, si se detectaron velocidades anómalas, se muestra un mensaje que indica el número de barco y la velocidad anómala correspondiente. De lo contrario, se muestra un mensaje indicando que no se encontraron velocidades anómalas.

En el código, se define el umbral para detectar las anomalías como la media más dos veces la desviación estándar de las velocidades. Este umbral se utiliza para identificar velocidades que están significativamente por encima del promedio y, por lo tanto, pueden ser consideradas como anómalas.

La razón para elegir la media más dos veces la desviación estándar como umbral se basa en la distribución normal de los datos. En una distribución normal, aproximadamente el 95% de los datos se encuentran dentro de dos desviaciones estándar de la media. Por lo tanto, cualquier valor que esté más allá de este umbral se considera estadísticamente inusual.

En el caso que se hayan detectado velocidades anómalas y el barco del cual proviene dicha anomalía, se procede a introducir los datos de la trayectoria de dicho barco (longitud, latitud, altitud y tiempo) en un archivo CSV llamado *trayectoria_anomala* dentro de la carpeta *trayec_anom* que posteriormente se analiza en un nuevo código *Python*.

3.9 Código para la interceptación por el dron al barco que presenta anomalías

Con los datos de la trayectoria anómala de un barco detectada en el código anterior, se procede al desarrollo del código que simule la interceptación de dicho barco (***intercept.py***) por parte del dron de superficie que se encuentra en las inmediaciones del puerto. Dicho código es el que se puede ver en las Figuras 3-15, 3-16 y 3-17.

```
1 import sys
2 import csv
3 import utm
4
5 import geopy.distance
6
7 from shapely.geometry import Point, LineString
8
9
10 # Punto inicial al lado de la ENM
11 (inicio_lat, inicio_lon) = (42.3949, -8.7125)
12 inicio = (inicio_lon, inicio_lat)
13
14 velocidad = 10 # nudos
15 delta_t = 1 # segundos
16
17 # Recorremos todas las lineas del fichero de entrada CSV y nos quedamos con lon, lat y time
18 trayectoria = []
19 with open("C:/Users/munil/OneDrive/Escritorio/trayec_anom/trayectoria_anomala.csv", "r") as fichero:
20     reader = csv.reader(fichero, delimiter=",")
21     for i, linea in enumerate(reader):
22         if len(linea) >= 4:
23             (lon, lat, alt, time) = linea
24             trayectoria.append((float(lon), float(lat), int(time)))
25
```

Figura 3-15 Código Python intercept.py parte 1 (fuente: propia)

```
26 # Tenemos cargado en trayectoria la ruta del objetivo vamos a calcular la del drone
27 trayectoria_drone = []
28
29 # Posicion de inicio del movimiento del drone
30 (drone_lon, drone_lat) = inicio
31 # Añadimos esa posicion a la trayectoria del drone (con el sello temporal y altitud del inicio de la trayectoria del objetivo)
32 trayectoria_drone.append((drone_lon, drone_lat, trayectoria[0][2], trayectoria[0][1]))
33
34 for i in range(0, len(trayectoria)-1):
35     # Segmento de trayectoria del objetivo
36     (obj_lon_init, obj_lat_init, obj_time_init) = trayectoria[i]
37     (obj_lon_end, obj_lat_end, obj_time_end) = trayectoria[i+1]
38
39     # Calculamos el paso de tiempo
40     delta_t = obj_time_end - obj_time_init
41
42     # Distancia del drone al objetivo
43     dist = geopy.distance.geodesic((obj_lat_end, obj_lon_end), (drone_lat, drone_lon)).nm
44
45     # Convertir GPS a UTM
46     (drone_x, drone_y, init_zone_num, init_zone_char) = utm.from_latlon(drone_lat, drone_lon)
47     (obj_x, obj_y, end_zone_num, end_zone_char) = utm.from_latlon(obj_lat_end, obj_lon_end)
48
49
```

Figura 3-16 Código Python intercept.py parte 2 (fuente: propia)

```
50 # Definir la linea con su origen y final (movimiento desde el drone al objetivo)
51 start_point = Point(drone_x, drone_y)
52 end_point = Point(obj_x, obj_y)
53 line_of_sight = LineString([start_point, end_point])
54
55 # El drone avanza en la direccion de la LOS una distancia que depende de la velocidad y tiempo
56 desplazamiento = (velocidad * (delta_t/3600))/dist
57 siguiente_punto = line_of_sight.interpolate(desplazamiento, normalized = True)
58
59 # Convertirlos a formato (lat, lon)
60 (drone_lat, drone_lon) = utm.to_latlon(siguiente_punto.x, siguiente_punto.y, end_zone_num, end_zone_char)
61
62 trayectoria_drone.append((drone_lon, drone_lat, obj_time_end))
63
64 # Escribir la trayectoria del drone en un archivo txt
65 with open("C:/Users/munil/OneDrive/Escritorio/trayectoria_dron/trayectoria_dron.csv", "w") as f:
66     writer = csv.writer(f, delimiter=",")
67     for punto in trayectoria_drone:
68         writer.writerow([punto[0], punto[1], 0, punto[2]]) # Agregar altitud = 0
```

Figura 3-17 Código Python intercept.py parte 3 (fuente: propia)

Este código tiene como objetivo simular el movimiento de un dron que sigue a un objetivo en movimiento ya que ha sido detectado con velocidades anómalas. Para ello, se utiliza un archivo CSV que contiene la trayectoria del objetivo.

En primer lugar, se importan las librerías necesarias: *sys* para interactuar con el sistema operativo, *csv* para leer y escribir archivos CSV, *utm* para convertir entre coordenadas GPS y UTM, *geopy.distance* para calcular la distancia entre dos puntos geográficos y *shapely.geometry* para trabajar con geometría y realizar cálculos.

A continuación, se definen los parámetros de velocidad y tiempo *delta_t* del dron, así como la posición inicial del dron (en coordenadas latitud/longitud). Luego, se lee el archivo CSV que contiene la trayectoria del objetivo, se procesan los datos y se almacenan en una lista llamada "trayectoria".

Por consiguiente, se define una lista vacía llamada "trayectoria_dron" que se utilizará para almacenar la trayectoria del dron. Se añade la posición inicial del dron a esta lista.

