



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Gestión de la Incertidumbre en Problemas Reales que Surgen en una Terminal Marítima de Contenedores

*Uncertainty Management in Real Problems in a
Maritime Container Terminal*

Jonathan Trujillo Estévez

La Laguna, 4 de julio de 2016

Doña **María Belén Melián Batista**, con N.I.F. 44.311.040-E profesora Titular de Universidad adscrita al Departamento Ingeniería Informática y de Sistemas de la Universidad de La Laguna y Don **Christopher Expósito Izquierdo**, con N.I.F. 78.851.649-J becario de investigación adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna,

C E R T I F I C A N

Que la presente memoria titulada:

"Gestión de la Incertidumbre en Problemas Reales que Surgen en una Terminal Marítima de Contenedores"

ha sido realizada bajo su dirección por Don **Jonathan Trujillo Estévez**, con N.I.F. 54.109.521-C.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 4 de julio de 2016

Agradecimientos

A María Belén Melián Batista y a Christopher Expósito Izquierdo por su labor en la dirección de este proyecto.

A Yayza Hernández Lorenzo por su inestimable apoyo en este proyecto.

A mi familia y amigos.

Resumen

El objetivo del presente Trabajo de Fin de Grado es abordar el problema de reparto de mercancías que se presenta en las terminales marítimas de contenedores, mediante la utilización de herramientas de simulación.

Gracias a la herramienta de simulación AnyLogic, cuyo núcleo está basado en el lenguaje de programación Java, se modeló el problema de reparto de mercancías de manera que nos permita optimizar el número de recursos disponibles en la terminal marítima de contenedores. Optimizar dichos recursos es importante pues éstos, influyen directamente en los términos económicos que implican a las terminales portuarias.

Para poder desarrollar este trabajo fue necesario la realización de un estudio previo sobre las terminales marítimas para comprender su funcionamiento correctamente, así como una aproximación a la herramienta de simulación AnyLogic y el correcto conocimiento sobre el lenguaje de programación Java. Seguidamente se implementó el modelo del problema y finalmente se procedió a la optimización del mismo.

Palabras clave: *Terminal marítima de contenedores, Puerto, Simulación, Optimización, Java, AnyLogic, Modelado.*

Abstract

The aim of this Final Degree Work is address the problem of goods' deliver in maritime container terminals, using simulation tools.

The aim of this Final Degree Work is to address the problem of delivering goods in maritime container terminals by using simulation tools. Thanks to the simulation tool AnyLogic, whose core is based on the Java programming language, the problem of distribution of goods with the goal of optimizing the available resources in the maritime container terminal was modeled. The optimization of these resources is essential given the fact that they have a direct economic impact in port terminals.

In order to succeed in this work, it was necessary to perform a preliminary study on maritime container terminals to understand their functioning properly. Moreover, it was required an approximation to the AnyLogic simulation tool and the correct knowledge about the Java programming language. Then, the problem model was implemented in AnyLogic and finally, we proceeded to the optimization of the designed model.

Keywords: *Maritime Container Terminal, Port, Simulation, Optimization, Java, AnyLogic, Model.*

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Alcance	3
1.3. Motivación del Proyecto	4
1.4. Estructura de la Memoria	4
2. Reparto de Mercancías en Terminales Marítimas de Contenedores	6
2.1. Introducción	6
2.2. Descripción	6
3. Simulación y Optimización	11
3.1. Introducción a la Simulación	11
3.1.1. Arena Software Simulation	11
3.1.2. AnyLogic	12
3.2. Introducción a la Optimización	14
3.3. Lenguaje de Programación	16
4. Implementación	19
4.1. Estructura del Modelo	19
4.2. Clase Principal <i>Main</i>	20
4.3. Clientes	22
4.4. Barcos Portacontenedores	23
4.5. Transporte Terrestre de Contenedores	24
4.6. Contenedores	24
4.7. Operarios de la Terminal	25
4.8. Pedidos	25
4.9. Terminal Marítima de Contenedores	26
5. Modelo de Simulación	29
6. Conclusiones y líneas futuras	34

