



## Trabajo de Fin de Grado

---

**Vessel Manager. Un planificador de  
atraque de buques en puertos marítimos**

*Vessel Manager. A vessel berthing scheduler in maritime  
ports*

Felipe Gómez Fuentes

---

La Laguna, 24 de mayo de 2024

D. **Christopher Expósito Izquierdo**, profesor Ayudante Doctor adscrito al Departamento Ingeniería Informática y de Sistemas de la Universidad de La Laguna como tutor

D. **Israel López Plata**, profesor Ayudante Doctor adscrito al Departamento Ingeniería Informática y de Sistemas de la Universidad de La Laguna como cotutor

## **C E R T I F I C A N**

Que la presente memoria titulada:

*"Vessel Manager. Un planificador de atraque de buques en puertos marítimos"*

ha sido realizada bajo su dirección por Don **Felipe Gómez Fuentes**.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 24 de mayo de 2024

# Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido de manera significativa en la realización de este trabajo de fin de grado.

En primer lugar, deseo agradecer profundamente a mis distinguidos tutores académicos, Christopher Expósito Izquierdo e Israel López Plata, por su invaluable orientación, apoyo y dedicación a lo largo de este proyecto. Su experiencia, sabiduría y compromiso han sido pilares fundamentales para mi desarrollo académico y la culminación exitosa de este trabajo.

También quiero extender mi gratitud hacia mi familia y amigos por su constante apoyo y aliento durante todo mi recorrido universitario. Sus palabras de ánimo, comprensión y motivación han sido una fuente de fortaleza en los momentos más desafiantes.

Por último, deseo reconocer y agradecer a todos aquellos que, de una forma u otra, han formado parte de mi trayectoria académica. Mi reconocimiento también se extiende a todas las personas que, de manera indirecta, han contribuido a mi formación mediante sus investigaciones, libros, artículos y recursos disponibles.

Su apoyo y colaboración han sido fundamentales en la realización de este trabajo, y estoy sinceramente agradecido por su invaluable contribución.

## **Resumen**

*Vessel Manager es una herramienta integral diseñada para la gestión eficiente de operaciones marítimas, centrándose en la programación de atraques, planificación de buques y coordinación de operaciones portuarias. Esta plataforma ofrece funcionalidades clave para organizar y optimizar la llegada de los buques de acuerdo con la disponibilidad de atraques en los puertos.*

*La aplicación permite a los usuarios programar y coordinar las operaciones de atraque, gestionar la flota de barcos y generar cuadros de mando con indicadores clave para evaluar el rendimiento y la eficiencia de las operaciones portuarias. Además, facilita la asignación de recursos, seguimiento de inventarios y cumplimiento de regulaciones marítimas para garantizar una operación segura y eficiente.*

*Dirigida a empresas navieras, operadores de puertos y autoridades portuarias, Vessel Manager ofrece una interfaz intuitiva y herramientas de análisis avanzadas para una gestión efectiva de operaciones portuarias, permitiendo una mejor planificación, optimización de recursos y reducción de costos operativos.*

**Palabras clave:** gestión de operaciones marítimas, programación de atraques, planificación de barcos, coordinación portuaria, asignación de recursos.

## **Abstract**

*Vessel Manager* is a comprehensive tool designed for efficient maritime operations management, focusing on berth scheduling, vessel planning, and port operations coordination. This platform offers key functionalities to organize and optimize the arrival of vessels according to berth availability in ports.

The application allows users to schedule and coordinate berthing operations, manage the fleet of vessels, and generate dashboards with key indicators to assess the performance and efficiency of port operations. Additionally, it facilitates resource allocation, inventory tracking, and compliance with maritime regulations to ensure safe and efficient operation.

Targeted at shipping companies, port operators, and port authorities, *Vessel Manager* provides an intuitive interface and advanced analytics tools for effective port operations management, enabling better planning, resource optimization, and operational cost reduction.

**Keywords:** *maritime operations management, berth scheduling, vessel planning, port operations coordination, resource allocation.*

# Índice general

|  |           |
|--|-----------|
| <b>1. Introducción</b>   | <b>1</b>  |
| 1.1. Justificación del proyecto                                | 2         |
| 1.2. Objetivos del proyecto                                    | 3         |
| <b>2. Motivación y Estado del arte</b>                         | <b>4</b>  |
| 2.1. Introducción  | 4         |
| 2.2. Antecedentes históricos y problemas actuales              | 4         |
| 2.3. Soluciones tradicionales y avances recientes              | 5         |
| 2.3.1. Sistemas de Control de Tráfico Marítimo (VTS)           | 5         |
| 2.3.2. Sistemas de Información Portuaria (PIS)                 | 7         |
| 2.4. Tecnologías emergentes en gestión de atraques             | 11        |
| 2.4.1. Aplicaciones de uso comercial                           | 12        |
| 2.4.2. Sistemas de Control de Tráfico Marítimo Avanzados (VTS) | 13        |
| 2.4.3. Soluciones basadas en IoT y Blockchain                  | 15        |
| 2.4.4. Sistemas de gestión de flotas marítimas                 | 16        |
| 2.4.5. Awake.ai  | 17        |
| 2.5. Perspectivas futuras y tendencias                         | 18        |
| <b>3. Tecnologías y arquitectura de software</b>               | <b>19</b> |
| 3.1. Arquitectura del sistema                                  | 19        |
| 3.1.1. Componentes principales                                 | 21        |
| 3.1.2. Despliegue y orquestación                               | 21        |
| 3.2. Tecnologías utilizadas                                    | 21        |
| 3.2.1. MongoDB   | 22        |
| 3.2.2. Spring y Spring Boot                                    | 22        |
| 3.2.3. Vue.js y Vuetify  | 22        |
| 3.2.4. Docker y Docker Hub                                     | 22        |
| 3.2.5. Git y GitHub  | 22        |
| 3.3. Estructura de la aplicación                               | 23        |
| 3.3.1. Servidor (Back-end)                                     | 23        |
| 3.3.2. Interfaz de usuario (Front-end)                         | 24        |
| 3.3.3. Algoritmo de organización de buques                     | 26        |
| <b>4. Caso práctico</b>  | <b>28</b> |
| 4.1. Despliegue del sistema                                    | 28        |
| 4.2. Funcionamiento del sistema                                | 29        |
| 4.2.1. Inicio  | 29        |
| 4.2.2. Lista de buques   | 30        |
| 4.2.3. Lista de atraques                                       | 34        |

|  |           |
|--|-----------|
| 4.2.4. Página de planificaciones . . . . . | 37        |
| 4.3. Caso práctico . . . . .               | 39        |
| <b>5. Presupuesto</b>                      | <b>42</b> |
| <b>6. Conclusiones</b>                     | <b>43</b> |
| 6.1. Líneas futuras . . . . .              | 43        |
| <b>7. Conclusions</b>                      | <b>45</b> |
| 7.1. Future Directions . . . . .           | 45        |
| <b>Bibliografía</b>                        | <b>46</b> |

# Índice de figuras

|  |    |
|--|----|
| 1.1. Representación de datos actuales y previstos de buques comerciales desde el año 2000 al 2026. . . . . | 2  |
| 2.1. Representación de un Sistemas de Control de Tráfico Marítimo. . . . .                                 | 5  |
| 2.2. VTS del Puerto de Singapur. . . . .   | 6  |
| 2.3. VTS del Puerto de Rotterdam. . . . .  | 7  |
| 2.4. Representación de GisWeb. . . . .   | 8  |
| 2.5. Sistema de Gestión de Terminal (TOS). . . . .   | 8  |
| 2.6. Ilustración del Sistema de Gestión de Flotas (FMS). . . . .   | 9  |
| 2.7. Sistema de Gestión de Almacenes (WMS). . . . .  | 10 |
| 2.8. Sistema de Gestión de Tráfico (TMS). . . . .  | 11 |
| 2.9. Interfaz de usuario de Awake.ai. . . . .  | 12 |
| 2.10 Interfaz de usuario de Berth Planner. . . . .   | 13 |
| 2.11 Interfaz de usuario de PortXchange en el puerto de Rotterdam. . . . .                                 | 13 |
| 2.12 VTS modernos. . . . .   | 14 |
| 2.13 Integración con plataformas de gestión de atraques. . . . .   | 14 |
| 2.14 Análisis de big data en tiempo real para control y monitorización de turistas. . . . .                | 15 |
| 2.15 Plataforma Maritime Blockchain Labs. . . . .  | 16 |
| 2.16 Plataforma Fleet Complete. . . . .  | 16 |
| 2.17 Interfaz de usuario de Awake.ai. . . . .  | 17 |
| 3.1. Representación de la arquitectura hexagonal. . . . .  | 20 |
| 3.2. Representación de la arquitectura de la aplicación. . . . .   | 20 |
| 3.3. Estructura del back-end. . . . .  | 23 |
| 3.4. Estructura del front-end. . . . .   | 25 |
| 3.5. Diseño casos de uso. . . . .  | 26 |
| 4.1. Aplicación entera desplegada con Docker Compose. . . . .  | 29 |
| 4.2. Página de inicio. . . . .   | 29 |
| 4.3. Página de inicio con el menú desplegado. . . . .  | 30 |
| 4.4. Página de lista de buques. . . . .  | 31 |
| 4.5. Página para añadir barco . . . . .  | 31 |
| 4.6. Página para añadir operaciones . . . . .  | 32 |
| 4.7. Página para actualizar barcos . . . . .   | 32 |
| 4.8. Página actualizar barcos añadiendo operaciones . . . . .  | 33 |
| 4.9. Página para eliminar barcos . . . . .   | 33 |
| 4.10 Página para ver detalles del buque . . . . .  | 34 |
| 4.11 Página de lista de atraques. . . . .  | 34 |
| 4.12 Página para añadir atraque . . . . .  | 35 |
| 4.13 Página del para actualizar atraque . . . . .  | 35 |

|      |   |    |
|------|---|----|
| 4.14 | Página para eliminar ataque . . . . .                           | 36 |
| 4.15 | Página para ver detalles del ataque . . . . .                   | 36 |
| 4.16 | Página de la lista de planificaciones. . . . .                  | 37 |
| 4.17 | Página de para añadir una planificación . . . . .               | 37 |
| 4.18 | Página para actualizar el nombre de una planificación . . . . . | 38 |
| 4.19 | Página para eliminar una planificación . . . . .                | 38 |
| 4.20 | Acción del botón See Dashboard . . . . .                        | 39 |
| 4.21 | Lista de buques. . . . .  | 40 |
| 4.22 | Lista de ataques. . . . .                                       | 40 |
| 4.23 | Cuadro de mando de la planificación. . . . .                    | 41 |
| 4.24 | Lista de planificaciones. . . . .                               | 41 |

# Capítulo 1

## Introducción

La gestión eficiente de buques en atraques es un aspecto crucial en la operación portuaria, donde la toma de decisiones efectiva puede marcar la diferencia en la competitividad y rentabilidad de las empresas navieras y operadores portuarios. La programación de atraques, la planificación de buques y la coordinación de operaciones marítimas son procesos complejos que requieren una atención meticulosa para garantizar un flujo de trabajo fluido y rentable.

Algunos datos respaldan la importancia de esta gestión son los siguientes:

- **Volumen de operaciones diarias en puertos.** Los puertos manejan un gran volumen de operaciones diariamente. Según datos de la Asociación Internacional de Puertos (IAPH)<sup>1</sup>, los principales puertos del mundo manejan miles de toneladas de carga diariamente y atienden a cientos de buques. Por ejemplo, el Puerto de Shanghái, el puerto más grande del mundo en términos de volumen de carga, maneja millones de TEUs (contenedores equivalentes a veinte pies) cada año, lo que se traduce en miles de operaciones diarias. Por ejemplo, en el año 2022 se manejaron 41 millones de TEUs[1].
- **Flujo de barcos que llegan a puerto.** El flujo de barcos que llegan a los puertos puede ser significativo. Según datos de la Organización Marítima Internacional (OMI)<sup>2</sup>, millones de buques comerciales transitan por los puertos de todo el mundo cada año. Estos buques transportan una gran variedad de carga, desde contenedores hasta productos a granel y petróleo crudo. De hecho el transporte marítimo depende de los 2 millones de marinos que operan los buques mercantes del mundo<sup>3</sup>, que transportan más del 80 por ciento del comercio mundial en volumen, tal y como vemos en la figura 1.1.

---

<sup>1</sup><https://www.iaphworldports.org/>

<sup>2</sup><https://www.imo.org/es/>

<sup>3</sup><https://www.imo.org/es/MediaCentre/PressBriefings/pages/19-IMO-UNCTAD-.aspx>



Figura 1.1: Representación de datos actuales y previstos de buques comerciales desde el año 2000 al 2026.

- Impacto de la optimización en las operaciones.** Optimizar las operaciones portuarias puede tener múltiples beneficios. Por ejemplo, la programación eficiente de atraques puede reducir los tiempos de espera de los buques, minimizar la congestión en los puertos y mejorar la utilización de los recursos portuarios, lo que a su vez puede aumentar la productividad y reducir los costos operativos. Además, una mejor planificación y coordinación pueden mejorar la seguridad y reducir los riesgos de incidentes marítimos.

Estos datos muestran cómo la gestión eficiente de buques en atraques es esencial para garantizar la competitividad y rentabilidad de las empresas navieras y operadores portuarios, así como para mantener el flujo seguro y eficiente del comercio internacional.

## 1.1. Justificación del proyecto

Vessel Manager se presenta como una solución integral diseñada específicamente para abordar estos desafíos, ofreciendo una plataforma que facilita la gestión eficiente de las operaciones portuarias. Con un enfoque en la organización y optimización de la llegada de buques de acuerdo con la disponibilidad de atraques, esta herramienta proporciona funcionalidades clave para mejorar la planificación, la asignación de recursos y el cumplimiento de regulaciones marítimas.

La aplicación cuenta con un algoritmo avanzado que permite a los usuarios programar y coordinar las operaciones de atraque, gestionar la flota de barcos y generar informes detallados con indicadores clave de rendimiento. Además, ofrece herramientas de análisis avanzadas para una toma de decisiones informada y una mejor gestión de los recursos, lo que se traduce en una mayor eficiencia operativa y una reducción de costos.

Destinada a empresas navieras, operadores de puertos y autoridades portuarias, Vessel Manager se presenta como una solución intuitiva y potente que proporciona las herramientas necesarias para una gestión efectiva de las operaciones portuarias. Con una interfaz fácil de usar y funciones de análisis avanzadas, esta plataforma tiene como

objetivo mejorar la planificación, optimizar los recursos y aumentar la rentabilidad en el sector marítimo.

## 1.2. Objetivos del proyecto

El objetivo general de este proyecto es desarrollar una herramienta de apoyo, en forma de plataforma web, para el proceso de definición y evaluación de alternativas asociadas a un determinado gestor de buques, con el objetivo de obtener como resultado una gestión eficiente de atraques en puertos. Para ello, se crea un algoritmo que resuelve el Problema de gestión de atraques (BAP).

Los objetivos específicos a conseguir en el desarrollo de este software son los siguientes:

1. *Creación del anteproyecto.* Elaboración de un documento donde se expone una introducción, antecedentes y estado actual, así como un listado de tareas a realizar y el plan de trabajo establecido para la realización de este Trabajo Fin de Grado.
2. *Creación y configuración de la estructura principal del proyecto.* Creación de los directorios en los que se alojará todo el código, distribuido en los directorios frontend, back-end y algorithm.
3. *Creación de la API-REST.* Desarrollo de un back-end capaz de comunicar la interfaz de usuario con una base de datos como lo es MongoDB, haciendo uso del lenguaje de programación Java y su framework Spring.
4. *Creación de la interfaz de usuario.* Desarrollo de un front-end desde el que el usuario pueda interactuar con la aplicación. Para el desarrollo de esta interfaz se hizo uso del framework Vue.js y Vuetify.
5. *Implementación e integración en la aplicación del BAP.* Implementación de un módulo, aparte de los dos anteriores, capaz de resolver el problema de gestión de atraques (BAP).
6. *Creación de una pantalla de resultados.* Desarrollo de una pantalla en la que se muestre de manera gráfica y detallada la salida obtenida al aplicar el BAP.
7. *Redacción de la memoria.* Se describe el Trabajo Fin de Grado desarrollado.