Se utiliza un bucle *for* para recorrer cada segmento de la trayectoria del objetivo y calcular la trayectoria del dron correspondiente. En cada iteración del bucle, se calcula el tiempo *delta_t* del segmento de trayectoria, la distancia entre la posición del dron y el final del segmento de trayectoria y se convierten las coordenadas GPS en UTM para poder trabajar con ellas.

La función *utm.from_latlon* se utiliza para convertir las coordenadas de latitud y longitud del dron y del objetivo a coordenadas UTM. El resultado de la función *utm.from_latlon* es una tupla que contiene la coordenada x, la coordenada y, el número de zona UTM y la letra de zona UTM del punto en cuestión.

Las variables *end_zone_num* y *end_zone_char* almacenan el número de zona UTM y la letra de zona UTM del punto objetivo, respectivamente, después de la conversión a UTM mediante *utm.from_latlon*. Estas variables se utilizan más adelante en el código para convertir las coordenadas cartesianas del dron a coordenadas geográficas (latitud y longitud) mediante la función *utm.to_latlon*.

A continuación, se define una línea de visión (LOS) entre la posición actual del dron y el final del segmento de trayectoria. Se utiliza la función *interpolate* de la librería *shapely* para calcular el siguiente punto en la trayectoria del dron, avanzando una distancia proporcional a la velocidad y el tiempo *delta_t*. Por último, se convierten las coordenadas UTM del punto obtenido en coordenadas GPS y se añade a la lista de trayectoria del dron.

Finalmente, se escribe la trayectoria del dron en un archivo CSV llamado "trayectoria_dron.csv". Cada fila del archivo contiene las coordenadas latitud, longitud y altitud (de valor 0) del punto.

3.10 Código para convertir a formato *GeoJSON*

A raíz de los archivos obtenidos, se ha desarrollado un código en *Python* para convertir a formato *GeoJSON* y poder visualizar en *Kepler.gl* los datos de una trayectoria anómala de un barco y los datos de la trayectoria de interceptación por parte del dron hacia ese barco (**geojson_tray.py**), ya que en *Kepler.gl* se trabaja con este formato de archivos. El código es el que se puede ver en las Figuras 3-18, 3-19, 3-20 y 3-21:

```
1 import csv
2 import json
3 import os
4
5 # Directorios de trayectorias
6 trayectorias_dron_dir = "C:/Users/munil/OneDrive/Escritorio/trayectoria_dron"
7 trayectorias_anomalas_dir = "C:/Users/munil/OneDrive/Escritorio/trayec_anom"
8
9 # Directorio de salida GeoJSON
10 geojson_dir = "C:/Users/munil/OneDrive/Escritorio/geojson"
11
12 # Leer trayectorias de drones
13 for filename in os.listdir(trayectorias_dron_dir):
14     if filename.endswith(".csv"):
15         file_path = os.path.join(trayectorias_dron_dir, filename)
16
17         # Obtener el nombre del archivo sin extensión
18         filename_no_ext = os.path.splitext(filename)[0]
19
20         # Lista para almacenar las coordenadas
21         coordinates = []
22
23         # Lectura del archivo de trayectoria
24         with open(file_path) as f:
25             for line in f:
26                 # Separar los datos por comas
27                 data = line.strip().split(',')
28
```

Figura 3-18 Código Python geojson_tray.py parte 1 (fuente: propia)

```
28
29         # Verificar que todos los valores en la línea son numéricos y no están vacíos
30         if all(val != '' for val in data):
31             # Añadir la coordenada a la lista
32             coordinates.append([float(data[0]), float(data[1]), float(data[2]), int(data[3])])
33
34     # Creación del diccionario GeoJSON
35     geojson = {
36         "type": "FeatureCollection",
37         "features": [
38             {
39                 "type": "Feature",
40                 "properties": {
41                     "vendor": "dron"
42                 },
43                 "geometry": {
44                     "type": "LineString",
45                     "coordinates": coordinates
46                 }
47             }
48         ]
49     }
50
51     # Archivo de salida
52     output_path = os.path.join(geojson_dir, f"{filename_no_ext}.json")

```

Figura 3-19 Código Python geojson_tray.py parte 2 (fuente: propia)

```
53
54     # Guardar el diccionario GeoJSON en un archivo
55     with open(output_path, "w") as f:
56         json.dump(geojson, f)
57         print(f"Trayectoria {filename_no_ext} convertida a GeoJSON en {output_path}")
58
59 # Leer trayectorias anómalas
60 for filename in os.listdir(trayectorias_anomalas_dir):
61     if filename.endswith(".csv"):
62         file_path = os.path.join(trayectorias_anomalas_dir, filename)
63
64         # Obtener el nombre del archivo sin extensión
65         filename_no_ext = os.path.splitext(filename)[0]
66
67         # Lista para almacenar las coordenadas
68         coordinates = []
69
70         # Lectura del archivo de trayectoria anómala
71         with open(file_path) as f:
72             reader = csv.reader(f)
73             # Saltar la primera línea del archivo, que contiene los encabezados
74             next(reader)
75             for row in reader:
76                 # Verificar que todos los valores en la fila son numéricos y no están vacíos
77                 if all(val != '' for val in row):
78                     # Añadir la coordenada a la lista
79                     coordinates.append([float(row[0]), float(row[1]), float(row[2]), int(row[3])])
80
```

Figura 3-20 Código Python geojson_tray.py parte 3 (fuente: propia)

```

81     # Creación del diccionario GeoJSON
82     geojson = {
83         "type": "FeatureCollection",
84         "features": [
85             {
86                 "type": "Feature",
87                 "properties": {
88                     "vendor": "anomalous"
89                 },
90                 "geometry": {
91                     "type": "LineString",
92                     "coordinates": coordinates
93                 }
94             }
95         ]
96     }
97
98     # Archivo de salida
99     output_path = os.path.join(geojson_dir, f"{filename_no_ext}.json")
100
101     # Guardar el diccionario GeoJSON en un archivo
102     with open(output_path, "w") as f:
103         json.dump(geojson, f)
104     print(f"Trayectoria anómala {filename_no_ext} convertida a GeoJSON en {output_path}")

```

Figura 3-21 Código Python `geojson_tray.py` parte 4 (fuente: propia)

Este código tiene como objetivo convertir archivos de trayectorias de drones y trayectorias anómalas en formato CSV a formato *GeoJSON* y almacenarlos en un directorio especificado.