7. Summary and Conclusions	36
8. Presupuesto	37
9. Bibliografía	38

Índice de figuras

1.1. Esquema de una terminal marítima de contenedores	1
1.2. Puertos localizados en Canarias	4
2.1. Diagrama de descarga de un barco	7
2.2. Diagrama de resolución de un pedido	8
2.3. Nueva base de contenedores Dársena del Este, S.C. de Tenerife	9
3.1. Interfaz Arena Software Simulation	12
3.2. Interfaz AnyLogic	14
3.3. Interfaz Eclipse IDE	17
4.1. Listado de clases del proyecto	19
4.2. Mapa georreferenciado de simulación	20
4.3. Lógica del funcionamiento de los barcos	21
4.4. Listado de objetos de la clase Main	22
4.5. Estructura interna de los objetos <i>Almacen</i>	23
4.6. Representación gráfica de los barcos	24
4.7. Estructura de la clase <i>Camion</i>	24
4.8. Estructura de la clase <i>Contenedor</i>	25
4.9. Estructura de la clase <i>Pedido</i>	25
4.10 Estructura de la clase Puerto	26
5.1. Vista gráfica de la ejecución	30
5.2. Objetivo principal a optimizar	31
5.3. Parámetros de la optimización	31
5.4. Requisitos a cumplir en la optimización	32
5.5. Resultados tras la optimización	32
5.6. Porcentajes de utilización de los recursos de la terminal . . .	33

Índice de tablas

8.1. Presupuesto aproximado del proyecto 37

Capítulo 1

Introducción

En este capítulo se realiza una introducción al Trabajo Fin de Grado desarrollado. Concretamente, se describe el contexto, alcance y motivaciones del mismo. Finalmente, se describe la organización del resto de la presente memoria.

1.1. Contexto

Una terminal marítima de contenedores, Figura 1.1, es una interfaz que conecta los medios de transporte terrestres y los medios de transporte marítimos dentro de una cadena de suministros. Las terminales marítimas de contenedores se encuentran habitualmente organizadas en tres áreas funcionales bien definidas:

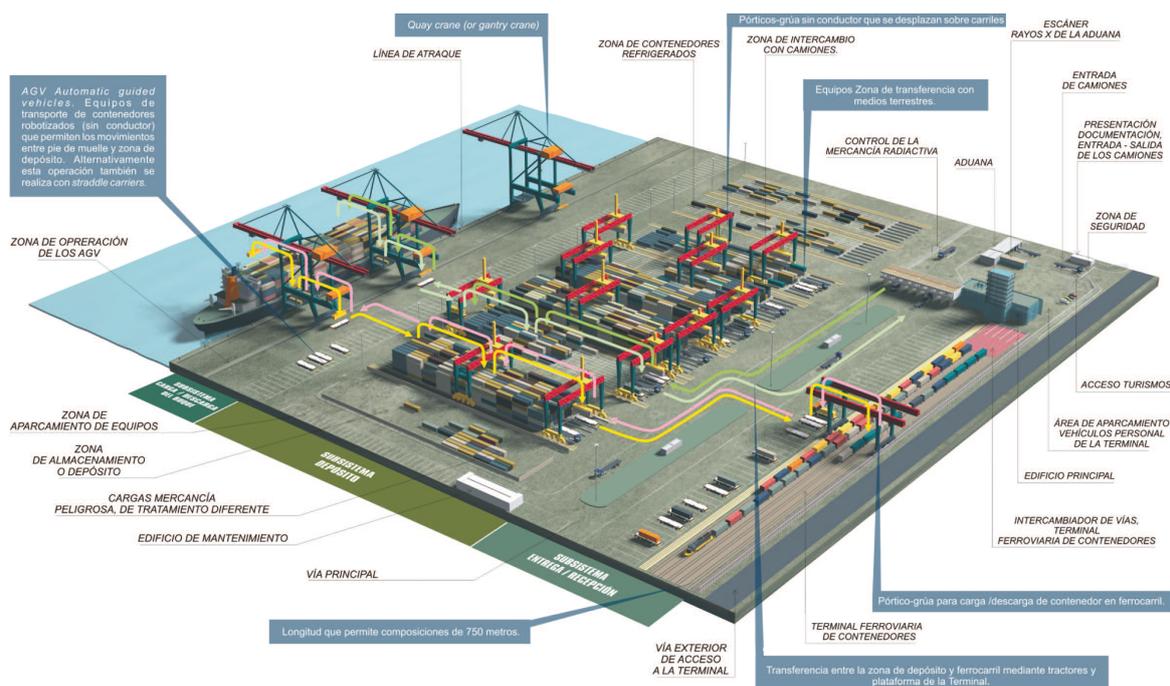


Figura 1.1: Esquema de una terminal marítima de contenedores

- *Lado del mar*. Es la parte de la terminal dedicada al atraque de los barcos de contenedores que llegan a la terminal para su carga y descarga.
- *Patio de contenedores*. Lugar que hace las veces de almacén para los contenedores hasta que éstos son retirados.
- *Lado de la tierra*. Es la interfaz que conecta el patio de contenedores con el transporte terrestre.