# Capítulo 2

## Motivación y Estado del arte

### 2.1. Introducción

La gestión eficiente de atraques en puertos marítimos es un aspecto vital para garantizar operaciones fluidas, seguras y rentables en el sector naviero. En este capítulo se explora en profundidad la evolución histórica, los desafíos actuales, las soluciones tradicionales y los avances tecnológicos recientes en la gestión de atraques. Además, se analizan las tendencias futuras en este campo crucial.

### 2.2. Antecedentes históricos y problemas actuales

La gestión de atraques en los puertos marítimos ha sido un desafío persistente desde los primeros días de la navegación comercial. En sus inicios, las asignaciones de atraques se llevaban a cabo de manera manual y con recursos limitados, lo que resultaba en una coordinación deficiente y tiempos de espera prolongados. Por ejemplo, datos recopilados por la Organización Marítima Internacional (OMI) muestran que en la década de 1960, el tiempo promedio de espera para atracar en un puerto era de aproximadamente 2 días [4].

Con el crecimiento exponencial del comercio marítimo [4] y la expansión de la industria naviera, estos problemas se intensificaron. La congestión en los puertos se convirtió en una preocupación creciente, con buques esperando durante días para cargar y descargar su mercancía. Además, la falta de eficiencia en la asignación de recursos portuarios y la limitada coordinación entre los actores involucrados complicaban aún más la situación.

Estudios más recientes respaldan estos datos históricos, mostrando que incluso en la actualidad, la gestión de atraques sigue siendo un desafío significativo. Informes de la Conferencia de las Naciones Unidas sobre Comercio y Desarrollo (UNCTAD) revelan que la congestión en los puertos sigue siendo común, con costos económicos sustanciales debido a los retrasos en la cadena de suministro y los tiempos de espera de los buques [2].

Además, la gestión del espacio portuario, la programación de atraques, la asignación

eficiente de recursos y las consideraciones medioambientales y de seguridad son problemas actuales que requieren una atención inmediata. La necesidad de abordar estos desafíos se ha vuelto aún más urgente con el aumento constante del tamaño de los buques y el volumen de carga, lo que coloca una presión adicional sobre la infraestructura portuaria existente.

## 2.3. Soluciones tradicionales y avances recientes

La gestión de atraques ha evolucionado desde soluciones tradicionales hasta avances tecnológicos recientes, transformando la eficiencia y seguridad de las operaciones portuarias.

### 2.3.1. Sistemas de Control de Tráfico Marítimo (VTS)

Los sistemas de control de tráfico marítimo (VTS) surgieron como una solución inicial para monitorear y gestionar el tráfico de buques en áreas críticas como puertos y estrechos como se muestra en la figura 2.1.



Figura 2.1: Representación de un Sistema de Control de Tráfico Marítimo.

Estas son algunas de sus características principales:

- **Monitoreo de Tráfico.** Utilizan radares y sistemas de posicionamiento global (GPS) para rastrear la posición y el movimiento de los buques en tiempo real, proporcionando una visión completa del tráfico marítimo en la zona de operaciones.
- **Comunicación.** Facilitan la comunicación entre las autoridades portuarias y los buques, transmitiendo información vital sobre condiciones del tráfico, alertas de seguridad y otros datos relevantes para la navegación segura.

- **Asistencia en Navegación.** Proporcionan asesoramiento y orientación a los capitanes de los buques, incluyendo rutas seguras, maniobras de atraque y desatraque, y advertencias sobre posibles peligros o áreas restringidas.
- **Gestión del Tráfico.** Ayudan a gestionar el tráfico de buques mediante la coordinación de movimientos, la prevención de colisiones y la optimización de las operaciones portuarias, contribuyendo así a una mayor eficiencia y seguridad.

Un ejemplo destacado de un Sistema de Control de Tráfico Marítimo es el VTS del Puerto de Singapur. Singapur es uno de los puertos más importantes y concurridos del mundo, y su VTS desempeña un papel crucial en la gestión del tráfico marítimo en la región.

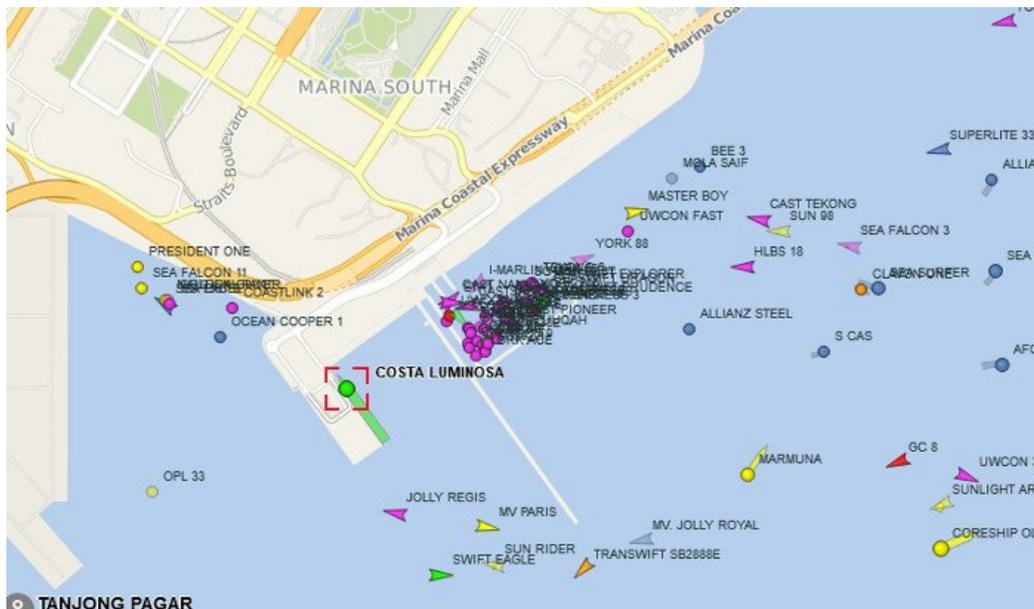


Figura 2.2: VTS del Puerto de Singapur.

El VTS de Singapur utiliza una combinación de tecnologías avanzadas, como radares de alta precisión, sistemas de posicionamiento global (GPS), sistemas de identificación automática (AIS) y cámaras de vigilancia, para monitorear y gestionar el tráfico de buques en el estrecho de Singapur y sus alrededores.

Algunas de las funciones clave del VTS de Singapur incluyen el monitoreo en tiempo real de la posición y el movimiento de los buques, la provisión de información sobre condiciones del tráfico y alertas de seguridad, la asistencia en la navegación y la gestión del tráfico para evitar colisiones y optimizar las operaciones portuarias.

Otro ejemplo destacado es el VTS del Puerto de Rotterdam, uno de los más avanzados y completos del mundo, que proporciona una amplia gama de funciones para monitorear y gestionar el tráfico marítimo en el puerto más grande de Europa.

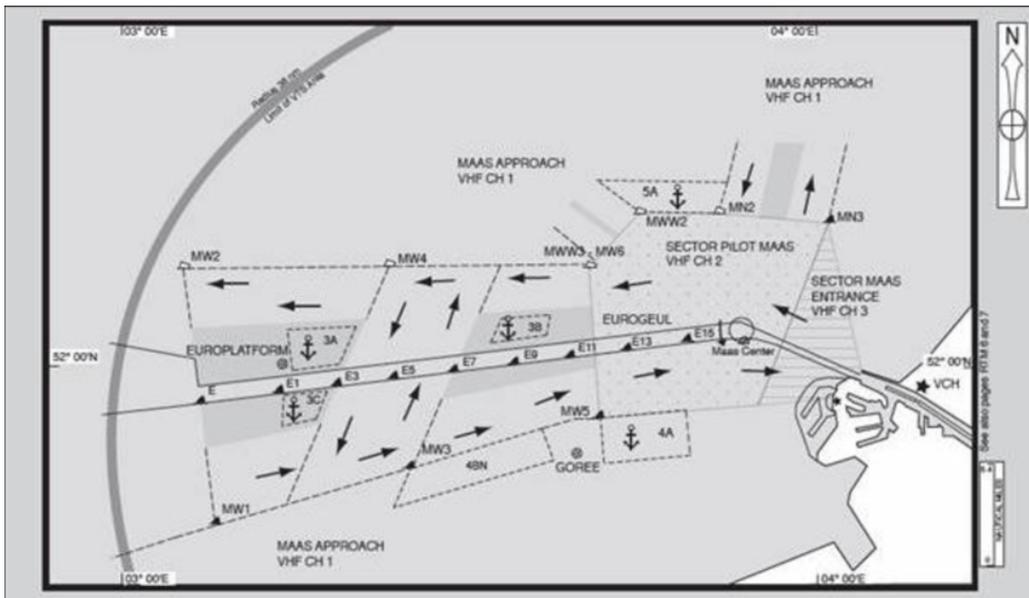


Figura 2.3: VTS del Puerto de Rotterdam.

El VTS del Puerto de Rotterdam utiliza una variedad de tecnologías, incluyendo radares, cámaras, sistemas de posicionamiento global (GPS) y software avanzado de análisis de datos para supervisar el tráfico marítimo, proporcionar asistencia en la navegación y garantizar la seguridad en el área portuaria. Es un excelente ejemplo de cómo los sistemas de control de tráfico marítimo pueden mejorar la eficiencia y la seguridad en los puertos y áreas marítimas críticas.

### 2.3.2. Sistemas de Información Portuaria (PIS)

Los sistemas de información portuaria (PIS) son herramientas clave para la gestión eficiente de operaciones dentro del puerto. Son esenciales para coordinar y optimizar el flujo de mercancías, embarcaciones y recursos, lo que contribuye a mejorar la productividad y la seguridad en las operaciones portuarias. En la figura 2.4 se muestra un ejemplo de PIS, en concreto de GisWeb.



Figura 2.4: Representación de GisWeb.

A continuación se presentan 4 ejemplos destacados de sistemas de información portuaria:

### Ejemplo 1: Sistema de Gestión de Terminal (TOS)



Figura 2.5: Sistema de Gestión de Terminal (TOS).

El Sistema de Gestión de Terminal (TOS) es un componente crucial en la gestión de operaciones en terminales de contenedores. Algunas de sus características incluyen:

- Planificación eficiente de carga y descarga de contenedores.
- Asignación óptima de recursos y equipos.
- Coordinación de actividades dentro del terminal.
- Seguimiento en tiempo real del estado de las operaciones.

## Ejemplo 2: Sistema de Gestión de Flotas (FMS)

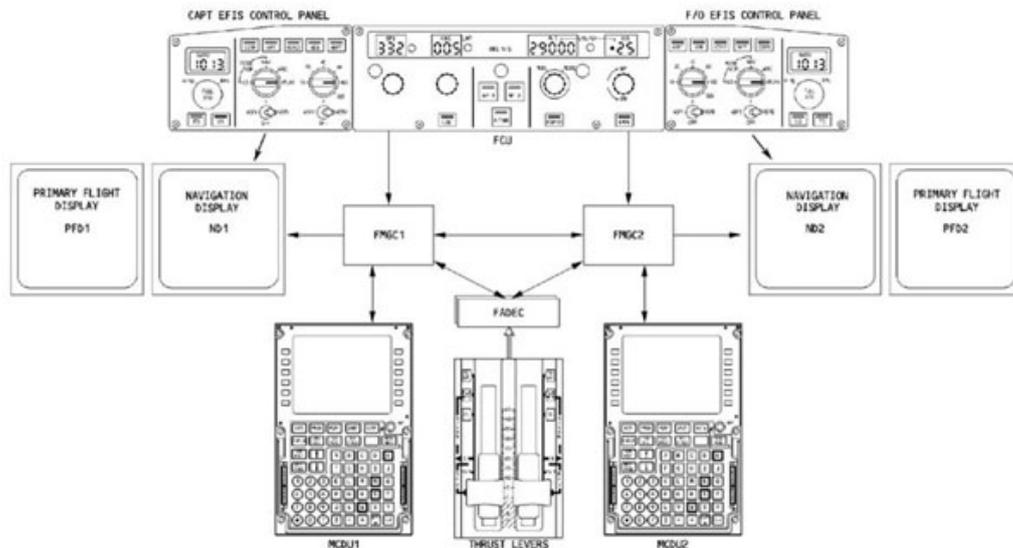


Figura 2.6: Ilustración del Sistema de Gestión de Flotas (FMS).

El Sistema de Gestión de Flotas (FMS) es esencial para la gestión eficiente de vehículos dentro del puerto, proporcionando características como la optimización de rutas y horarios, seguimiento en tiempo real de la ubicación y estado de los vehículos, gestión del mantenimiento y coordinación con otros sistemas de información portuaria para una logística integrada. Algunas de sus características incluyen:

- Optimización de rutas y horarios de los vehículos.
- Seguimiento en tiempo real de la ubicación y el estado de los vehículos.
- Gestión del mantenimiento para garantizar la disponibilidad operativa.
- Coordinación con otros sistemas de información portuaria para una logística integrada.

### Ejemplo 3: Sistema de Gestión de Almacenes (WMS)

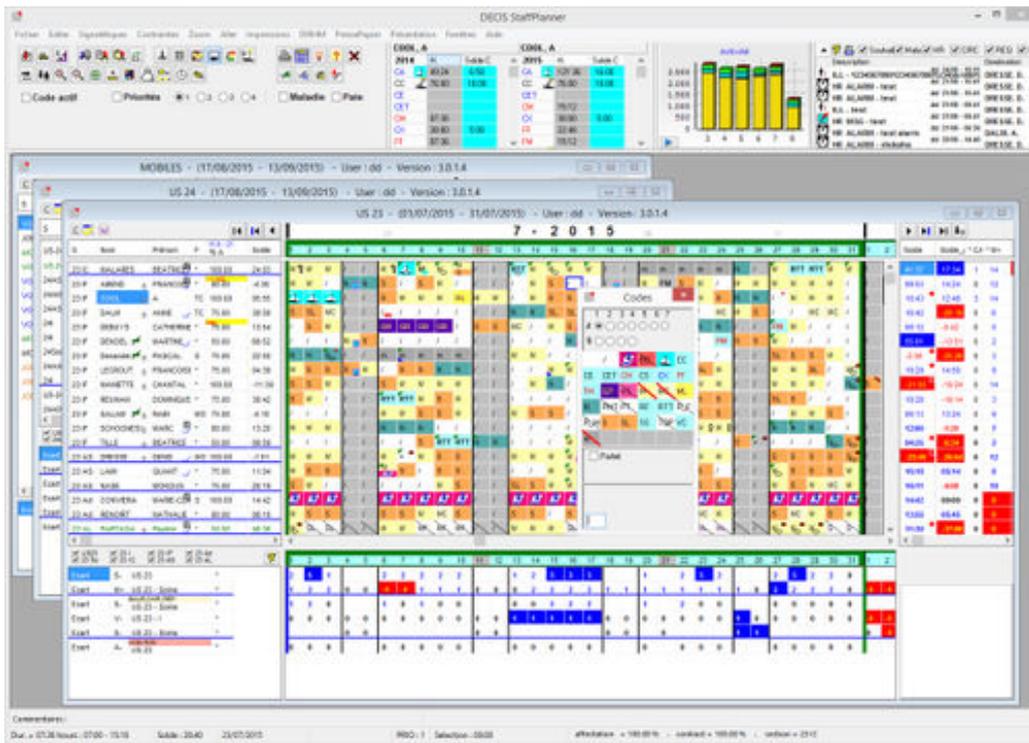


Figura 2.7: Sistema de Gestión de Almacenes (WMS).

El Sistema de Gestión de Almacenes (WMS) es fundamental para la gestión eficiente de operaciones de almacenamiento dentro del puerto. Algunas de sus características incluyen:

- Recepción y almacenamiento eficientes de mercancías.
- Gestión de pedidos y seguimiento del inventario en tiempo real.
- Optimización del espacio de almacenamiento.
- Coordinación con sistemas de gestión de flotas y terminales para una logística integrada.