Primero, se establecen las rutas de los directorios donde se encuentran los archivos de las trayectorias de los drones y de las trayectorias anómalas, así como la ruta del directorio donde se almacenarán los archivos *GeoJSON* resultantes.

Luego, se utiliza un ciclo *for* para leer los archivos CSV de las trayectorias de los drones en el directorio correspondiente. Se verifica si cada archivo tiene la extensión `.csv` y se obtiene la ruta completa del archivo. También se obtiene el nombre del archivo sin la extensión para usarlo más adelante como nombre del archivo de salida.

Después, se crea una lista vacía para almacenar las coordenadas. Se abre el archivo y se lee línea por línea, separando los valores de cada línea por comas. Se verifica que todos los valores de la línea sean numéricos y no estén vacíos, y si es así, se añade la coordenada a la lista.

Luego, se crea un diccionario *GeoJSON* que contiene información sobre la trayectoria de drones. Se establece el tipo de la colección de características como *FeatureCollection*, se establece el tipo de la característica como *Feature*, se agrega información adicional como el proveedor del dron y se establece el tipo de geometría como *LineString*. La lista de coordenadas previamente creada se agrega como el valor de la clave *coordinates*.

A continuación, se establece la ruta de salida del archivo *GeoJSON*, que es la ruta del directorio de salida especificado y el nombre del archivo actual con extensión `.json`. Se guarda el diccionario *GeoJSON* en un archivo utilizando el módulo *json*.

Finalmente, se imprime un mensaje en la consola indicando que la trayectoria del dron correspondiente se ha convertido en *GeoJSON* y se ha guardado en el directorio de salida especificado.

Luego, se hace un proceso similar para leer los archivos CSV de las trayectorias anómalas en el directorio correspondiente, utilizando el módulo *csv* para leer el archivo y saltar la primera línea que contiene los encabezados. También se cambia el valor de la clave *vendor* en el diccionario *GeoJSON* a *anomalous* para indicar que se trata de una trayectoria anómala. El resto del proceso es el mismo que para las trayectorias de los drones.

4 PRUEBA Y VALIDACIÓN

4.1 Estudio de las velocidades

A la hora de simular la detección de un comportamiento anómalo, el trabajo se centra en el análisis de velocidades, uno de los escenarios más simples. Aunque, por completitud, se ha hecho un estudio de las velocidades declaradas en las tramas AIS, a la hora de hacer detección de anomalías se usan los parámetros estadísticos obtenidos en el análisis de las velocidades estimadas a partir de la posición, ya que es probable que un barco con intenciones sospechosas no proporcione sus tramas AIS, por lo tanto es óptimo establecer los umbrales de las anomalías con dichos datos, que se muestran en la Tabla 4-1.

Datos	Valores
Media	9,01 kts
Mediana	9,90 kts
1er Q	6,49 kts
3er Q	11,92 kts
Lím. inf	0,56 kts
Lím. sup	15,35 kts
Desv. Estándar	3,56 kts

Tabla 4-1 Datos de las velocidades estimadas a partir de la posición (fuente: propia)

4.2 Explicación con ejemplo

Si un barco navega por la Ría de Pontevedra, se identifica si su trayectoria es de aproximación al Puerto de Marín, si no lo es, ese barco se descarta. A continuación, se hace un estudio de sus velocidades y se determina si hay anomalías. Se asume que el barco no transmite sus datos AIS, pero se procede a estimar su velocidad teniendo en cuenta su posición y los intervalos de tiempo que transcurren (asumiendo que existen mecanismos que permitan obtener la información).

Posteriormente, se determina si esas velocidades estimadas son anómalas en base a unos criterios establecidos. En el caso de que no se detecten anomalías, el dron de superficie permanece en las inmediaciones del puerto ya que dicho barco no debe ser interceptado.

En caso contrario, si el barco presenta anomalías en sus velocidades, se procede a la parte de la interceptación, en la que el dron detecta el momento en el que la velocidad del barco es anómala y comienza su trayectoria de aproximación en base a lo establecido en los códigos mencionados en el apartado del desarrollo del trabajo. Dicho dron permanece junto al barco en toda su trayectoria de aproximación.

Una vez detectado que su trayectoria se aproxima al puerto mencionado, se procede a la estimación de su velocidad con el código **estima_velo.py**, en el cual se introducen los datos de las coordenadas GPS (longitud y latitud) y el tiempo. A continuación, el código estima la velocidad de dicho barco obteniendo lo que se puede ver en la Figura 4-1.

```
1  barcos1,10.80
2  barcos1,11.23
3  barcos1,12.63
4  barcos1,14.18
5  barcos1,14.15
6  barcos1,45.00
7  barcos1,13.31
8  barcos1,8.14
9  barcos1,7.80
10 barcos1,7.52
11 barcos1,3.43
12 barcos1,2.61
13 barcos1,1.90
```

Figura 4-1 Ejemplo velocidades estimadas (fuente: propia)

Prosiguiendo con los pasos a seguir, se lleva a cabo la detección de las posibles anomalías en la velocidad del barco en cuestión por medio del código **det_anom.py**. En el caso de que se hayan detectado velocidades anómalas se imprime lo que se puede ver en la Figura 4-2.

```
✓ import csv ...
... Velocidades anómalas encontradas:
    Velocidad anómala 45.0 encontrada en el barco barcos1.
```

Figura 4-2 Ejemplo velocidad anómala encontrada (fuente: propia)

Dado que se ha detectado una anomalía en la velocidad del barco 1, se procede a denominar su trayectoria como anómala para poder llevar a cabo la interceptación por parte del dron. Esto se hace por medio del código **intercept.py** que, después del estudio de la trayectoria del objetivo a interceptar, muestra la trayectoria de interceptación que debe seguir dron como vemos en la Figura 4-3.

```
-8.7125,42.3949,0,1672677741  
-8.731421244424418,42.38882924201162,0,1672678071  
-8.749724798170673,42.38298613231545,0,1672678390  
-8.766868696288961,42.37714998452431,0,1672678692  
-8.783982879806851,42.37043509554288,0,1672679002  
-8.78429699999731,42.36868500078953,0,1672679312  
-8.768355871134029,42.3766695221602,0,1672679620  
-8.752541741348658,42.3842953723801,0,1672679922  
-8.739904999996714,42.39106800079358,0,1672680231  
-8.729994999996588,42.3955620007944,0,1672680532  
-8.719724999996462,42.398793000795,0,1672680841  
-8.70524499999628,42.40076500079537,0,1672681762  
-8.701571999996235,42.400062000795245,0,1672682071  
-8.698951999996206,42.39931800079513,0,1672682381
```

Figura 4-3 Ejemplo archivo trayectoria interceptación dron (fuente: propia)

Con los datos de la trayectoria de interceptación del dron y los datos de la trayectoria anómala del barco, se emplea el código `geojson_dron.py` y `geojson_trayec.py`, que convierten ambos archivos a un formato *GeoJSON* para que pueda ser representado posteriormente en *Kepler.gl*.