La sincronización de estas tres áreas funcionales, permiten el correcto funcionamiento de una terminal portuaria.

Los barcos portacontenedores que llegan a las terminales marítimas tienen que ser atracados en el muelle, en la parte denominada anteriormente como *lado del mar*. Una vez los barcos han atracado, hay que realizar una comprobación del lugar donde se pretende almacenar los contenedores que los barcos van a depositar en el patio de contenedores de la terminal. Cuando el lugar de los contenedores ha sido localizado, se procede a la descarga y transporte de los contenedores de los barcos, mediante las grúas apropiadas de las que disponen las terminales. Estas grúas cogen los contenedores a descargar y los depositan en los vehículos internos de la terminal, los cuales se encargan de mover los contenedores desde la línea de atraque hasta el lugar previamente seleccionado del patio de contenedores. Al llegar a este lugar, los contenedores son movidos desde los vehículos y depositados en la sección correspondiente mediante grúas. Estos serán los pasos realizados con todos y cada uno de los contenedores que tengan que ser descargados de los barcos.

Los barcos no solo se encargan de entregar contenedores a la terminal portuaria, sino que también cuentan con un listado de contenedores almacenados en el patio de contenedores de la terminal marítima, los cuales han de llevarse. Esto implica que el proceso que se realiza para la descarga, también es realizado de forma inversa para la carga de los barcos.

Estos dos procesos de carga y descarga se realizan de manera simultánea. Ambos procesos requieren de una alta cantidad de tiempo para ser completados, lo hace que la duración de la estancia de los barcos portacontenedores en la terminal sea larga.

Los contenedores almacenados en el patio de contenedores no solo salen de la terminal portuaria vía marítima, en barcos, como se comentó con anterioridad, sino que también muchos de ellos salen de la terminal vía terrestre.

Los contenedores almacenados en la terminal, son requeridos por clientes como pueden ser las grandes superficies, distribuidores, concesionarios, etc. Cuando uno de estos clientes realiza el pedido de un contenedor, éste es preparado para su retirada. Dependiendo de la infraestructura interna que presente la terminal, el contenedor puede ser manejado de diferentes formas. Pueden ser los propios vehículos internos de la terminal los que muevan los contenedores hasta la zona de espera de los camiones que los llevarán hasta su destino, o pueden ser los propios camiones finales los que acudan a buscar el contenedor hasta el lugar donde se encuentra dentro del patio de la terminal. La terminal también puede contar con su propia flota de camiones que será la encargada de transportar los contenedores hasta el cliente que lo ha solicitado, y posteriormente volver al puerto.

Éste es un primer acercamiento a las terminales portuarias. En el capítulo dos, se especificará de manera más detallada, el funcionamiento concreto que tendrá la terminal que se ha modelado durante el proceso de simulación.

1.2. Alcance

El alcance de este proyecto consta de describir el problema del transporte de mercancías, vía terrestre, de las terminales portuarias y darle solución utilizando una herramienta de simulación apropiada para el trabajo. Para ello fue necesario dividir el trabajo en objetivos a realizar. Dichos objetivos son los siguientes:

- Estudiar las terminales marítimas portuarias.
- Seleccionar y aprender el uso de las herramientas de simulación a emplear.
- Importar un mapa georreferenciado.
- Implementar los clientes.
- Implementar el puerto.
- Implementar el transporte terrestre de mercancías.
- Implementar el transporte marítimo de mercancías.
- Implementar el sistema de almacenamiento de la mercancía.
- Optimizar el modelo.

1.3. Motivación del Proyecto

De acuerdo con el informe del banco mundial¹, el tráfico anual de contenedores en el año 2014 creció hasta los 679,264,658,4 TEUs².

El archipiélago canario cuenta, tal y como se puede ver en la Figura 1.2, con una extensa red compuesta por 46 puertos, de los cuales 27 son puertos comerciales y 19 puertos deportivos. Cabe destacar el puerto de Santa Cruz de Tenerife, ya que es el puerto que se utilizará para el desarrollo del proyecto.