## Ejemplo 4: Sistema de Gestión de Tráfico (TMS)

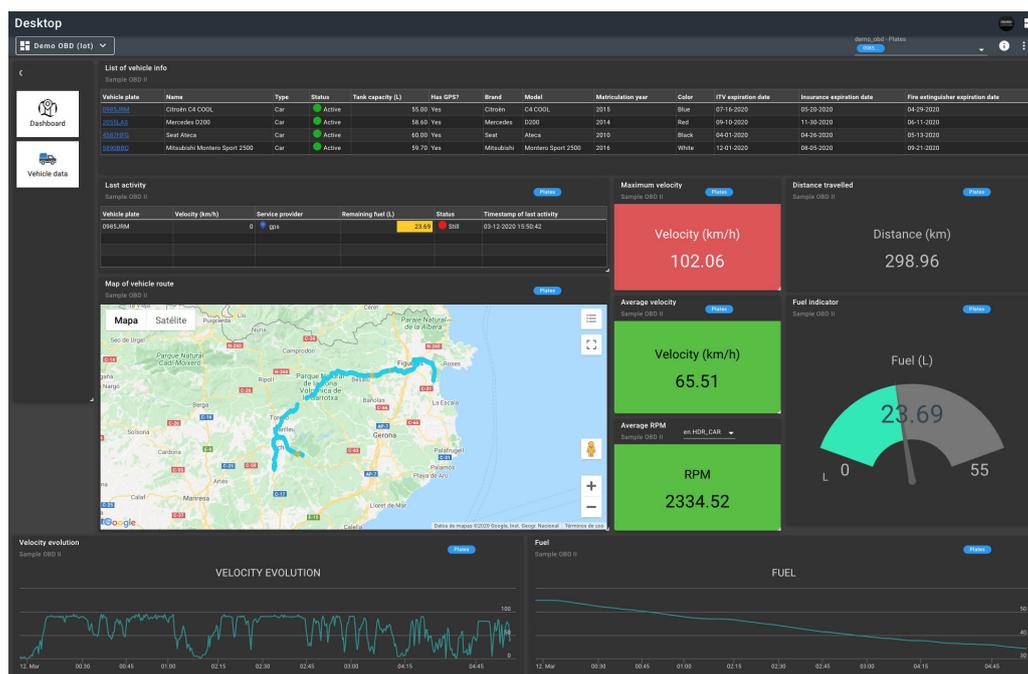


Figura 2.8: Sistema de Gestión de Tráfico (TMS).

El Sistema de Gestión de Tráfico (TMS) administra el tráfico de vehículos dentro del puerto para minimizar congestiones y optimizar los tiempos de espera. Algunas de sus características incluyen:

- Gestión de puertas de entrada y salida.
- Enrutamiento eficiente de vehículos dentro del puerto.
- Programación de horarios de entrega y recogida.
- Coordinación con sistemas de gestión de flotas y terminales para una logística integrada.

Estos ejemplos ilustran cómo los sistemas de información portuaria desempeñan un papel fundamental en la optimización de las operaciones portuarias, mejorando la eficiencia y la seguridad en los puertos marítimos modernos.

## 2.4. Tecnologías emergentes en gestión de atraques

La gestión de atraques se ha beneficiado enormemente de las tecnologías emergentes, que van desde sistemas de optimización basados en inteligencia artificial hasta plataformas de análisis de datos en tiempo real. A continuación, se presentan algunas de las principales soluciones utilizadas en la industria:

### 2.4.1. Aplicaciones de uso comercial

- **Awake.ai:** Esta plataforma utiliza inteligencia artificial y análisis de datos para optimizar la asignación de atraques, mejorar la eficiencia operativa y reducir los tiempos de espera de los buques. Awake.ai es reconocida por su capacidad de adaptación a las necesidades específicas de cada puerto y su enfoque centrado en la eficiencia y la seguridad.

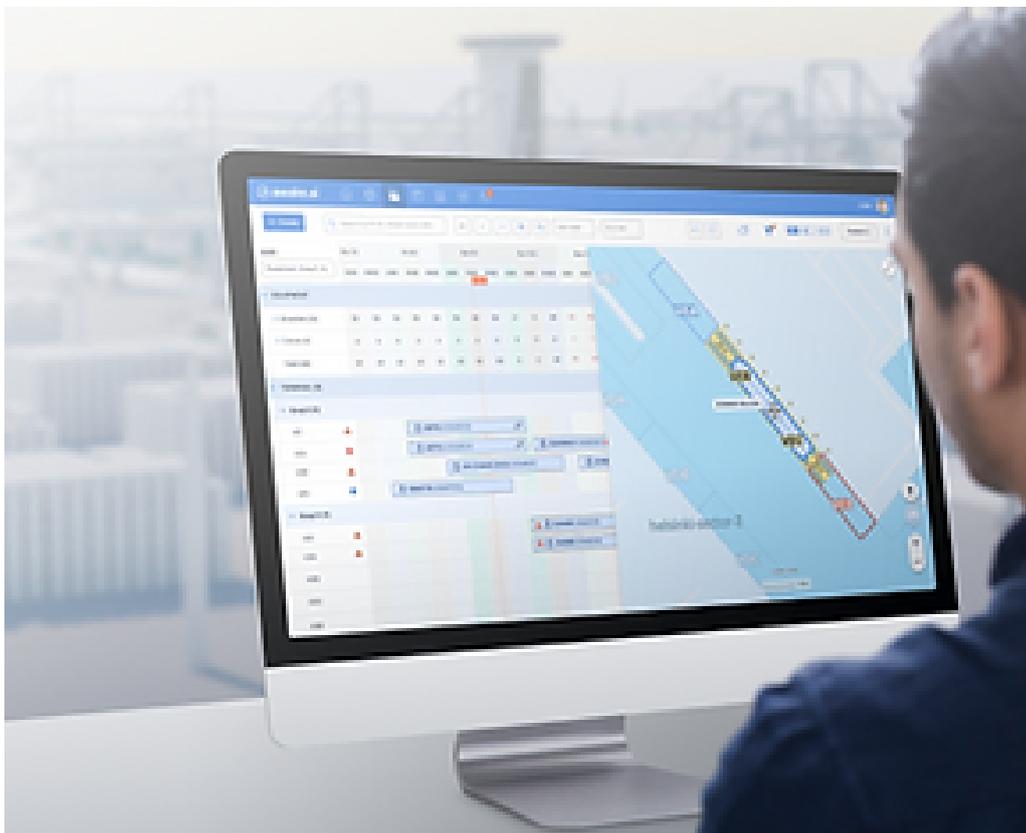


Figura 2.9: Interfaz de usuario de Awake.ai.

- **Portchain:** Portchain<sup>1</sup> es una plataforma de gestión de atraques que utiliza algoritmos de optimización avanzados para predecir la llegada y salida de buques, asignar recursos de manera eficiente y minimizar los tiempos de espera en los puertos. Su enfoque se centra en la mejora de la planificación y la coordinación entre todas las partes involucradas en las operaciones portuarias.
- **Berth Planner:** Berth Planner<sup>2</sup> es una plataforma basada en la nube que ofrece herramientas para la planificación y gestión de atraques en puertos y terminales marítimos. Permite a los operadores portuarios optimizar la asignación de espacios de atraque, coordinar las operaciones de carga y descarga, y mejorar la eficiencia general de las operaciones portuarias.

<sup>1</sup><https://www.portchain.com>

<sup>2</sup><https://www.ats.services/berth-planner>

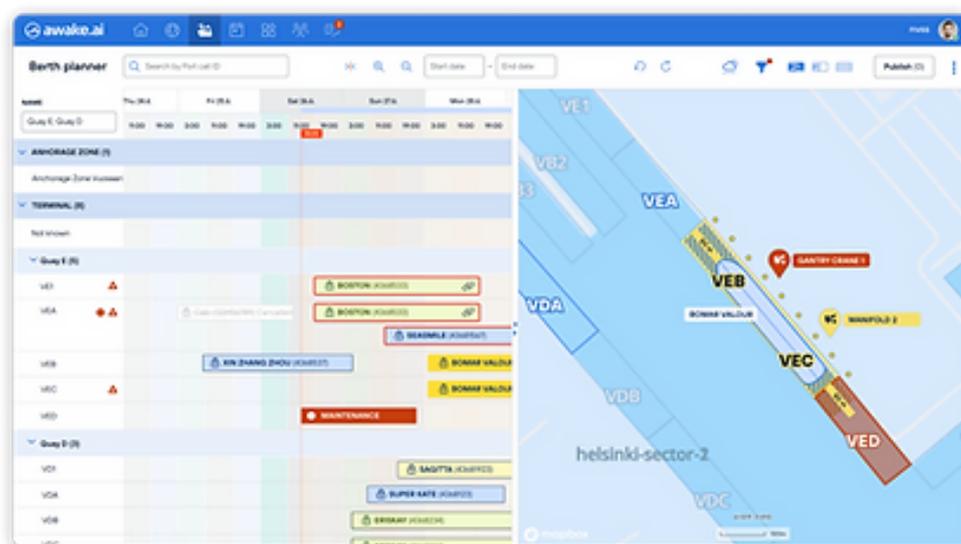


Figura 2.10: Interfaz de usuario de Berth Planner.

- PortXchange:** PortXchange<sup>3</sup> es una plataforma colaborativa que conecta a todas las partes involucradas en las operaciones portuarias, incluyendo autoridades portuarias, agentes marítimos y operadores de terminales. Facilita la coordinación en tiempo real, el intercambio de información y la planificación anticipada para reducir la congestión y mejorar la eficiencia en los puertos.

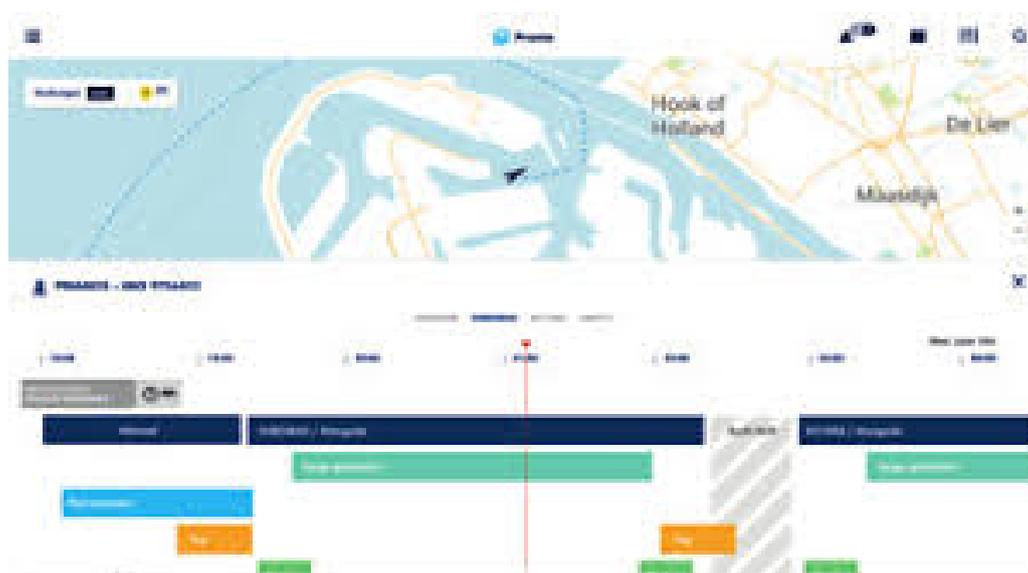


Figura 2.11: Interfaz de usuario de PortXchange en el puerto de Rotterdam.

## 2.4.2. Sistemas de Control de Tráfico Marítimo Avanzados (VTS)

Los sistemas de control de tráfico marítimo han evolucionado para incorporar tecnologías avanzadas como inteligencia artificial y análisis predictivo. Estos sistemas proporcio-

<sup>3</sup><https://port-xchange.com>

nan una mayor capacidad de monitoreo y gestión del tráfico marítimo, lo que contribuye a una navegación más segura y eficiente en los puertos.



Figura 2.12: VTS modernos.

Los VTS modernos pueden integrarse con plataformas de gestión de atraques para compartir datos en tiempo real y optimizar la asignación de atraques en función de las condiciones del tráfico y la disponibilidad de muelles. La figura 2.13 muestra un esquema de integración de VTS con plataformas de gestión de atraques.

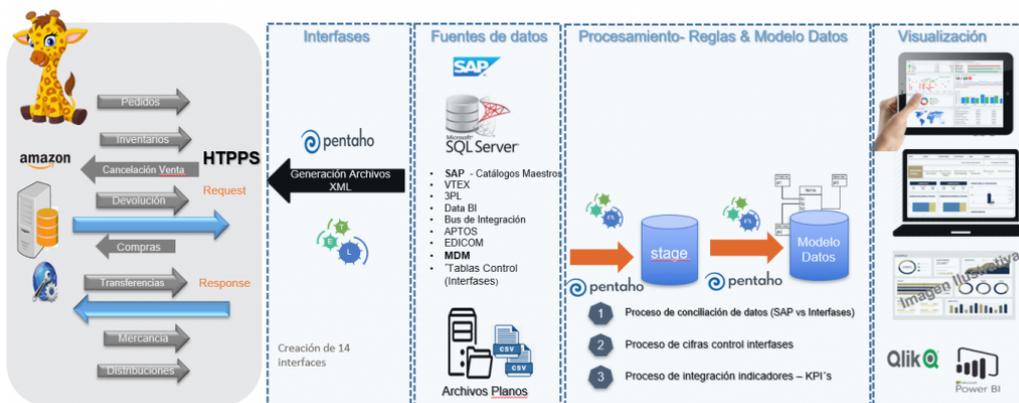


Figura 2.13: Integración con plataformas de gestión de atraques.

Los VTS pueden aprovechar el análisis de big data para procesar grandes volúmenes de datos de tráfico marítimo y proporcionar insights valiosos para la toma de decisiones en la gestión de atraques y la planificación de operaciones portuarias. Un ejemplo del uso de big data en VTS se muestra en la figura 2.14.

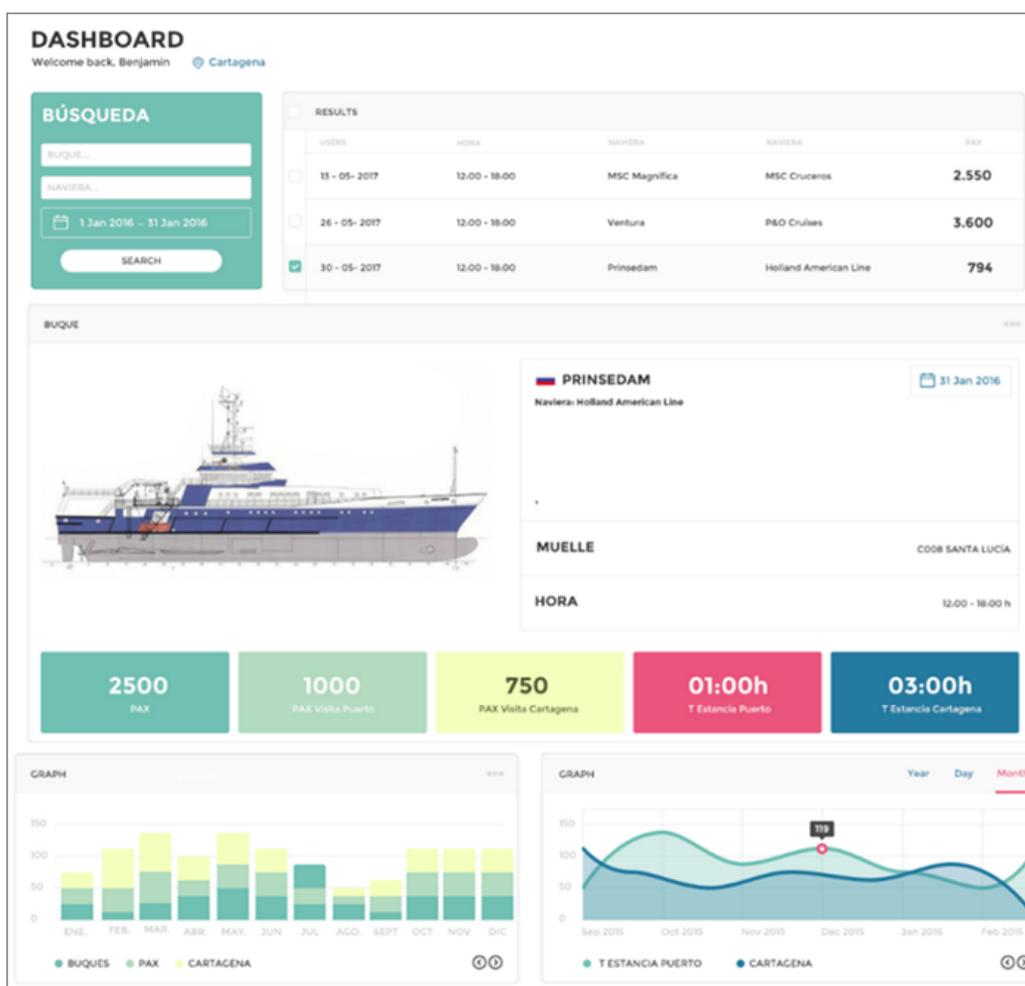


Figura 2.14: Análisis de big data en tiempo real para control y monitorización de turistas.