4.3 Representación y validación en *Kepler.gl*

Finalmente, los archivos *GeoJSON* de la trayectoria de interceptación del dron y de la trayectoria anómala del barco, se pueden representar de manera visual la forma en la que el dron actuaría en este tipo de situaciones (ver Figuras 4-4, 4-5 y 4-6) y corroborar la validación de los archivos de datos obtenidos gracias a la plataforma *Kepler.gl*.

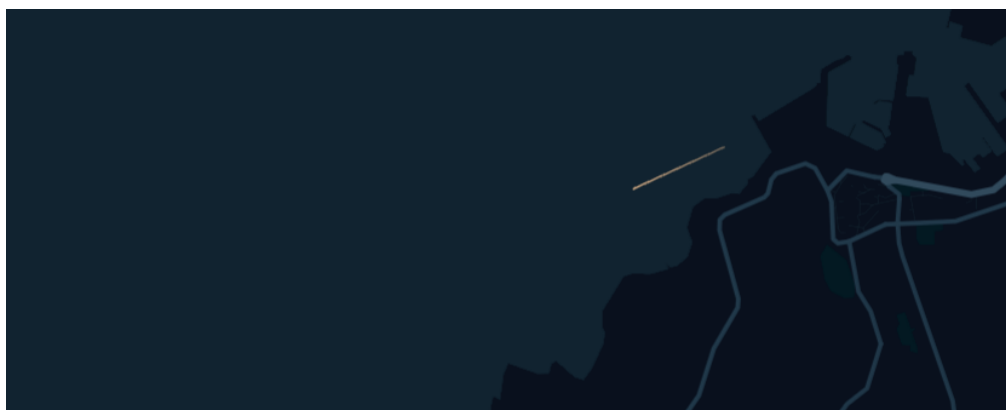


Figura 4-4 Representación visual del dron saliendo del puerto (fuente: propia)



Figura 4-5 Representación visual del dron aproximándose al barco sospechoso (fuente: propia)



Figura 4-6 Representación visual del dron interceptando al barco sospechoso (fuente: propia)

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

A lo largo del trabajo, se han desarrollado una serie de técnicas que podrían ser utilizadas en la implementación de un simulador que, mediante el uso de mecanismos de aprendizaje por refuerzo (por ejemplo), pudiera utilizarse para entrenar una inteligencia artificial que forme parte del sistema de control de un vehículo USV de vigilancia portuaria. Estas técnicas incluyen el análisis estadístico de los datos de velocidad, la detección de anomalías en los patrones de comportamiento y el cálculo del curso de interceptación mediante el mecanismo de persecución directa.

En este trabajo se han dado los primeros pasos para la implementación de ese entorno simulado, en un escenario muy acotado: el entorno geográfico se limita al Puerto de Marín, sólo hay una embarcación "objetivo" y el dron (no hay terceros buques implicados), se han considerado únicamente anomalías relacionadas con la velocidad (sin considerar cambios de rumbo u otros factores), se asume que se dispone de sensores que permitan estimar con precisión la posición del objetivo cada cierto tiempo, no se han implementado mecanismos de detección de colisiones, etc.

A título personal, este trabajo ha sido extremadamente enriquecedor para mí ya que me ha introducido en el mundo de los drones y sus aplicaciones en el mundo militar, la programación, detección de anomalías, inteligencia artificial y análisis estadísticos. Me ha permitido profundizar en el conocimiento de estas tecnologías y entender su gran potencial para desarrollar sistemas autónomos de vigilancia y control en diferentes ámbitos.

En particular, he podido apreciar la importancia de la combinación de estas tecnologías para lograr sistemas autónomos eficientes y seguros, especialmente en el ámbito de la seguridad y defensa. Además, he podido experimentar de primera mano el potencial de la inteligencia artificial para la identificación de patrones y la toma de decisiones en tiempo real, lo que puede ser fundamental para la prevención de amenazas y la protección de infraestructuras críticas.

5.2 Líneas futuras

La simulación de un sistema autónomo de vigilancia del Puerto de Marín es un paso importante para mejorar la seguridad y eficiencia de la vigilancia marítima. Sin embargo, hay un gran potencial para seguir mejorando y expandiendo el uso de sistemas autónomos de vigilancia en el futuro.

En primer lugar, se pueden explorar y desarrollar nuevas técnicas y tecnologías para mejorar la eficacia y el rendimiento del sistema autónomo de vigilancia. La incorporación de inteligencia artificial y aprendizaje automático puede permitir que el sistema autónomo de vigilancia tenga una capacidad de

detección más sofisticada y pueda identificar patrones de comportamiento sospechosos de manera más eficiente. Además, la integración de sensores y cámaras avanzados puede permitir que el sistema autónomo de vigilancia tenga una mayor capacidad de detección y pueda operar en diferentes condiciones ambientales.

Otra línea futura de investigación es la exploración de posibles aplicaciones de sistemas autónomos de vigilancia en otras áreas, como la vigilancia de áreas forestales o urbanas. Los sistemas autónomos de vigilancia pueden ser una herramienta valiosa para mejorar la seguridad y la eficiencia en diferentes contextos, y su potencial de aplicación puede extenderse mucho más allá del Puerto de Marín.

Además, es importante considerar la privacidad y los derechos civiles al utilizar sistemas autónomos de vigilancia. Se deben desarrollar medidas para garantizar la privacidad y protección de los derechos civiles de las personas en relación con la utilización de sistemas autónomos de vigilancia. La transparencia y la rendición de cuentas en la implementación de sistemas autónomos de vigilancia también deben ser una consideración clave.

No obstante, la seguridad de los sistemas autónomos de vigilancia es crucial para garantizar su eficacia y protegerlos contra posibles ciberataques o sabotajes. Se debe trabajar en el desarrollo de medidas de seguridad avanzadas y en la integración de sistemas autónomos de vigilancia con otros sistemas de seguridad marítima existentes para mejorar la eficiencia y la coordinación de la vigilancia.

En el contexto de la Armada Española, el sistema autónomo de vigilancia del Puerto de Marín podría tener una gran aplicación y potencial. Algunas líneas futuras para su implementación en la Armada Española podrían ser:

- Integración con sistemas de vigilancia existentes: el sistema autónomo de vigilancia podría ser integrado con sistemas de vigilancia ya existentes en la Armada Española, como radares y cámaras de vigilancia, para mejorar la eficacia y coordinación de la vigilancia marítima.
- Desarrollo de sistemas autónomos de vigilancia para otros puertos: puede servir como modelo para desarrollar sistemas similares para otros puertos de la Armada Española. Esto permitiría una vigilancia más eficiente y efectiva en diferentes puertos y zonas marítimas.
- Uso de sistemas autónomos de vigilancia para misiones de búsqueda y rescate: pueden ser utilizados para misiones de búsqueda y rescate en caso de emergencia en el mar. Su capacidad de operar en diferentes condiciones ambientales y de identificar objetos y personas a distancia los hace ideales para esta tarea.
- Integración de sistemas autónomos de vigilancia en operaciones militares: pueden tener aplicaciones en operaciones militares, como la detección y seguimiento de objetivos enemigos. La integración de sistemas autónomos de vigilancia con otras tecnologías militares puede aumentar la eficacia y precisión de las operaciones.
- Desarrollo de sistemas autónomos de vigilancia avanzados: la investigación y el desarrollo de sistemas autónomos de vigilancia más avanzados, que incluyan inteligencia artificial y aprendizaje automático, pueden mejorar significativamente la capacidad de detección y la eficiencia del sistema. La Armada Española podría liderar el desarrollo de sistemas autónomos de vigilancia avanzados en el ámbito militar.

6 BIBLIOGRAFÍA

- [1] «Embention,» 09 01 2018. [En línea]. Available: <https://www.embention.com/es/news/modos-de-control-con-uavs/>.
- [2] R. Fernández, «Eagle Dron,» 2022. [En línea]. Available: <https://www.eagledron.es/los-10-usos-principales-de-los-drones/>.
- [3] P. Scharre, *Army of None: Autonomus Weapons and the Future of War*, 978-0393356588: WW Norton & Co, 2018.
- [4] P. Belmonte, «Revista Ejército,» 2021. [En línea]. Available: <https://www.revistaejercitos.com/2021/08/12/el-empleo-de-drones-en-las-unidades-de-reconocimiento/>.
- [5] J. J. Fernández, «El Confidencial,» 16 05 2021. [En línea]. Available: https://www.elconfidencial.com/tecnologia/2021-05-16/drones-tecnologia-militar-black-hornet-rq11-raven_3081456/.
- [6] «Maldita,» 04 11 2022. [En línea]. Available: <https://maldita.es/malditatecnologia/20221104/drones-kamikaze-uso-militar/>.
- [7] «El Español,» 09 10 2022. [En línea]. Available: https://www.elespanol.com/omicrono/defensa-y-espacio/20221009/preciso-sistema-controlar-enjambres-drones-atacar-objetivos/708179497_0.html.
- [8] «Gobierno de España. Ministerio de Defensa,» 09 09 2022. [En línea]. Available: https://ume.defensa.gob.es/Noticias/2022/09/Noticias/rpas_eu_.html.
- [9] G. Dubois, «Aviacionline,» 04 03 2022. [En línea]. Available: <https://www.aviacionline.com/2022/03/evolution-el-futuro-de-los-drones-de-general-atomics/>.
- [10] «Aragón Noticias,» 17 10 2022. [En línea]. Available: <https://www.cartv.es/aragonnoticias/noticias/el-uso-de-los-drones-aumenta-aunque-desde-el-sector-critican-el-coste-de-las-tasas-y-la-actual-legislacion-14061>.

- [11] «Hispanaviación,» 26 04 2022. [En línea]. Available: <https://www.hispaviacion.es/los-drones-militares-del-futuro-podrian-llevar-sistemas-microondas-para-derribar-a-otros-drones/>.
- [12] J. M. Navarro, «Defensa,» 20 10 2022. [En línea]. Available: <https://www.defensa.com/industria/el-euromale-supera-otro-hito>.
- [13] Á. d. Santos, «La Razón,» 19 05 2022. [En línea]. Available: <https://www.larazon.es/espana/20220519/axiaix4vubfqjcvkohbtjuqvga.html>.
- [14] E. Pérez, «Xataka,» 27 01 2022. [En línea]. Available: <https://www.xataka.com/drones/euromale-asi-futurista-dron-militar-que-espana-invertira-1-900-millones-euros>.
- [15] «BBC News,» 28 07 2022. [En línea]. Available: <https://www.bbc.com/mundo/noticias-internacional-62279288>.
- [16] N. García, «Internacional,» 11 03 2022. [En línea]. Available: <https://www.epe.es/es/internacional/20220311/bayraktar-tb2-dron-ucrania-rusia-13354730>.
- [17] «ABC Tecnología,» 20 04 2022. [En línea]. Available: https://www.abc.es/tecnologia/informatica/soluciones/abci-orlan-10-dron-espia-ruso-mofa-ejercito-ucraniano-202204200125_noticia.html.
- [18] «El Radar,» 20 07 2021. [En línea]. Available: <https://www.elradar.es/guerras-de-robots-los-ugv-en-el-campo-de-batalla/>.
- [19] L. Drusila, «Infodefensa,» 01 02 2023. [En línea]. Available: <https://www.infodefensa.com/texto-diario/mostrar/4159440/marker-ugv-ruso-enfrentara-leopard-ucrania>.
- [20] «El Radar,» 08 11 2022. [En línea]. Available: <https://www.elradar.es/ucrania-ataca-con-usvs-kamikazes-las-bases-rusas-ya-no-son-seguras/>.
- [21] K. Atherton, «Brookings,» 04 08 2021. [En línea]. Available: <https://www.brookings.edu/techstream/loitering-munitions-preview-the-autonomous-future-of-warfare/>.
- [22] J. Herrero, «La Razón,» 14 09 2022. [En línea]. Available: <https://www.larazon.es/internacional/20220914/widk6cuf25e3njawykcukxyvcq.html>.
- [23] «Gobierno de España. Ministerio de Defensa,» [En línea]. Available: <https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspañola/buquesaeronaves/prefLang-es/03flotilla-aeronaves-aeronaves-no-tripuladas>.
- [24] Á. Luis, «La Razón,» 03 12 2021. [En línea]. Available: <https://www.larazon.es/espana/20211203/3feg3dmgcffyzk3zhuri75kpl4.html>.
- [25] L. Drusila, «Infodron,» 31 10 2022. [En línea]. Available: <https://www.infodron.es/texto-diario/mostrar/3939593/usv-operan-espana-vendaval-kaluga-sead-23>.
- [26] B. Carrasco, «Infodefensa,» 25 09 2021. [En línea]. Available: <https://www.infodefensa.com/texto-diario/mostrar/3208439/asi-vendaval-navantia-vehiculo-no-tripulado-superficie-operativo-puerto-ceuta>.
- [27] «Repsol. Energía Innovación,» [En línea]. Available: <https://www.repsol.com/es/energia-innovacion/energia-futuro/tecnologia-innovacion/inteligencia->