Por último, algunos VTS modernos incluyen capacidades de automatización para agilizar los procesos de gestión de tráfico marítimo, como la detección automática de colisiones potenciales y la generación de alertas para acciones preventivas.

### 2.4.3. Soluciones basadas en IoT y Blockchain

Las aplicaciones de IoT y Blockchain están ganando popularidad en la gestión de atraques marítimos debido a su capacidad para mejorar la trazabilidad y la seguridad. Un ejemplo destacado de esta combinación es la plataforma "Maritime Blockchain Labs", cuyo esquema de funcionamiento se muestra en la figura 2.15. Esta plataforma utiliza tecnología Blockchain para garantizar la integridad y la seguridad de los datos relacionados con la gestión de atraques, mientras que la tecnología IoT se utiliza para recopilar datos en tiempo real de los buques y otros activos marítimos.

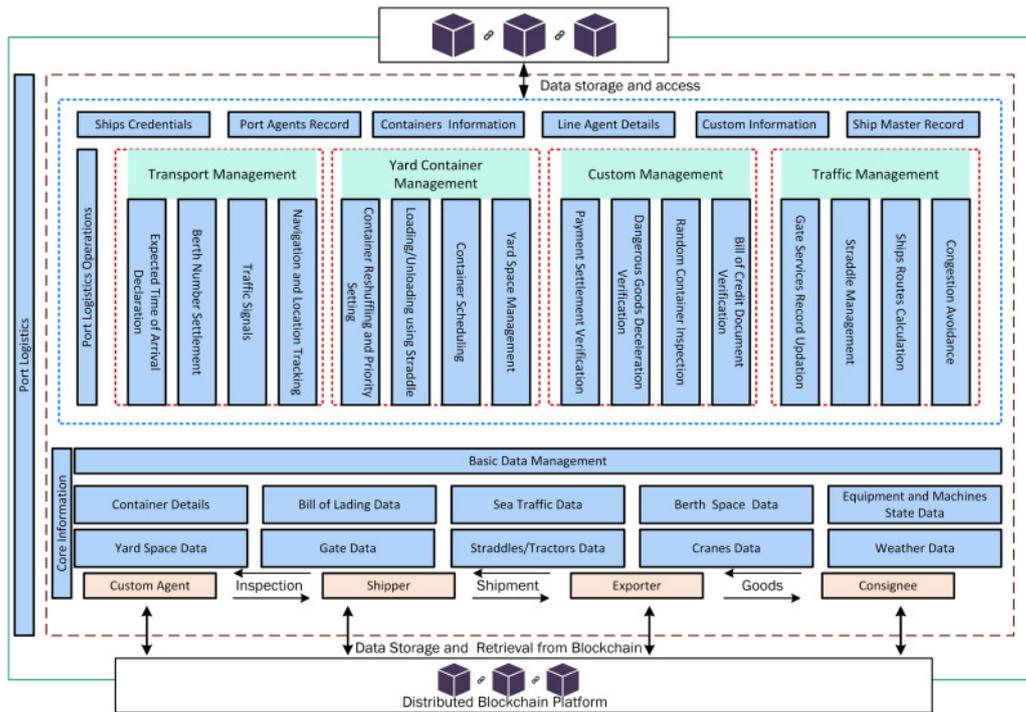


Figura 2.15: Plataforma Maritime Blockchain Labs.

La plataforma Maritime Blockchain Labs proporciona una visión integral de las operaciones portuarias y marítimas, permitiendo una mayor transparencia y eficiencia en la gestión de atraques.

#### 2.4.4. Sistemas de gestión de flotas marítimas

Las plataformas de gestión de flotas marítimas son herramientas esenciales para optimizar la operación de buques y mejorar la eficiencia de las operaciones portuarias. Un ejemplo destacado en esta categoría es el sistema "Fleet Complete".

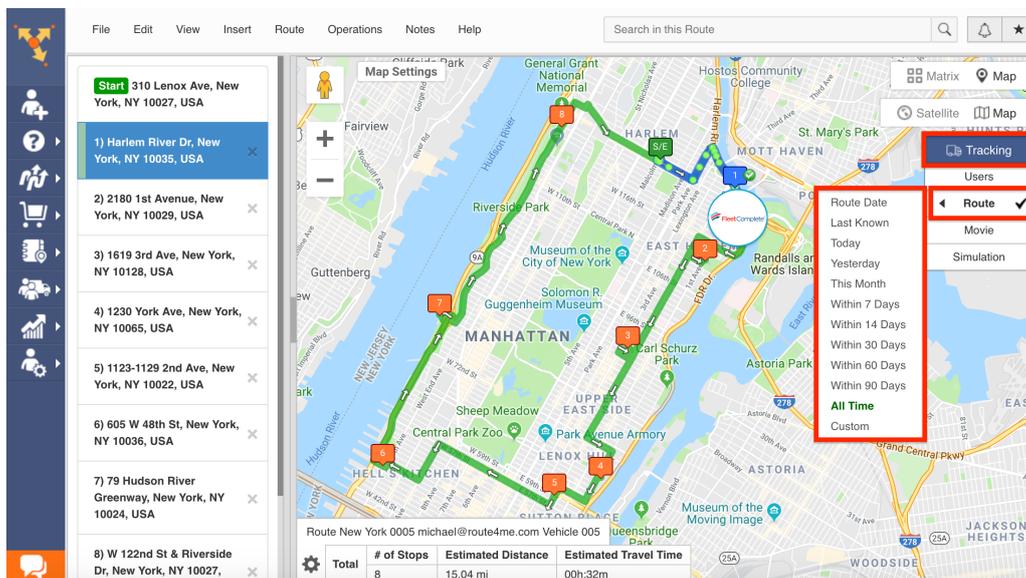


Figura 2.16: Plataforma Fleet Complete.

Fleet Complete ofrece una amplia gama de funcionalidades, incluyendo la programación de rutas optimizadas, el monitoreo del rendimiento del buque en tiempo real y la gestión eficiente del combustible. Esta plataforma ayuda a las compañías navieras a reducir costos, mejorar la seguridad y maximizar la eficiencia de sus flotas marítimas.

## 2.4.5. Awake.ai

Awake.ai<sup>4</sup> es una start-up finlandesa, fundada en 2018 en Turku, con el objetivo de crear una plataforma de software para la digitalización portuaria y el transporte marítimo y autónomo, y cuyas soluciones se centran en el desarrollo de modelos AI/ML personalizados para optimizar el flujo de carga a través de los puertos, reduciendo los tiempos de espera y las emisiones.

Su principal característica es el uso de inteligencia artificial y analítica de datos para optimizar la gestión de atraques. También ofrece funcionalidades como optimización en tiempo real, análisis predictivo y colaboración entre actores portuarios. Sin embargo, su utilización requiere un alto costo de implementación y mantenimiento.

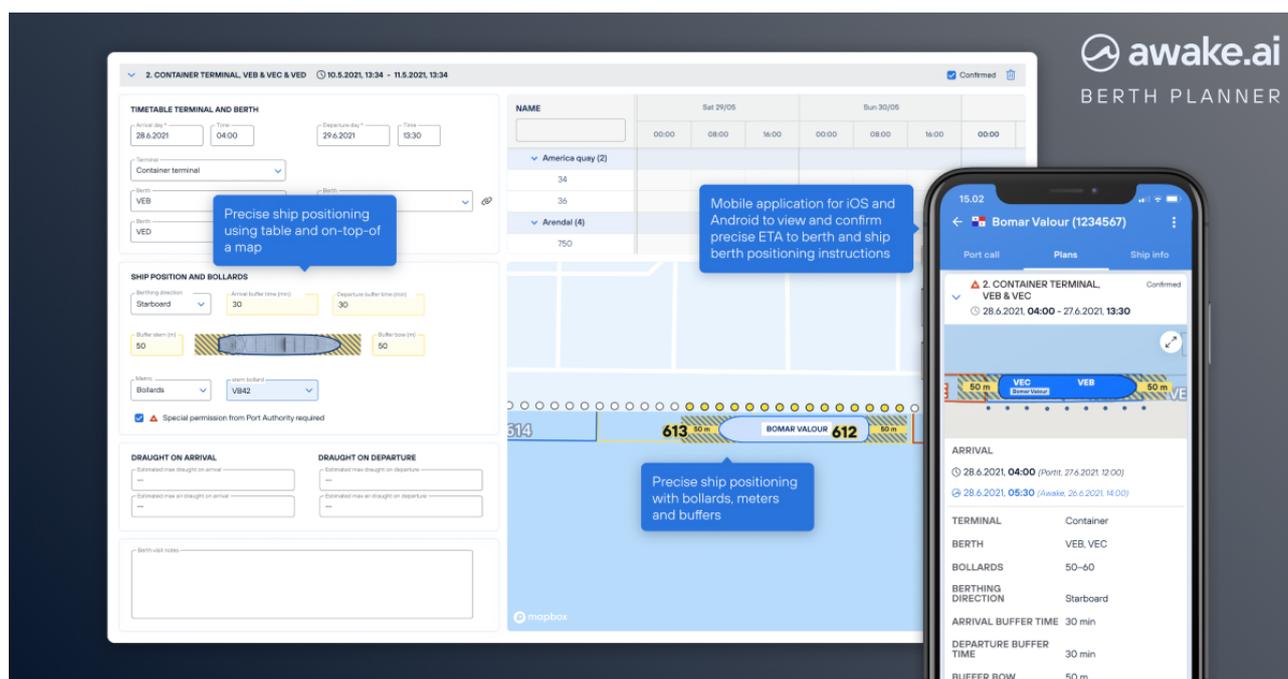


Figura 2.17: Interfaz de usuario de Awake.ai.

La figura 2.17 muestra la interfaz de usuario de Awake.ai, que proporciona a los usuarios acceso a las funcionalidades de la plataforma, como la optimización en tiempo real y el análisis predictivo.

<sup>4</sup><https://www.awake.ai>

## 2.5. Perspectivas futuras y tendencias

El futuro de la gestión de atraques estará marcado por una serie de avances tecnológicos y cambios en las prácticas operativas que transformarán la eficiencia y la sostenibilidad en los puertos marítimos. Además de la continua influencia de la inteligencia artificial (IA), el Internet de las Cosas (IoT) y la tecnología blockchain, se vislumbran otras tendencias y desarrollos clave como los siguientes.

- **Optimización predictiva y análisis avanzado de datos.** Se espera que la optimización predictiva y el análisis avanzado de datos desempeñen un papel fundamental en la gestión de atraques. Los sistemas de IA y análisis avanzado de datos permitirán una planificación más precisa al prever la demanda de atraques, identificar patrones de tráfico y optimizar la asignación de recursos en tiempo real. Esto no solo mejorará la eficiencia operativa, sino que también reducirá los tiempos de espera y la congestión en los puertos.
- **Integración de energías renovables y electrificación de flotas.** La integración de energías renovables y la electrificación de flotas serán aspectos cruciales para lograr prácticas portuarias más sostenibles. Se espera que los puertos inviertan en infraestructuras de carga eléctrica para buques y vehículos terrestres, reduciendo así las emisiones de gases de efecto invernadero y mejorando la calidad del aire en las zonas portuarias. La adopción de energía solar, eólica e incluso mareomotriz también contribuirá a la reducción de la huella de carbono de los puertos.
- **Automatización y robótica en operaciones portuarias.** La automatización y la robótica jugarán un papel cada vez más importante en las operaciones portuarias. Se espera que los puertos implementen sistemas autónomos para tareas como carga y descarga de contenedores, remolque de buques y mantenimiento de infraestructuras portuarias. Esto no solo mejorará la eficiencia y la seguridad, sino que también reducirá la dependencia de la mano de obra humana y mitigará los riesgos laborales asociados.
- **Colaboración y estandarización en la industria.** La colaboración entre los actores del puerto y la estandarización de procesos serán fundamentales para una gestión de atraques más eficiente y transparente. Se espera que los puertos trabajen en conjunto con navieras, autoridades portuarias, proveedores de servicios y otros socios para desarrollar estándares comunes, compartir datos en tiempo real y optimizar la cadena de suministro marítima. Esto facilitará la interoperabilidad entre diferentes sistemas y promoverá una mayor eficiencia en todo el ecosistema portuario.
- **Desarrollo de infraestructuras portuarias inteligentes.** El desarrollo de infraestructuras portuarias inteligentes será clave para enfrentar los desafíos futuros de la gestión de atraques. Se espera que los puertos inviertan en sistemas de sensores avanzados, redes de comunicación de alta velocidad y plataformas de gestión centralizada para mejorar la monitorización y el control de las operaciones portuarias. Esto permitirá una toma de decisiones más informada, una respuesta más rápida a los cambios en las condiciones operativas y una mayor adaptabilidad a las demandas del mercado.

# Capítulo 3

## Tecnologías y arquitectura de software

En este capítulo explora en detalle las tecnologías utilizadas en el desarrollo de la aplicación web, así como su arquitectura a nivel de software y su composición. Estos aspectos son fundamentales para comprender la estructura y el funcionamiento de la aplicación.

### 3.1. Arquitectura del sistema

La arquitectura de software de la aplicación sigue un enfoque basado en microservicios y se adhiere al patrón de arquitectura hexagonal. Cada componente se desarrolla de forma independiente y se despliega como un contenedor Docker, lo que permite una mayor modularidad, escalabilidad y mantenimiento del sistema.

El patrón de arquitectura hexagonal, también conocido como arquitectura de puertos y adaptadores, se caracteriza por separar claramente las preocupaciones y promover la independencia entre las diferentes capas del sistema. En esta arquitectura, el núcleo de la aplicación se encuentra en el centro y está rodeado por capas de adaptadores que se encargan de interactuar con el exterior. Esto facilita la integración con tecnologías externas y permite realizar cambios en la implementación interna sin afectar a la interfaz externa. El esquema básico de la arquitectura hexagonal se muestra en la figura 3.1.

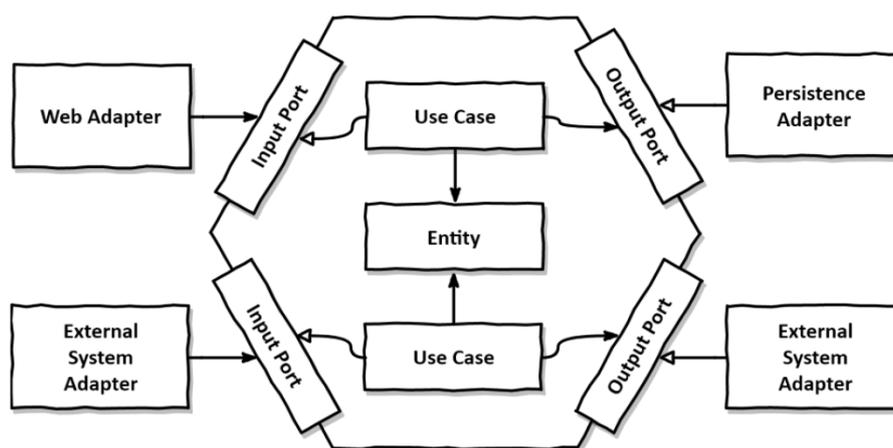


Figura 3.1: Representación de la arquitectura hexagonal.

Cada componente del sistema sigue esta estructura hexagonal, lo que proporciona una organización clara y coherente del código. Además, el uso de contenedores Docker para el despliegue garantiza la consistencia del entorno de ejecución y simplifica la gestión de dependencias.

En resumen, la arquitectura del sistema se basa en principios de modularidad, independencia y flexibilidad, lo que permite adaptarse fácilmente a los cambios y mantener un alto nivel de calidad en el desarrollo y despliegue de la aplicación. En la figura 3.2 se muestra la arquitectura general de la aplicación desarrollada.

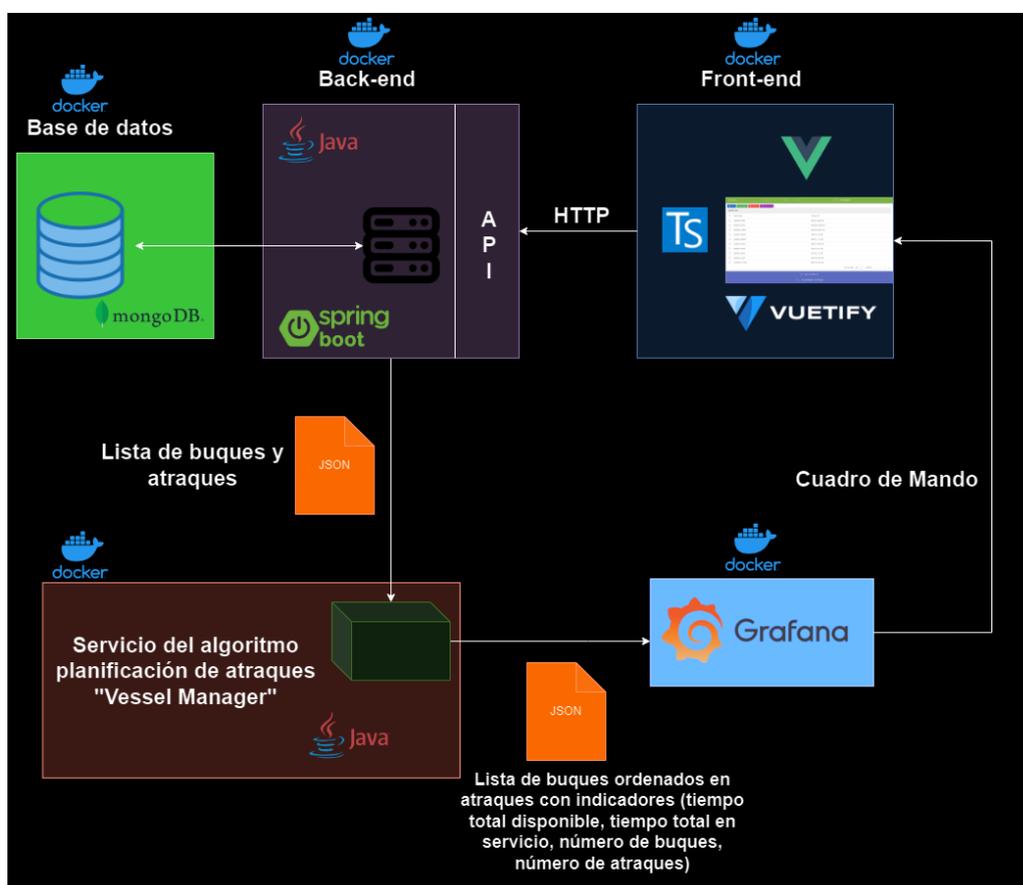


Figura 3.2: Representación de la arquitectura de la aplicación.

### 3.1.1. Componentes principales

La aplicación se compone de varios microservicios que se comunican entre sí mediante APIs RESTful. Los componentes principales de la aplicación son los siguientes:

1. **Back-end.** Implementa las reglas de negocio y proporciona servicios web para la gestión de atraques y otras operaciones relacionadas mediante una API REST. Este módulo actúa como el núcleo del sistema.
2. **Front-end (Interfaz de usuario).** Presenta una interfaz gráfica para que los usuarios interactúen con el sistema. Este módulo se comunica con el back-end a través de los puertos definidos.
3. **Algoritmo de gestión de atraques.** Algoritmo que realiza la asignación de buques a los atraques de manera eficiente. Se invoca desde el back-end. Este componente se conecta al back-end a través de puertos definidos.
4. **Base de datos.** Almacena toda la información relevante sobre los buques, atraques, operaciones, contenedores, almacenamientos y planificaciones. Es crucial para la persistencia y recuperación eficiente de los datos.
5. **Dashboard principal.** Interfaz centralizada que muestra indicadores como la comparación de los tiempos de duración de los buques, el número de buques y el número de atraques. Permite a los usuarios monitorear y gestionar las operaciones portuarias de manera efectiva.

La comunicación entre los componentes se realiza utilizando el patrón de "puertos y adaptadores", donde los puertos representan las interfaces a través de las cuales los componentes se comunican. Los adaptadores proporcionan implementaciones específicas para conectar los puertos con tecnologías externas.

### 3.1.2. Despliegue y orquestación

El despliegue de la aplicación se realiza mediante Docker y Docker Compose, que permiten definir y gestionar los contenedores de manera eficiente. Los contenedores se ejecutan en entornos locales durante el desarrollo y pruebas, y se pueden desplegar en la nube o en servidores locales para su producción. Docker Hub se utiliza para almacenar y compartir imágenes de contenedores Docker, lo que facilita el despliegue de la aplicación en diferentes entornos.

## 3.2. Tecnologías utilizadas

Para el desarrollo de este proyecto se han empleado varias tecnologías clave que proporcionan las bases para su implementación y funcionamiento.

### **3.2.1. MongoDB**

MongoDB es un sistema de base de datos NoSQL orientado a documentos. Se caracteriza por su almacenamiento de datos en documentos con una estructura BSON, lo que proporciona rapidez y facilidad de integración. La elección de MongoDB se basa en su escalabilidad, flexibilidad y extensa documentación.

### **3.2.2. Spring y Spring Boot**

Spring es un framework de aplicación Java que proporciona una amplia gama de características para el desarrollo empresarial. Spring Boot, por otro lado, es una herramienta que simplifica el proceso de configuración inicial y la preparación de las aplicaciones para la producción. Spring Boot permite compilar aplicaciones web como un archivo .jar que se puede ejecutar como una aplicación Java normal. Entre las características de Spring se incluyen la inversión de control (IoC), la inyección de dependencias y el soporte para transacciones declarativas, lo que permite el desarrollo de aplicaciones robustas y escalables.

### **3.2.3. Vue.js y Vuetify**

Vue.js es un framework progresivo de JavaScript para construir interfaces de usuario. Destaca por su curva de aprendizaje suave y su flexibilidad. Vue.js permite el desarrollo de aplicaciones front-end complejas de forma modular y eficiente. Vuetify, por su parte, es una librería de Vue.js que implementa componentes basados en la filosofía de Material Design, lo que facilita la creación de interfaces de usuario atractivas y funcionales. Entre las características de Vue.js se incluyen la reactividad, las directivas y la capacidad de crear componentes reutilizables.

### **3.2.4. Docker y Docker Hub**

Docker es una plataforma de código abierto que permite la creación, el despliegue y la ejecución de aplicaciones en contenedores. Los contenedores de Docker son unidades de software ligeros y portátiles que incluyen todo lo necesario para ejecutar una aplicación, incluidas las bibliotecas, las herramientas y el código. Docker Hub es un servicio en la nube que permite a los usuarios almacenar y compartir imágenes de contenedores Docker. Esto facilita la distribución y el despliegue de aplicaciones en entornos de producción.

### **3.2.5. Git y GitHub**

Git es un sistema de control de versiones distribuido que se utiliza para gestionar el código fuente del proyecto. GitHub es una plataforma de alojamiento en la nube que permite a los desarrolladores colaborar en proyectos de software utilizando Git. GitHub

proporciona herramientas para el seguimiento de problemas, la revisión de código y la integración continua, lo que facilita el desarrollo colaborativo y el mantenimiento del código.

### 3.3. Estructura de la aplicación

En el desarrollo de este proyecto, se ha creado una aplicación web Full Stack desde cero, abarcando tanto la parte visual de la interfaz de usuario (front-end) como el desarrollo del servidor (back-end) capaz de procesar las solicitudes del usuario, junto con un módulo dedicado al algoritmo para organizar los buques en los ataques. A continuación, se describen en detalle los componentes principales de la aplicación.

#### 3.3.1. Servidor (Back-end)

En la figura 3.3 se muestra la estructura principal del back-end.



Figura 3.3: Estructura del back-end.

El back-end no solo proporciona un conjunto de endpoints para operaciones CRUD, sino que también implementa la lógica de negocio de la aplicación, incluida la ejecución del algoritmo para organizar los buques en los ataques de manera eficiente. Además de

los endpoints para manipular entidades como contenedores, almacenamientos, atraques, operaciones y buques, se han agregado los siguientes:

- **/schedules:** Este endpoint recibe un intervalo de tiempo y guarda una lista de buques disponibles en ese intervalo de tiempo y una lista de atraques disponibles en ese intervalo de tiempo, en un archivo JSON para enviarlo al algoritmo.
- **/solutions:** Este endpoint representa una solución generada por el algoritmo, con atributos como un ID para identificarlo, un nombre de la planificación, el nombre del archivo en el servidor, la URL del archivo y su contenido. Cuando se solicita información detallada de una planificación, se guarda la URL del archivo para que Grafana pueda leerlo y mostrar el cuadro de mando.

La operativa del back-end se ofrece al resto de componentes a través de un API REST. Todos los endpoints de esta API cuentan con operaciones de GET, POST, PUT y DELETE para obtener, crear, modificar y eliminar entidades respectivamente. Además, proporcionan un método GET específico para obtener una entidad por su ID.

A continuación se muestra el listado de endpoints disponibles en la API REST creada.

- **/containers:** Representa un contenedor con atributos como un ID para identificar el recurso y un nombre.
- **/storages:** Representa un almacenamiento con atributos como un nombre para identificar el recurso y una cantidad.
- **/berths:** Representa un atraque con atributos como una fecha de apertura y una de cierre.
- **/operations:** Representa una operación con atributos como un ID para identificar el recurso, un tipo (carga o descarga), una fuente, un destino, un tiempo de operación y un conjunto de contenedores.
- **/vessels:** Representa un buque con atributos como un ID para identificar el recurso, un nombre, un tiempo de llegada, un tiempo de salida y un conjunto de operaciones.

### 3.3.2. Interfaz de usuario (Front-end)

En este módulo se desarrolla la interfaz de usuario utilizando el framework Vue.js y su librería Vuetify. En la figura 3.4 se muestra la estructura principal del front-end.



Figura 3.4: Estructura del front-end.

Entre las principales funcionalidades que permite el front-end a los usuarios encontramos las siguientes.

- **Gestión de entidades.** Los usuarios pueden realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en las entidades principales del sistema, como buques y atraques. Estas operaciones les permiten definir y mantener la información relacionada con los buques y atraques de manera eficiente.
- **Definición de planificaciones.** Los usuarios pueden definir planificaciones especificando un intervalo de tiempo para generar soluciones con el algoritmo de planificación de buques. Esta funcionalidad les permite establecer criterios temporales para la generación de soluciones.
- **Generación de cuadros de mando.** Los usuarios pueden generar cuadros de mando a partir de las soluciones generadas por el algoritmo. Estos cuadros de mando proporcionan una visualización clara y detallada de la distribución de buques en los atraques, lo que facilita la toma de decisiones informadas.
- **Descarga de soluciones.** Los usuarios pueden descargar las soluciones generadas por el sistema en formato JSON. Esta funcionalidad les permite almacenar y compartir las soluciones obtenidas para su posterior análisis o referencia.

Para implementar las anteriores funcionalidades se crean una serie de casos de uso accesibles desde el front-end. Estos casos de uso se muestran en el diagrama de la figura 3.5.

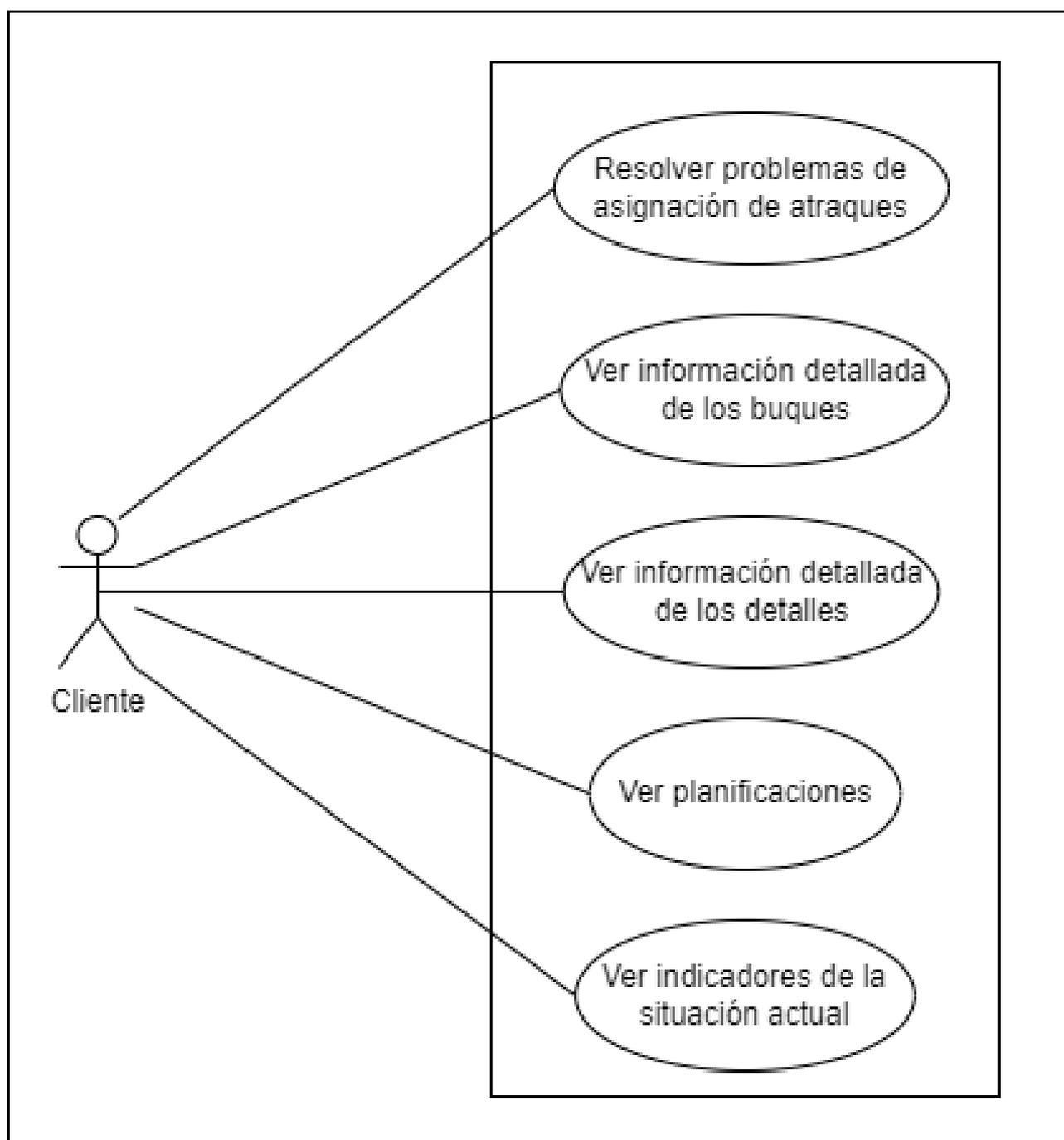


Figura 3.5: Diseño casos de uso.

Cada uno de estos casos de uso se implementa mediante una serie de vistas y componentes de Vue.js que permiten al usuario interactuar con la aplicación de manera intuitiva. En la sección 4.2 se explica en detalle cada uno de estos casos de uso.