- artificial/index.cshtml?gclid=CjwKCAiAioifBhAXEiwApzCztlr_NL64TB7zm1UnUhVAXVOL6BbDj0YfQC31Z5ASlBuNrnBxx73UzhoCvTkQAvD_BwE.
- [28] «Noticias. Parlamento Europeo,» 08 09 2020. [En línea]. Available: <https://www.europarl.europa.eu/news/es/headlines/society/20200827STO85804/que-es-la-inteligencia-artificial-y-como-se-usa>.
- [29] «Goom Spain,» [En línea]. Available: <https://www.goomspain.com/inteligencia-artificial-aplicada-a-drones/>.
- [30] C. C. Aggarwal, Neural Networks and Deep Learning, 978-0393608984: Springer; 1st ed., 2018.
- [31] C. Sanjuán, «Repositorio Comillas,» 2018. [En línea]. Available: <https://repositorio.comillas.edu/xmlui/handle/11531/23014>.
- [32] A. Peña, «El Ideal Gallego,» 25 07 2021. [En línea]. Available: <https://www.elidealgalego.com/texto-diario/mostrar/3063757/drones-092-estaran-equipados-camaras-termicas-reconoceran-rostros-300-metros>.
- [33] «ESSS,» [En línea]. Available: <https://www.esss.co/es/biblioteca-tecnica/articulo-internet-de-las-cosas-aplicada-a-drones/>.
- [34] «TIBCO,» [En línea]. Available: <https://www.tibco.com/es/reference-center/what-is-supervised-learning>.
- [35] «TIBCO,» [En línea]. Available: <https://www.tibco.com/es/reference-center/what-is-unsupervised-learning>.
- [36] «IMF,» Blog de Tecnología, [En línea]. Available: <https://blogs.imf-formacion.com/blog/tecnologia/reglas-asociacion-big-data-202007/>.
- [37] «Aprende Machine Learning,» 24 12 2020. [En línea]. Available: <https://www.aprendemachinelearning.com/aprendizaje-por-refuerzo/>.
- [38] «IPS Intelligence,» [En línea]. Available: <https://www.ips-intelligence.com/es/maritime-intelligence>.
- [39] O. Tournon, «BoatIndustry,» 27 04 2022. [En línea]. Available: <https://www.boatindustry.es/noticias/40055/kaeonit-aplicaciones-de-deteccion-de-buques-para-la-gestion-portuaria>.
- [40] «Muy Interesante,» [En línea]. Available: <https://www.muyinteresante.es/tecnologia/4352.html>.
- [41] «Marine Traffic,» [En línea]. Available: <https://www.marinetraffic.com/en/ais/home/centerx:-23.5/centery:22.3/zoom:7>.
- [42] «Global Fishing Watch,» [En línea]. Available: <https://globalfishingwatch.org/es/nuestra-tecnologia/>.
- [43] «World Meteorological Organization,» 2018. [En línea]. Available: https://library.wmo.int/?lvl=notice_display&id=7471#.Y_6W3HbMK3A.
- [44] «Agencia Estatal de Meteorología,» [En línea]. Available: <https://www.aemet.es/es/portada>.

- [45] A. L. Mula, *Algoritmos de detección de anomalías y sus aplicaciones en el ámbito marítimo*, Universidad de Vigo: TFG, 2020-2021.
- [46] «Navemoción,» [En línea]. Available: <http://www.navemocion.com/astillero/sistema-identificacion.php>.
- [47] «Python,» [En línea]. Available: <https://www.python.org/>.
- [48] «OpenStreetMap,» [En línea]. Available: <https://www.openstreetmap.org/#map=6/40.007/-2.488>.
- [49] A. Morales, «Mapping GIS,» [En línea]. Available: <https://mappinggis.com/2021/04/openstreetmap-la-plataforma-de-mapas-libre-mas-grande-del-mundo/>.
- [50] «Kepler,» [En línea]. Available: <https://kepler.gl/>.
- [51] «Mapbox,» [En línea]. Available: <https://docs.mapbox.com/mapbox-gl-js/api/>.
- [52] «Geomapik,» [En línea]. Available: <https://www.geomapik.com/webmapping-gis/kepler-gl-mapbox-mapas-interactivos-spatial-data/>.
- [53] «GeoJSON,» [En línea]. Available: <https://geojson.io/#map=2/0/20>.
- [54] «Stack overflow,» [En línea]. Available: <https://stackoverflow.com/questions/4913349/haversine-formula-in-python-bearing-and-distance-between-two-gps-points>.
- [55] R. Andrés, «Computer Hoy,» 17 Junio 2018. [En línea]. Available: <https://computerhoy.com/listas/tecnologia/mejores-drones-camara-que-puedes-comprar-2018-255781>.
- [56] «VigoÉ,» 01 Octubre 2021. [En línea]. Available: <https://www.vigoe.es/vigo/maritima/el-sea23-vehiculo-marino-no-tripulado-construido-en-vigo-capta-la-atencion-de-la-otan/>.
- [57] A. Conde, «Global Strategy,» [En línea]. Available: <https://global-strategy.org/los-grandes-vehiculos-submarinos-autonomos-el-sueno-de-julio-verne/>.
- [58] «Infodron,» 05 02 2023. [En línea]. Available: <https://www.infodron.es/texto-diario/mostrar/3529077/turquia-ucrania-acuerdan-coproducir-drones-bayraktar-tb2>.
- [59] P. Estañol, «RFI,» 17 10 2022. [En línea]. Available: <https://www.rfi.fr/es/europa/20221020-c%C3%B3mo-son-los-shahed-136-los-drones-iran%C3%ADes-kamikazes-que-rusia-usa-en-ucrania>.
- [60] «US Defense Story,» 03 11 2020. [En línea]. Available: <https://usdefensestory.com/cjtf-hoa-cg-visits-cadj/>.
- [61] «Infodefensa,» 22 02 2023. [En línea]. Available: <https://www.infodefensa.com/texto-diario/mostrar/3360206/marine-instruments-recibe-financiacion-defensa-mejorar-rpas-airfox>.
- [62] R. Holgado, «20minutos,» 24 01 2023. [En línea]. Available: <https://www.20minutos.es/tecnologia/actualidad/rusia-escondia-un-arma-secreta-asi-es-marker-el-robot-de-combate-que-usara-contr-ucrania-5094739/>.