### 3.3.3. Algoritmo de organización de buques

El algoritmo de organización de buques es una técnica constructiva que utiliza un comportamiento semialeatorio para asignar los buques a los ataques disponibles de manera eficiente y equitativa. El algoritmo 1 muestra un pseudocódigo del algoritmo

propuesto. Dado que es un algoritmo constructivo, partimos de una solución vacía (línea 1) y se hacen tantas iteraciones como elementos hay en la lista de buques (línea 2). El proceso de asignación empieza seleccionando un buque aleatorio de la lista (línea 3) así como un atraque aleatorio siempre que se le pueda asignar un buque (línea 4). Una vez encontrados buque y atraque, se realiza la asignación (línea 5) y se elimina el buque de la lista para que no pueda volver a ser seleccionado (línea 6). Al terminar el bucle, tendremos una solución factible la cual se devuelve (línea 8).

---

**Algoritmo 1:** Pseudocódigo del algoritmo de organización de buques

---

```

1  Entrada:  $L_v$ . Lista de buques
2  Entrada:  $L_b$ . Lista de atraques
   1:  $solution \leftarrow$  solución vacía
   2: while  $L_v$  no sea vacía do
   3:    $vessel \leftarrow getRandomVessel(L_v)$ 
   4:    $berth \leftarrow gerRandomBerth(L_b, vessel)$ 
   5:    $solution \leftarrow assignToBerth(solution, vessel, berth)$ 
   6:    $L_v.remove(vessel)$ 
   7: end while
8: Return  $solution$ 

```

---

El algoritmo de organización de buques utiliza una técnica semialeatoria para asignar eficientemente los buques a los atraques disponibles, asegurando una distribución equitativa y maximizando el uso de los recursos portuarios. Este método garantiza que el proceso de planificación sea tanto efectivo como justo, facilitando la operación óptima del puerto.

### Beneficios del Algoritmo

- **Eficiencia:** El algoritmo asigna cada buque a un atraque de manera rápida mediante la selección aleatoria, asegurando que todas las entradas sean procesadas.
- **Equidad y ausencia de sesgo:** La selección aleatoria de buques y atraques asegura que no haya un patrón sesgado en la asignación, promoviendo una distribución equitativa de los buques.
- **Utilización óptima de recursos:** Al maximizar la utilización de los atraques disponibles, se mejora la eficiencia operativa del puerto.

# Capítulo 4

## Caso práctico

En este capítulo se explora el proceso de despliegue del sistema y su funcionamiento, utilizando pantallazos de la aplicación para ilustrar cada paso.

### 4.1. Despliegue del sistema

Para el despliegue de la aplicación se utilizan contenedores Docker, de tal forma que el mismo sea más escalable y portable. Los 2 elementos fundamentales para el despliegue del sistema son los siguientes:

1. **GitHub Actions.** Crea las imágenes Docker de los componentes del sistema y las publica en Docker Hub cada vez que se realiza un commit en los repositorios correspondientes.
2. **Docker Compose.** Construye la aplicación a partir de los contenedores Docker publicados en Docker Hub.

Los servicios desplegados al levantar la aplicación son los siguientes.

- **Front-end.** La interfaz de usuario accesible en `http://localhost:80`.
- **Back-end.** El servidor que maneja la lógica de negocio accesible en `http://localhost:8080`.
- **Algoritmo.** Algoritmo de planificación de ataques. Accesible en `http://localhost:8081/algorithms/vessel-manager_algorithm`.
- **MongoDB.** La base de datos es accesible desde `mongodb://localhost:27017`.
- **Grafana.** Accesible desde `http://localhost:3000`.

Al ejecutar el docker compose, el sistema descarga las imágenes correspondientes de Docker Hub y levanta todos los contenedores necesarios, tal y como se muestra en la figura 4.1.

```
sudo docker-compose --profile full up
Creating network "fullstack-network" with the default driver
Creating kaizen-task-lite ... done
Creating mongo ... done
Creating grafana ... done
Creating back-end ... done
Creating front-end ... done
Attaching to mongo, grafana, kaizen-task-lite, back-end, front-end
```

Figura 4.1: Aplicación entera desplegada con Docker Compose.

## 4.2. Funcionamiento del sistema

Esta sección describe en detalle como se desarrolla cada uno de los casos de uso explicados en la sección 3.3.2 a través de la aplicación.

### 4.2.1. Inicio

La página de inicio o dashboard (figura 4.2) muestra los cuadros de mando que proporcionan información en tiempo real sobre el número de atraques, número de buques, entre otros.



Figura 4.2: Página de inicio.

Los principales indicadores del cuadro de mando son los siguientes.

- **Número de atraques y buques.** Proporciona una visión del número total de atraques y buques que han utilizado los servicios del puerto en un período de tiempo determinado. Estos datos ayudan a comprender la carga de trabajo del puerto y la utilización de sus recursos.
- **Programación de atraques.** Muestra la asignación de barcos a los muelles en momentos específicos. Permite visualizar la ocupación de los muelles en diferentes momentos del día, lo que facilita la planificación y la optimización de los recursos.
- **Duración de buques.** Aparece un diagrama de barras donde muestra la comparación de tiempos por cada buque en horas.

Además, la aplicación proporciona un menú de acceso rápido a las diferentes secciones de la aplicación. En la figura 4.3 se muestra un ejemplo del menú desplegado.

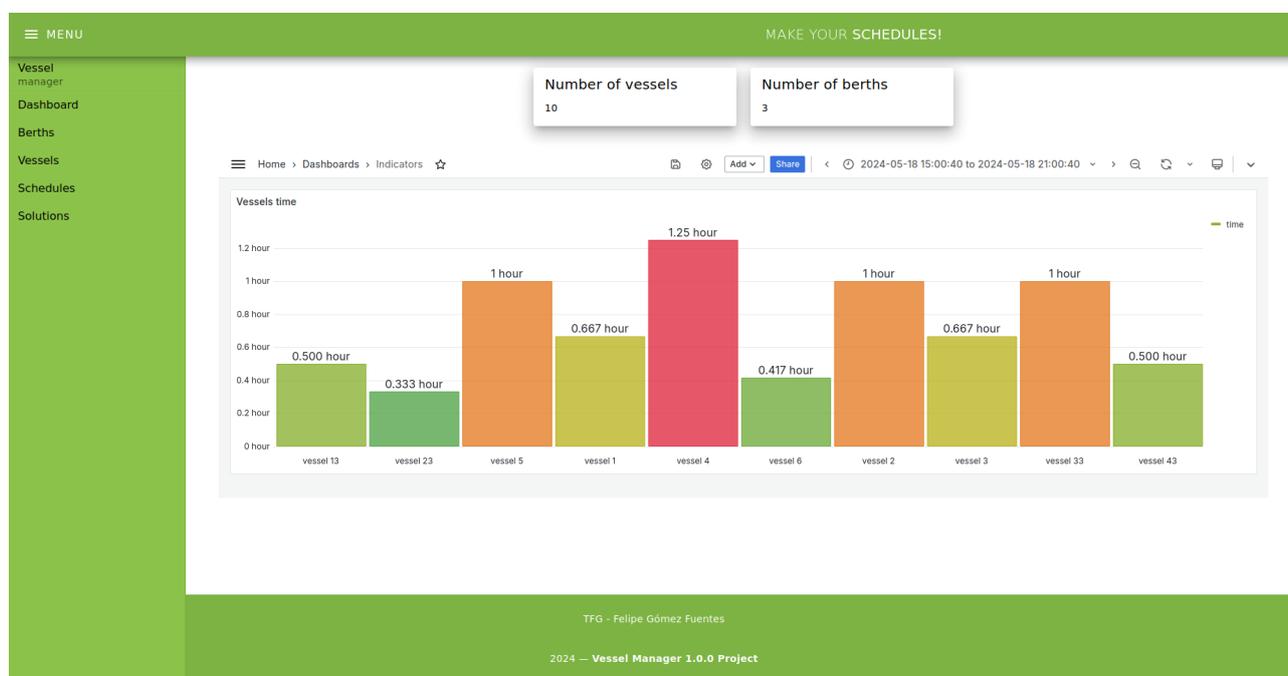


Figura 4.3: Página de inicio con el menú desplegado.

### 4.2.2. Lista de buques

Muestra una lista de todos los barcos registrados en el sistema junto con sus propiedades (figura 4.21). Además, permite el acceso a diversas funcionalidades para administrar los buques. En la tabla principal se muestra el nombre del buque, la fecha de llegada, la fecha de salida y las operaciones que realizadas.

| <input type="checkbox"/> | Name      | Arrival time      | Departure time    | Operations  |
|--------------------------|-----------|-------------------|-------------------|---|
| <input type="checkbox"/> | vessel 1  | 8/6/2023, 4:00:00 | 8/6/2023, 4:40:00 | <ul style="list-style-type: none"> <li>LOADING - Las Palmas -&gt; Los Cristianos</li> <li>Potatoes</li> </ul>   |
| <input type="checkbox"/> | vessel 2  | 8/6/2023, 4:00:00 | 8/6/2023, 5:00:00 |   |
| <input type="checkbox"/> | vessel 3  | 8/6/2023, 5:00:00 | 8/6/2023, 5:40:00 | <ul style="list-style-type: none"> <li>UNLOADING - Santa Cruz de Tenerife -&gt; Los Cristianos</li> <li>Bananas</li> <li>Tomatoes</li> </ul>                    |
| <input type="checkbox"/> | vessel 4  | 8/6/2023, 5:00:00 | 8/6/2023, 6:15:00 |   |
| <input type="checkbox"/> | vessel 5  | 8/6/2023, 5:30:00 | 8/6/2023, 6:30:00 |   |
| <input type="checkbox"/> | vessel 6  | 8/6/2023, 6:15:00 | 8/6/2023, 6:40:00 | <ul style="list-style-type: none"> <li>LOADING - Cadiz -&gt; Barcelona</li> <li>Cereals</li> <li>UNLOADING - Barcelona -&gt; Barcelona</li> <li>Vine</li> </ul> |
| <input type="checkbox"/> | vessel 13 | 8/6/2023, 4:00:00 | 8/6/2023, 4:30:00 | <ul style="list-style-type: none"> <li>LOADING - La Gomera -&gt; Lisboa</li> <li>Lettuce</li> <li>Potatoes</li> <li>Tomatoes</li> </ul>                         |
| <input type="checkbox"/> | vessel 23 | 8/6/2023, 5:00:00 | 8/6/2023, 5:20:00 | <ul style="list-style-type: none"> <li>LOADING - Buenos Aires -&gt; Santa Cruz de Tenerife</li> <li>Argentine mate</li> <li>Avocados</li> </ul>                 |
| <input type="checkbox"/> | vessel 33 | 8/6/2023, 6:00:00 | 8/6/2023, 7:00:00 |   |
| <input type="checkbox"/> | vessel 43 | 8/6/2023, 7:00:00 | 8/6/2023, 7:30:00 | <ul style="list-style-type: none"> <li>UNLOADING - Murcia -&gt; La Gomera</li> <li>Cereals</li> <li>Wine</li> </ul>   |

Figura 4.4: Página de lista de buques.

Además, se proporciona acceso a diferentes funcionalidades a través de los siguientes botones.

- Añadir buque.** Permite agregar un nuevo buque al sistema. En la pantalla podremos ver los campos a rellenar: nombre del barco, fecha de llegada y fecha de ida, como aparece en la figura 4.5.

Figura 4.5: Página para añadir barco

Al presionar el botón **Add operation** se despliega un segundo formulario (figura 4.6)

donde aparecen los siguientes campos: tipo de operación (carga/descarga), origen, destino, tiempo de operación, así como la opción de añadir contenedores.

The screenshot shows the 'Add vessel' form in the Vessel Manager application. The form is titled 'Add vessel' and includes the following fields and buttons:

- Navigation:** A green header bar with 'MENU' on the left and 'MAKE YOUR SCHEDULES!' on the right. Below it is an orange button labeled '← BACK TO LIST'.
- Vessel Information:**
  - 'Vessel's name' text input field.
  - 'Arrival time' date-time picker with format 'mm/dd/yyyy, --:-- --' and a calendar icon.
  - 'Departure time' date-time picker with format 'mm/dd/yyyy, --:-- --' and a calendar icon.
- Operation 1:**
  - 'Operation type' dropdown menu with 'LOADING' selected.
  - 'Source' text input field.
  - 'Target' text input field.
  - 'Operation time' numeric input field with value '0' and a refresh icon.
  - A red error message: 'Operation time must be greater than 0.'
- Containers:**
  - 'Container name 1' text input field.
  - 'REMOVE' button.
- Buttons:**
  - 'ADD CONTAINER' (blue), 'REMOVE OPERATION' (white), 'ADD OPERATION' (blue with checkmark), and 'ADD VESSEL' (blue with checkmark).

The footer of the page is green and contains the text: 'TFG - Felipe Gómez Fuentes' and '2024 — Vessel Manager 1.0.0 Project'.

Figura 4.6: Página para añadir operaciones

Cabe recalcar que el botón para añadir buque solo está disponible cuando se rellenan los campos obligatorios.

- Actualizar buque.** Permite actualizar la información de un buque existente en el sistema. Entre los campos a rellenar tenemos el nombre del barco, la fecha de llegada y la fecha de salida, tal y como aparece en la figura 4.5.

The screenshot shows the 'Update vessel' form in the Vessel Manager application. The form is titled 'Update vessel' and includes the following fields and buttons:

- Navigation:** A green header bar with 'MENU' on the left and 'MAKE YOUR SCHEDULES!' on the right. Below it is an orange button labeled '← BACK TO LIST'.
- Vessel Information:**
  - 'Vessel's name' text input field.
  - 'Arrival time' date-time picker with format 'mm/dd/yyyy, --:-- --' and a calendar icon.
  - 'Departure time' date-time picker with format 'mm/dd/yyyy, --:-- --' and a calendar icon.
- Buttons:**
  - 'ADD OPERATION' (blue with checkmark) and 'UPDATE VESSEL' (grey with checkmark).

The footer of the page is green and contains the text: 'TFG - Felipe Gómez Fuentes' and '2024 — Vessel Manager 1.0.0 Project'.

Figura 4.7: Página para actualizar barcos

Al igual que en la operación de añadir un buque, se pueden añadir operaciones tal y como se muestra en la figura 4.8.

← BACK TO LIST

### Update vessel

Vessel's name

Arrival time  
mm/dd/yyyy, --:-- --

Departure time  
mm/dd/yyyy, --:-- --

**Operation 1**

Operation type  
LOADING

Source

Target

Operation time  
0

Operation time must be greater than 0.

Container name 1

REMOVE

ADD CONTAINER REMOVE OPERATION

ADD OPERATION UPDATE VESSEL

TFG - Felipe Gómez Fuentes

2024 - Vessel Manager 1.0.0 Project

Figura 4.8: Página actualizar barcos añadiendo operaciones

Cabe recalcar que el botón para actualizar buque solo está disponible cuando se rellenan los campos obligatorios.

- **Borrar buque.** Permite eliminar un buque del sistema (figura 4.9).

← BACK TO LIST

### Remove vessel

¿Are you sure that you want to delete this vessel?

REMOVE VESSEL

TFG - Felipe Gómez Fuentes

2024 - Vessel Manager 1.0.0 Project

Figura 4.9: Página para eliminar barcos

- **Ver buque.** Permite ver más detalles sobre un buque específico, como sus operaciones con sus contenedores, sus horas de llegada y salida, su nombre y su identificador

(figura 4.10).

The screenshot shows a web application interface for vessel management. At the top, there is a green header with a menu icon and the text 'MAKE YOUR SCHEDULES!'. Below the header, there is a navigation bar with a 'BACK TO LIST' button. The main content area is titled 'Vessel details' and contains a table with the following data:

| Id                                   | Name     | ArrivalTime          | DepartureTime        | Operations  |
|--------------------------------------|----------|----------------------|----------------------|---|
| 68bbc5fe-0e69-4c73-b1c7-6e37c215af09 | vessel 6 | 6/8/2023, 6:15:00 AM | 6/8/2023, 6:40:00 AM | <ul style="list-style-type: none"> <li>• LOADING - Cadiz -&gt; Barcelona</li> <li>◦ Cereals</li> <li>• UNLOADING - Barcelona -&gt; Barcelona</li> <li>◦ Vine</li> </ul> |

At the bottom right of the table, there is a pagination control showing 'Items per page: All', '1-1 of 1', and navigation arrows.

At the bottom of the page, there is a green footer with the text 'TFG - Felipe Gómez Fuentes' and '2024 - Vessel Manager 1.0.0 Project'.

Figura 4.10: Página para ver detalles del buque

### 4.2.3. Lista de atraques

En la página de lista de atraques, los usuarios pueden ver una lista de todos los atraques disponibles junto con sus propiedades. Al igual que en la página de lista de buques, tienen acceso a funcionalidades para administrar los atraques. En la tabla principal vemos la fecha de apertura y la fecha de cierre, como se muestra en la figura 4.22.

The screenshot shows a web application interface for berth management. At the top, there is a green header with a menu icon and the text 'MAKE YOUR SCHEDULES!'. Below the header, there is a navigation bar with four buttons: 'ADD BERTH', 'UPDATE BERTH', 'DELETE BERTH', and 'SEE BERTH'. The main content area is titled 'Berths list' and contains a table with the following data:

| <input type="checkbox"/> | Name     | Opening date      | Closing date      |
|--------------------------|----------|-------------------|-------------------|
| <input type="checkbox"/> | Berth377 | 8/6/2023, 4:00:00 | 8/6/2023, 7:00:00 |
| <input type="checkbox"/> | Berth845 | 8/6/2023, 4:00:00 | 8/6/2023, 7:00:00 |
| <input type="checkbox"/> | Berth897 | 8/6/2023, 4:00:00 | 8/6/2023, 8:00:00 |

At the bottom right of the table, there is a pagination control showing 'Items per page: 10', '1-3 of 3', and navigation arrows.

At the bottom of the page, there is a green footer with the text 'TFG - Felipe Gómez Fuentes' and '2024 - Vessel Manager 1.0.0 Project'.

Figura 4.11: Página de lista de atraques.

Además, se proporciona acceso a las siguientes operaciones.

- **Añadir atraque.** Permite agregar un nuevo atraque al sistema. Para ello hay que rellenar los campos de fechas de apertura y cierre, tal y como aparece en la figura 4.12.

The screenshot shows a web interface for adding a berth. At the top, there is a green header bar with a menu icon and the text 'MAKE YOUR SCHEDULES!'. Below the header, there is an orange button labeled '← BACK TO LIST'. The main form is titled 'Add berth' and contains two date input fields: 'Open date' and 'Close date', both with the placeholder 'mm/dd/yyyy, --:-- --' and a calendar icon. Below the date fields is a blue button labeled 'ADD BERTH'. At the bottom of the page, there is a green footer bar with the text 'TFG - Felipe Gómez Fuentes' and '2024 — Vessel Manager 1.0.0 Project'.

Figura 4.12: Página para añadir atraque

- **Actualizar atraque.** Permite actualizar la información de un atraque existente en el sistema. Para ello hay que rellenar los campos de fechas de apertura y cierre, tal y como aparece en la figura 4.13.

The screenshot shows a web interface for updating a berth. At the top, there is a green header bar with a menu icon and the text 'MAKE YOUR SCHEDULES!'. Below the header, there is an orange button labeled '← BACK TO LIST'. The main form is titled 'Update berth' and contains two date input fields: 'New start date' and 'New close date', both with the placeholder 'mm/dd/yyyy, --:-- --' and a calendar icon. Below the date fields is a blue button labeled 'UPDATE BERTH'. At the bottom of the page, there is a green footer bar with the text 'TFG - Felipe Gómez Fuentes' and '2024 — Vessel Manager 1.0.0 Project'.

Figura 4.13: Página del para actualizar atraque

- **Borrar atraque.** Permite eliminar un atraque del sistema.

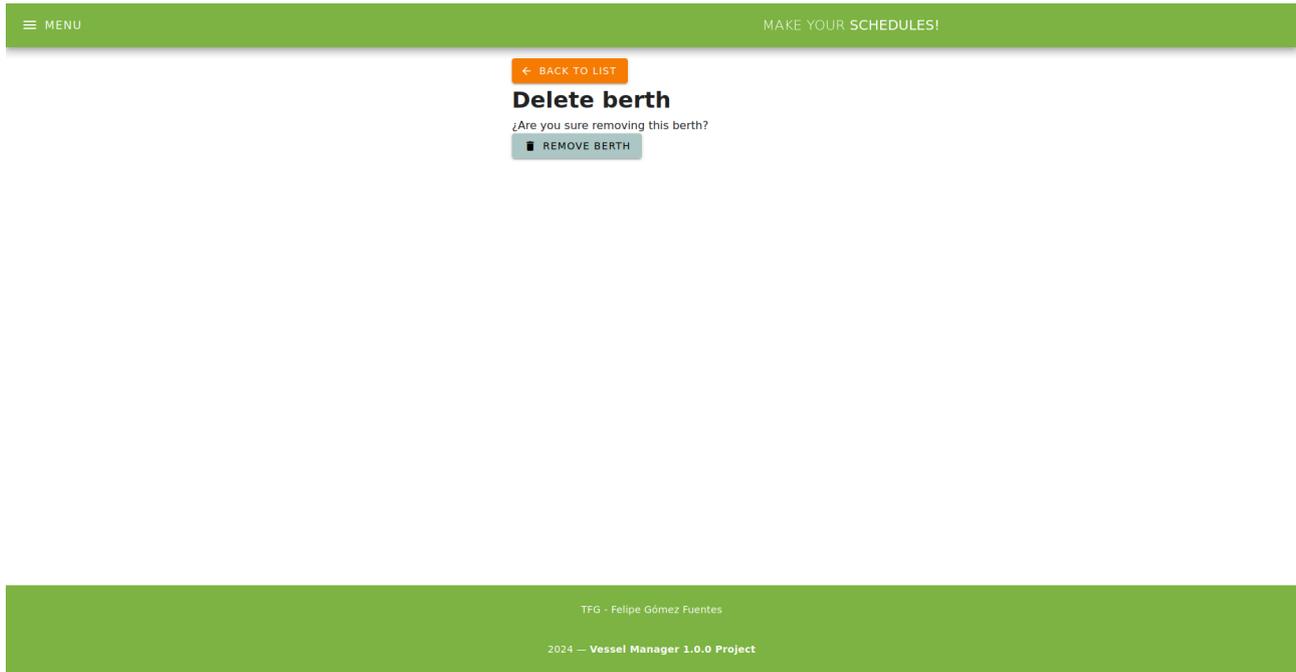


Figura 4.14: Página para eliminar atraque

- **Ver atraque.** Permite ver más detalles sobre un buque específico, como su identificador, así como su fecha de apertura y cierre.

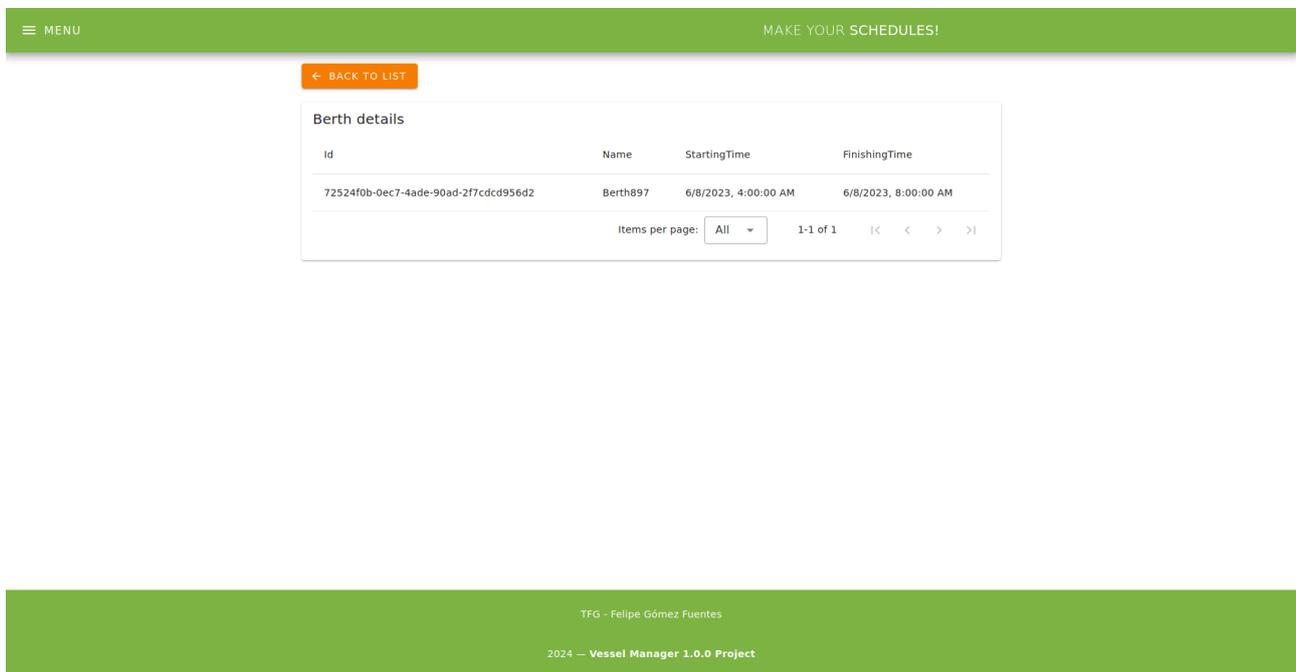


Figura 4.15: Página para ver detalles del atraque

#### 4.2.4. Página de planificaciones

En la página de planificaciones (figura 4.24), los usuarios pueden ver una lista de todas las planificaciones generadas por el sistema. Además, tienen la opción de ver el cuadro de mando de una planificación elegida de la lista y descargarla en formato JSON.

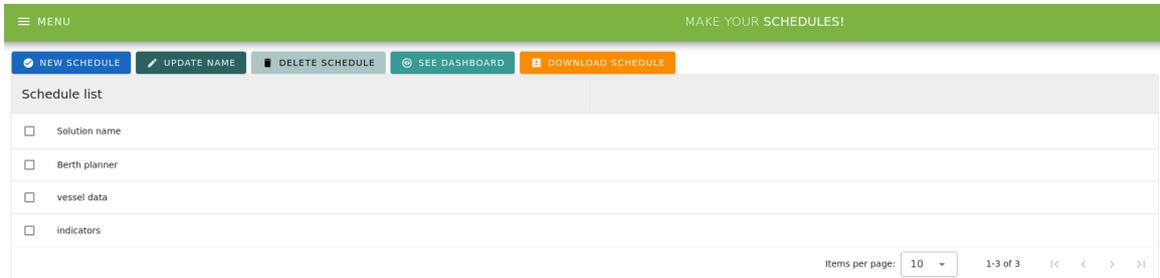


Figura 4.16: Página de la lista de planificaciones.

En esta página se tiene acceso a las siguientes funcionalidades:

- **Nueva planificación.** Permite crear una nueva planificación a partir del intervalo de tiempo deseado. Ejecuta el algoritmo de asignación de buques a atraques. Un ejemplo de pantalla de nueva planificación se muestra en la figura 4.17.



Figura 4.17: Página de para añadir una planificación

- **Actualizar nombre de planificación.** Permite actualizar el nombre de una planificación existente en el sistema (figura 4.18).

Figura 4.18: Página para actualizar el nombre de una planificación

- **Borrar planificación.** Permite eliminar una planificación del sistema (figura 4.19).

Figura 4.19: Página para eliminar una planificación

- **Ver planificación.** Permite ver más detalles sobre la planificación a través de un cuadro de mando (figura 4.23). En el cuadro de mando se muestra la información detallada de la planificación, incluyendo su nombre y su identificador. Además, se

presenta un diagrama donde el eje X representa los atraques y el eje Y los buques asignados a cada atraque para cada periodo de tiempo.

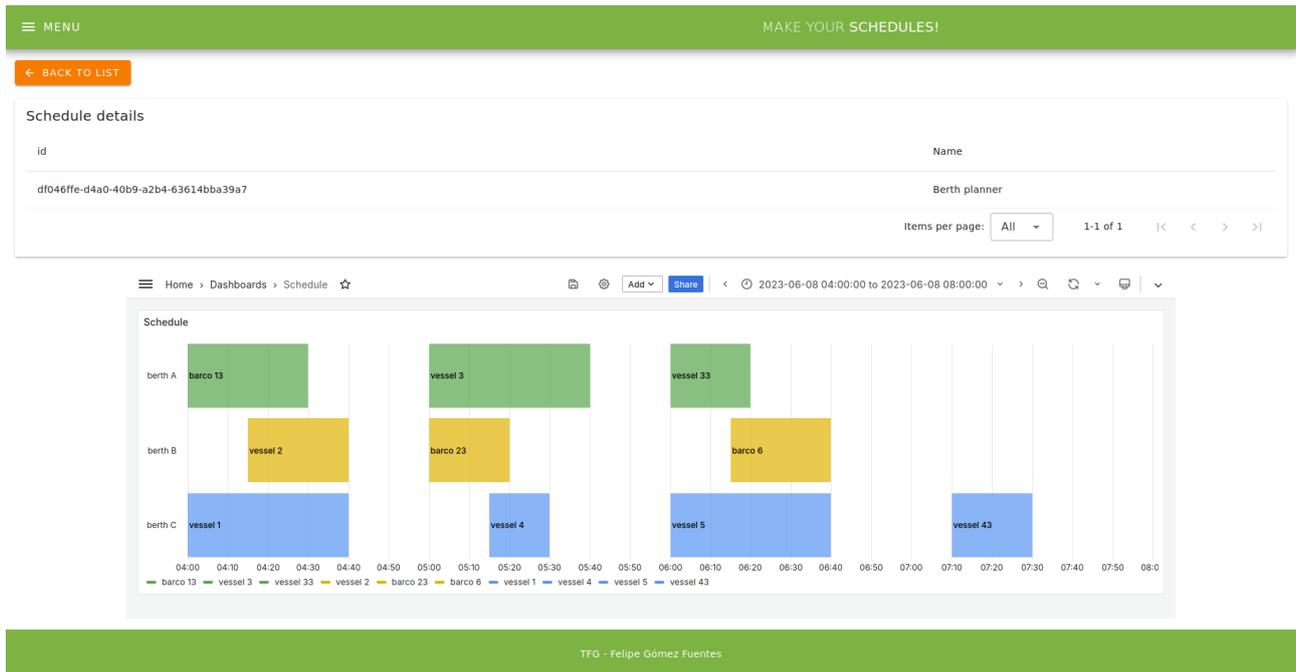


Figura 4.20: Acción del botón See Dashboard

- **Download schedule:** Permite descargar la planificación en formato JSON.

### 4.3. Caso práctico

En esta sección se describe un caso práctico del uso de la aplicación, donde se muestra como realizar una planificación de la llegada de buques en los distintos atraques desde cero.