- [63] A. Barro, «El Confidencial,» 29 09 2021. [En línea]. Available: https://www.elconfidencial.com/mundo/2021-09-29/drones-afganistan-nueva-guerra-remota-joe-biden_3296947/.
- [64] «El Español,» 12 11 2022. [En línea]. Available: https://www.elespanol.com/omicron/tecnologia/20221118/ejercito-drones-kamikazes-navales-ucrania-blindar-buques/718928228_0.html.
- [65] «The Baltic Time,» 03 Agosto 2018. [En línea]. Available: https://www.baltictimes.com/estonia__finland__latvia_looking_to_develop_unmanned_land_systems_in_framework_of_pesco/.
- [66] M. Fernández, «El Español,» 2022. [En línea]. Available: https://www.elespanol.com/omicron/tecnologia/20220525/impactante-video-dron-suicida-ucraniano-tanque-ruso/675182554_0.html, 2022.

ANEXO I: IMPLICACIONES SOCIALES, Y/O ECONÓMICAS, Y/O AMBIENTALES

Los drones son una tecnología emergente que tiene muchas aplicaciones en diferentes campos, incluyendo la vigilancia, el transporte, la fotografía, la agricultura y la investigación científica. En el ámbito militar, los drones se utilizan cada vez más para misiones de vigilancia, reconocimiento y ataque, lo que ha llevado a debates éticos y legales sobre el uso de la tecnología en conflictos armados. Sin embargo, también existen muchas aplicaciones civiles de los drones que pueden tener un gran impacto social y económico.

En cuanto al impacto social, los drones pueden utilizarse para llevar a cabo tareas peligrosas como el rescate de personas en áreas remotas o peligrosas, la evaluación de daños después de un desastre natural o el mantenimiento de infraestructuras de difícil acceso, como puentes o torres de energía. Además, los drones pueden mejorar la seguridad pública al ayudar a las fuerzas del orden a vigilar áreas urbanas o a controlar el tráfico en las carreteras.

En términos de impacto económico, la industria de drones está en pleno crecimiento, y se espera que tenga un valor de mercado de varios miles de millones de euros en los próximos años. Las empresas que se dedican a desarrollar drones están creando empleos en campos como la ingeniería, la programación y la fabricación, y también están impulsando la innovación en áreas como la inteligencia artificial y la robótica. Además, los drones pueden utilizarse para mejorar la eficiencia en sectores como la agricultura, el transporte y la logística, lo que puede reducir los costes y aumentar la productividad.

Sin embargo, el uso de drones también puede tener implicaciones negativas en términos de empleo, ya que algunas tareas que antes se realizaban manualmente pueden ser automatizadas con drones. Por ejemplo, los drones pueden utilizarse para entregar paquetes, lo que podría afectar a los trabajadores de la entrega. Además, el uso de drones para fines de vigilancia y control puede plantear preocupaciones sobre la privacidad y la seguridad, lo que requiere un cuidadoso equilibrio entre la tecnología y la protección de los derechos individuales.

Los drones tienen el potencial de tener un gran impacto social y económico, tanto en el ámbito civil como en el militar. A medida que la tecnología continúa evolucionando y desarrollándose, es importante considerar cuidadosamente las implicaciones sociales y económicas de su uso, y asegurarse de que se utilice de manera ética y responsable para beneficio de la sociedad en general.

ANEXO II: REFLEXIONES ÉTICAS Y SOCIALES

Los drones y otras tecnologías autónomas tienen el potencial de mejorar significativamente la precisión y la eficiencia de las operaciones militares, pero su uso también plantea preocupaciones sobre la falta de control humano en la toma de decisiones y la posibilidad de errores fatales.

Por lo tanto, se puede destacar la gran importancia de los drones en el mundo militar y su impacto en la forma en que se llevan a cabo las operaciones militares. Sin embargo, existe la necesidad de una regulación ética y legal adecuada para asegurar que estas armas se usen de manera responsable.

Estos sistemas pueden tener varios inconvenientes, como el riesgo para la seguridad pública, ya que pueden ser utilizados para realizar actividades ilegales o peligrosas; la interferencia con la privacidad, ya que pueden ser utilizados para vigilar a personas sin su consentimiento; el riesgo para la aviación, ya que pueden causar accidentes aéreos si colisionan con otros aviones o helicópteros; el riesgo para la fauna y el medioambiente.

Pueden suponer problemas legales, ya que el uso de drones puede violar leyes y regulaciones, especialmente en cuanto a los derechos de privacidad y la seguridad aérea; también tienen como inconveniente el coste, ya que pueden llegar a ser muy costosos de comprar y mantener, requieren personal capacitado y licencias para operarlos; por último, se podría decir que causan problemas de ruido, ya que generan un ruido molesto para las personas cercanas a su operación.