El primer paso consiste en introducir los buques en el sistema a partir de la lista de buques, rellenando los campos del formulario correspondiente (figura 4.21).

| MENU  |           | MAKE YOUR SCHEDULES! |                   |   |
|---|-----------|----------------------|-------------------|---|
| <a href="#">ADD VESSEL</a> <a href="#">UPDATE VESSEL</a> <a href="#">DELETE VESSEL</a> <a href="#">SEE VESSEL</a> |           |                      |                   |   |
| Vessels list  |           |                      |                   |   |
| <input type="checkbox"/>  | Name      | Arrival time         | Departure time    | Operations  |
| <input type="checkbox"/>  | vessel 1  | 8/6/2023, 4:00:00    | 8/6/2023, 4:40:00 | <ul style="list-style-type: none"> <li>LOADING - Las Palmas -&gt; Los Cristianos</li> <li>Potatoes</li> </ul>   |
| <input type="checkbox"/>  | vessel 2  | 8/6/2023, 4:00:00    | 8/6/2023, 5:00:00 |   |
| <input type="checkbox"/>  | vessel 3  | 8/6/2023, 5:00:00    | 8/6/2023, 5:40:00 | <ul style="list-style-type: none"> <li>UNLOADING - Santa Cruz de Tenerife -&gt; Los Cristianos</li> <li>Bananas</li> <li>Tomatoes</li> </ul>                    |
| <input type="checkbox"/>  | vessel 4  | 8/6/2023, 5:00:00    | 8/6/2023, 6:15:00 |   |
| <input type="checkbox"/>  | vessel 5  | 8/6/2023, 5:30:00    | 8/6/2023, 6:30:00 |   |
| <input type="checkbox"/>  | vessel 6  | 8/6/2023, 6:15:00    | 8/6/2023, 6:40:00 | <ul style="list-style-type: none"> <li>LOADING - Cadiz -&gt; Barcelona</li> <li>Cereals</li> <li>UNLOADING - Barcelona -&gt; Barcelona</li> <li>Vine</li> </ul> |
| <input type="checkbox"/>  | vessel 13 | 8/6/2023, 4:00:00    | 8/6/2023, 4:30:00 | <ul style="list-style-type: none"> <li>LOADING - La Gomera -&gt; Lisboa</li> <li>Lettuce</li> <li>Potatoes</li> <li>Tomatoes</li> </ul>                         |
| <input type="checkbox"/>  | vessel 23 | 8/6/2023, 5:00:00    | 8/6/2023, 5:20:00 | <ul style="list-style-type: none"> <li>LOADING - Buenos Aires -&gt; Santa Cruz de Tenerife</li> <li>Argentine mate</li> <li>Avocados</li> </ul>                 |
| <input type="checkbox"/>  | vessel 33 | 8/6/2023, 6:00:00    | 8/6/2023, 7:00:00 |   |
| <input type="checkbox"/>  | vessel 43 | 8/6/2023, 7:00:00    | 8/6/2023, 7:30:00 | <ul style="list-style-type: none"> <li>UNLOADING - Murcia -&gt; La Gomera</li> <li>Cereals</li> <li>Wine</li> </ul>   |

Items per page: 10 1-10 of 10 < > >|

Figura 4.21: Lista de buques.

Una vez añadidos los buques deseados, se realiza el mismo proceso para los atraques, a través de la lista de atraques (4.22).

| MENU  |          | MAKE YOUR SCHEDULES! |                   |
|---|----------|----------------------|-------------------|
| <a href="#">ADD BERTH</a> <a href="#">UPDATE BERTH</a> <a href="#">DELETE BERTH</a> <a href="#">SEE BERTH</a> |          |                      |                   |
| Berths list   |          |                      |                   |
| <input type="checkbox"/>  | Name     | Opening date         | Closing date      |
| <input type="checkbox"/>  | Berth377 | 8/6/2023, 4:00:00    | 8/6/2023, 7:00:00 |
| <input type="checkbox"/>  | Berth845 | 8/6/2023, 4:00:00    | 8/6/2023, 7:00:00 |
| <input type="checkbox"/>  | Berth897 | 8/6/2023, 4:00:00    | 8/6/2023, 8:00:00 |

Items per page: 10 1-3 of 3 < > >|

Figura 4.22: Lista de atraques.

Una vez se poseen los datos necesarios, se crea una nueva planificación indicando su nombre así como el intervalo de tiempo en que se desarrolla (4.17).

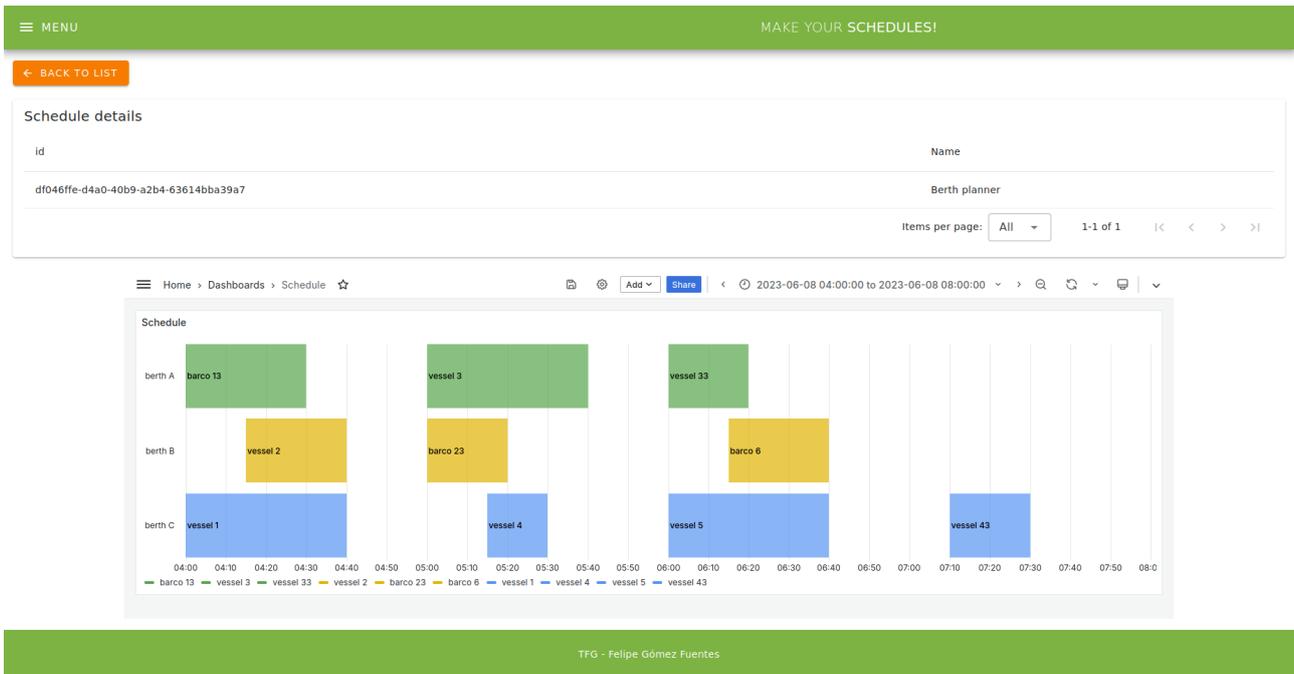


Figura 4.23: Cuadro de mando de la planificación.

Una vez creada la planificación, el sistema muestra una página con los detalles de la planificación y un cuadro de mando correspondiente (4.23).

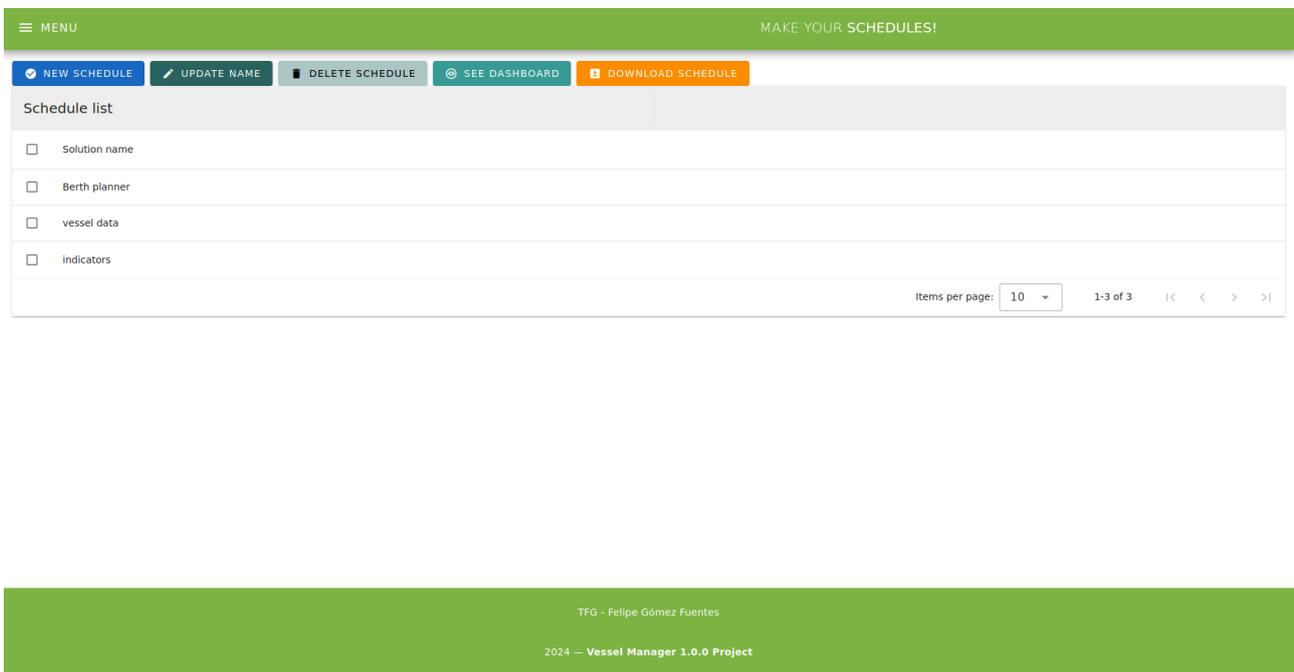


Figura 4.24: Lista de planificaciones.

Para interactuar con las planificaciones existentes, se puede acceder a la lista de planificaciones (4.24), desde donde es posible actualizar el nombre, ver el cuadro de mando o eliminar cualquier planificación disponible.

# Capítulo 5

## Presupuesto

En la sección de presupuesto se realiza una estimación de los costes para realizar el proyecto. En este apartado se detallan los costos asociados a la planificación, creación del prototipo, estudio de herramientas, investigación del ámbito, implementación front-end y back-end de la solución propuesta y realización del algoritmo, documentación y experimentación.

Para establecer el precio por hora de lo que costaría el proyecto se tiene en cuenta el salario medio que obtiene un desarrollador full-stack junior en España. De esta forma se han extraído los datos de diversos portales web que publican esta información, estos portales han sido: Glassdoor<sup>1</sup>, Talent<sup>2</sup> e Indeed<sup>3</sup>. Tras revisar los datos se ha establecido un precio por hora de 17,00€.

En la tabla 5.1 se especifican las tareas y las horas invertidas, así como el precio final de cada tarea, el tiempo total y el precio final.

| <b>Tarea</b>             | <b>Nº Horas</b> | <b>Precio final de la tarea (€)</b> |
|--------------------------|-----------------|-------------------------------------|
| Planificación            | 10              | 170 €                               |
| Prototipado              | 20              | 340 €                               |
| Estudio de tecnologías   | 50              | 850 €                               |
| Investigación            | 20              | 340 €                               |
| Frontend                 | 120             | 2.040 €                             |
| Backend                  | 100             | 1.700 €                             |
| Integración Grafana      | 40              | 680 €                               |
| Desarrollo del Algoritmo | 60              | 1.020 €                             |
| Documentación            | 40              | 680 €                               |
| Experimentación          | 20              | 340 €                               |
| <b>TOTAL</b>             | <b>480</b>      | <b>8160 €</b>                       |

Tabla 5.1: Presupuesto

<sup>1</sup><https://www.glassdoor.es>

<sup>2</sup><https://es.talent.com>

<sup>3</sup><https://es.indeed.com/career/desarrollador-junior/salaries>

# Capítulo 6

## Conclusiones

En este Trabajo de Fin de Grado se ha desarrollado una aplicación Full Stack que permite gestionar de forma eficiente los atraques de un puerto, con un enfoque en la aplicación del algoritmo de Berthing Allocation Problem (BAP). Este algoritmo aborda específicamente los desafíos relacionados con la asignación de atraques en puertos.

La aplicación ha sido estructurada en dos componentes principales: un front-end y un back-end. El front-end se ha desarrollado utilizando el framework Vue.js y su librería Vuetify, mientras que el back-end se ha implementado con el framework Spring. El aprendizaje de estas tecnologías ha representado uno de los principales desafíos de este proyecto, además de proporcionar la oportunidad de trabajar en el desarrollo Full Stack.

Durante el proceso de desarrollo se han enfrentado varias dificultades, principalmente relacionadas con la novedad de las tecnologías y herramientas utilizadas. Sin embargo, todas estas dificultades han sido superadas, lo que ha permitido ampliar los conocimientos y la experiencia en el campo de la Ingeniería Informática.

### 6.1. Líneas futuras

Las posibles mejoras y líneas futuras de este proyecto se centran en la finalización y mejora de características incompletas, así como en la adición de nuevas funcionalidades no consideradas durante el desarrollo del Trabajo de Fin de Grado. Algunas propuestas de mejora para el futuro incluyen:

- Introducción de datos en crudo. Permitir a los usuarios introducir datos desde un fichero en formato JSON. Con esto se evita la introducción manual de datos.
- Aplicación móvil. Adaptar la aplicación para que funcione como una aplicación móvil, lo que aumentaría su accesibilidad y conveniencia para los usuarios.
- Implementación de técnicas de aprendizaje automático. Utilizar técnicas de aprendizaje automático para analizar patrones en los datos históricos de llegadas de buques y así generar predicciones más precisas sobre futuras llegadas.

Estas líneas futuras representan oportunidades para mejorar la aplicación y expandir su funcionalidad, lo que podría aumentar su utilidad y beneficio en diversos contextos de aplicación.

# Capítulo 7

## Conclusions

In this Final Degree Project, a Full Stack application has been developed to efficiently manage the berths of a port, focusing on the application of the Berthing Allocation Problem (BAP) algorithm. This algorithm specifically addresses the challenges related to berth allocation in ports.

The application has been structured into two main components: a front-end and a back-end. The front-end was developed using the Vue.js framework and its Vuetify library, while the back-end was implemented with the Spring framework. Learning these technologies has been one of the main challenges of this project, in addition to providing the opportunity to work on Full Stack development.

Several difficulties were encountered during the development process, mainly related to the novelty of the technologies and tools used. However, all these difficulties were overcome, allowing the expansion of knowledge and experience in the field of Computer Engineering.

### 7.1. Future Directions

The possible improvements and future directions of this project focus on completing and enhancing incomplete features, as well as adding new functionalities that were not considered during the development of the Final Degree Project. Some proposed improvements for the future include:

- Raw data input. Allow users to input data from a file in JSON format. This avoids manual data entry.
- Mobile application. Adapt the application to work as a mobile app, which would increase its accessibility and convenience for users.
- Implementation of machine learning techniques. Use machine learning techniques to analyze patterns in historical ship arrival data to generate more accurate predictions of future arrivals.

These future directions represent opportunities to improve the application and expand its functionality, which could increase its utility and benefit in various application contexts.

# Bibliografía

- [1] R. J. Contreras Aguilar. Análisis de los costos de transferencias, alimentación y re-estibas en operaciones de buques de crowley, guatemala. enero 2023.
- [2] Conferencia de las Naciones Unidas sobre Comercio y Desarrollo. Informe de la junta de comercio y desarrollo. *Reuniones ejecutivas 33ª, 34ª y 35ª, 21º período extraordinario de sesiones y 51º período de sesiones Asamblea General Documentos Oficiales Quincuagésimo noveno período de sesiones Suplemento No. 15 (A/59/15)*, 2004.
- [3] Luis García Durán. Simulación mediante arena del flujo de mercancías en el puerto de algeciras. 2019.
- [4] CEPAL (Comisión Económica para América Latina y el Caribe). Transporte marítimo y puertos: Desafíos y oportunidades en busca de un desarrollo sostenible en américa latina y el caribe. *Recursos Naturales e Infraestructura*, 2015. Documento preparado por Octavio Doerr, Azhar Jaimurzina, Gabriel Pérez, Ricardo J. Sánchez y Gordon Wilmsmeier, funcionarios de la División de Recursos Naturales e Infraestructura (DRNI) de la CEPAL. Agradecimientos especiales a Francisca Pinto por su búsqueda de antecedentes estadísticos y a Pablo Chauvet por sus comentarios y aportes. Las opiniones expresadas en este documento son de exclusiva responsabilidad de los autores y pueden no coincidir con las de la Organización. ISSN 1680-9017, LC/L.4119, Copyright © Naciones Unidas. Impreso en Naciones Unidas, Santiago de Chile. S.15-01003.