Una propuesta para mejorar el uso de los drones es establecer regulaciones claras y coherentes para su uso, que incluyan requisitos para la seguridad aérea, privacidad y responsabilidad por daños, proporcionar capacitación y certificación para los operadores de drones para garantizar que manejan los drones de manera segura y responsable.

Otra manera de mejorar el uso de los drones sería realizar investigaciones para evaluar los impactos ambientales, sociales y económicos de los drones y utilizar esta información para mejorar la regulación y la política. Así como trabajar con la comunidad y organizaciones locales para desarrollar políticas y prácticas sostenibles para el uso de drones, no permitir su uso en lugares prohibidos como aeropuertos, prisiones, hospitales y otros lugares donde puedan causar problemas de seguridad.

En los últimos años, ha habido un creciente debate sobre los sistemas autónomos con capacidad de matar, también conocidos como "robots asesinos". La preocupación principal es que estos sistemas podrían tomar decisiones de vida o muerte sin la intervención humana, lo que plantea preocupaciones éticas y morales importantes. La campaña "*Stop Killer Robots*" es una iniciativa internacional que busca prohibir el desarrollo y uso de estos sistemas autónomos en conflictos armados.

Es importante tener en cuenta que, aunque un sistema autónomo pueda ser entrenado para tomar decisiones complejas, siempre debe haber una supervisión humana para garantizar que las decisiones tomadas sean éticas y legales.

La campaña "*Stop Killer Robots*" destaca la importancia de involucrar a la sociedad en este debate sobre la tecnología militar, y de garantizar que se tomen en cuenta los aspectos éticos y morales de la tecnología antes de ser utilizada. Es esencial que se desarrollen regulaciones adecuadas para garantizar que la tecnología se utilice de manera ética y responsable.

En el caso de los sistemas autónomos de vigilancia, como el que se ha desarrollado en este trabajo, pueden tener un impacto significativo en la seguridad y el bienestar de las personas. Este sistema está diseñado para simular el acercamiento a embarcaciones potencialmente sospechosas de manera autónoma, pero sin realizar acciones ofensivas. Esto significa que el sistema no tiene capacidad para tomar decisiones de vida o muerte y, por lo tanto, no plantea problemas éticos o morales relacionados con los sistemas autónomos con capacidad de matar, pero sí es un aspecto clave a tener en cuenta.

ANEXO III: DISTRIBUCIÓN DE HORAS

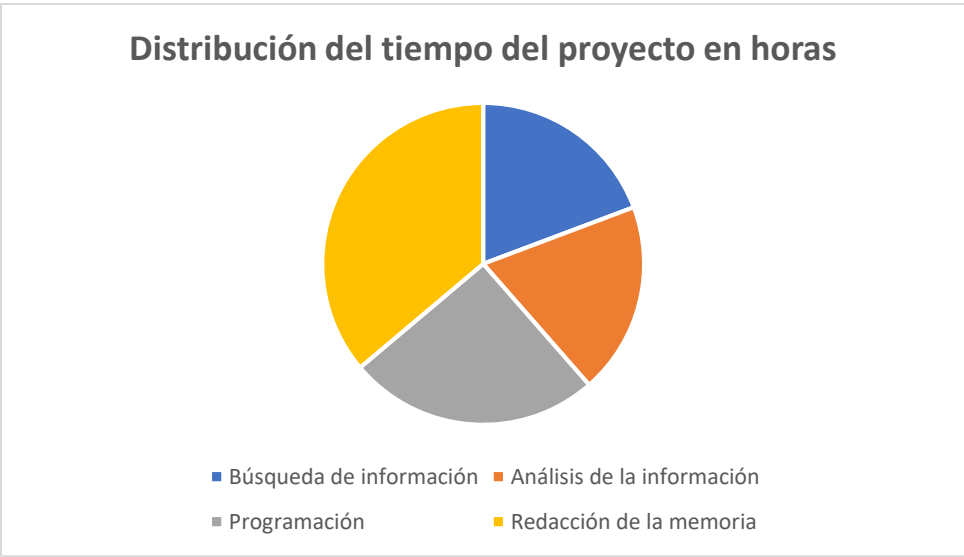


Figura A-1 Gráfica distribución del tiempo del proyecto

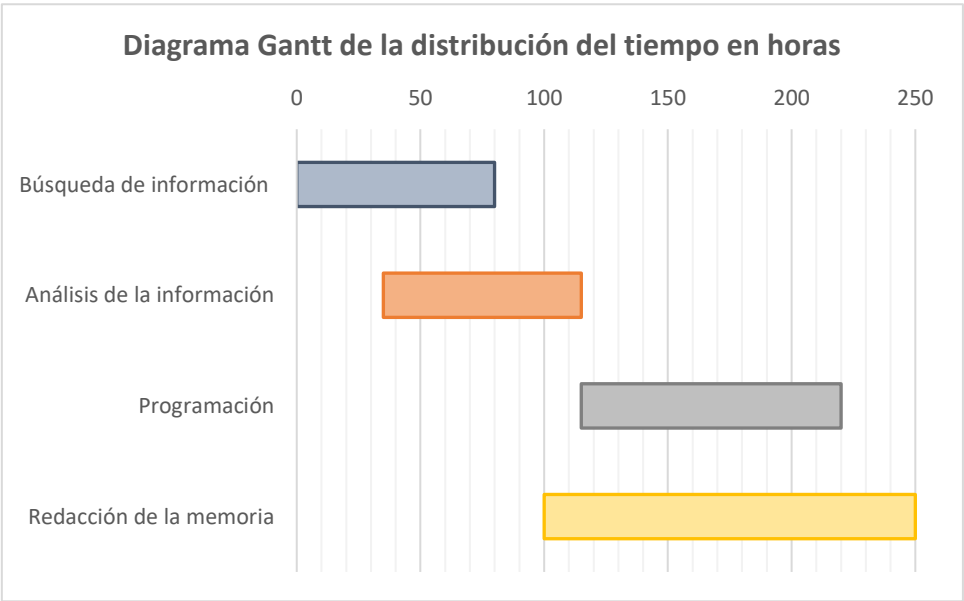


Figura A-2 Diagrama Gantt distribución del tiempo del proyecto