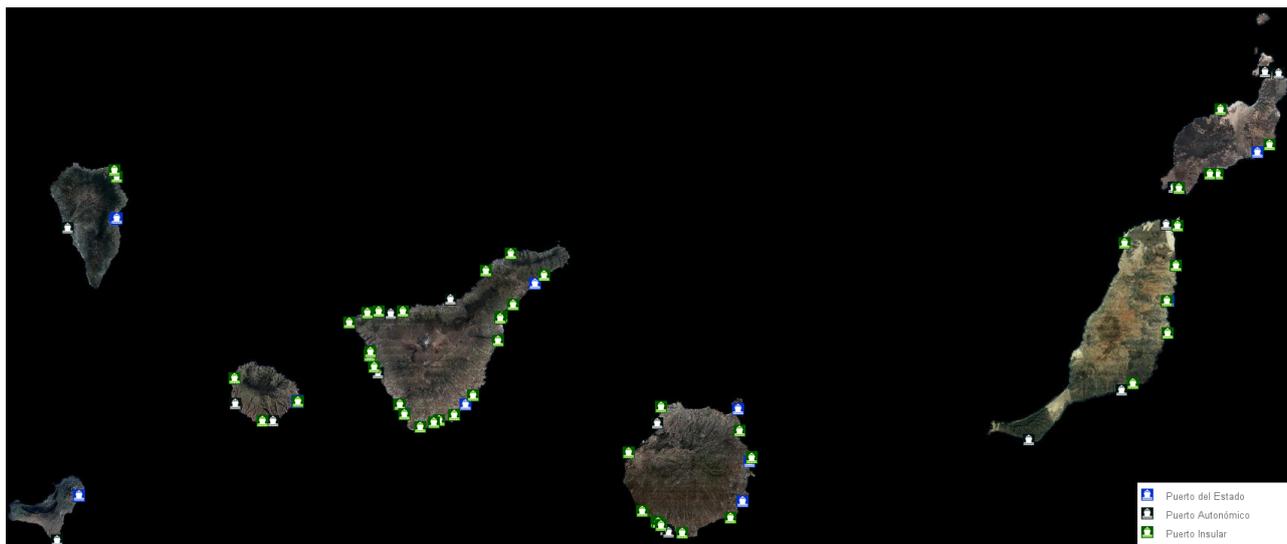


Figura 1.2: Puertos localizados en Canarias

Hoy en día, uno de los principales problemas a los que se enfrentan los puertos, y en particular las terminales marítimas de contenedores, es el de alcanzar una gestión eficiente de sus recursos, tanto materiales como humanos. Por ello, este trabajo intenta dar solución a la necesidad del Puerto de Santa Cruz de Tenerife de transportar la mercancía vía terrestre hacia los diferentes solicitantes que se encuentran repartidos por la isla de Tenerife de una manera eficiente.

1.4. Estructura de la Memoria

El resto de esta memoria se encuentra dividida tal y como se muestra a continuación:

- **Capítulo 2:** Descripción del problema de optimización a resolver.

¹<http://datos.bancomundial.org/indicador/IS.SHP.GOOD.TU>

²Unidad equivalente a 20 pies

- **Capítulo 3:** Descripción de la logística empleada para resolver el proyecto, así como de las herramientas necesarias para llevar a cabo el desarrollo del objeto del proyecto.
- **Capítulo 4:** Descripción de la implementación llevada a cabo en el desarrollo del trabajo.
- **Capítulo 5:** Optimización del modelo de simulación desarrollado.
- **Capítulo 6:** Conclusiones obtenidas del trabajo realizado. Además de las próximas líneas de trabajo que servirán como guía para investigaciones futuras.
- **Capítulo 7:** Conclusiones obtenidas del trabajo realizado. Además de las próximas líneas de trabajo que servirán como guía para investigaciones futuras. Redactado en inglés.

Capítulo 2

Reparto de Mercancías en Terminales Marítimas de Contenedores

2.1. Introducción

El problema objeto de estudio en este Trabajo de Fin de Grado es el de optimizar el reparto de contenedores, almacenados en una terminal portuaria a los diferentes clientes que los soliciten vía terrestre.

2.2. Descripción

El problema de repartir los contenedores de la terminal portuaria, como la tarea que debe completarse para transportar un contenedor desde la terminal hasta el cliente que lo ha solicitado.

El problema antes mencionado, se aborda desde el punto de vista de la simulación, ya que en los métodos analíticos se suele hacer un gran número de suposiciones, mientras que los modelos de simulación permiten analizar sistemas de mayor complejidad.

El objetivo principal de la resolución de este problema y el centro primordial de este Trabajo de Fin de Grado es el de optimizar los recursos requeridos por la terminal portuaria para llevar a cabo la tarea de entrega de mercancía a los diferentes clientes demandantes de la misma. Para lograr este objetivo, es necesario tener en cuenta todo el recorrido realizado por cada uno de los contenedores que se reciben en la terminal portuaria. Desde el momento mismo de su llegada y hasta que éstos sean entregados a su destinatario (Figura 2.1).

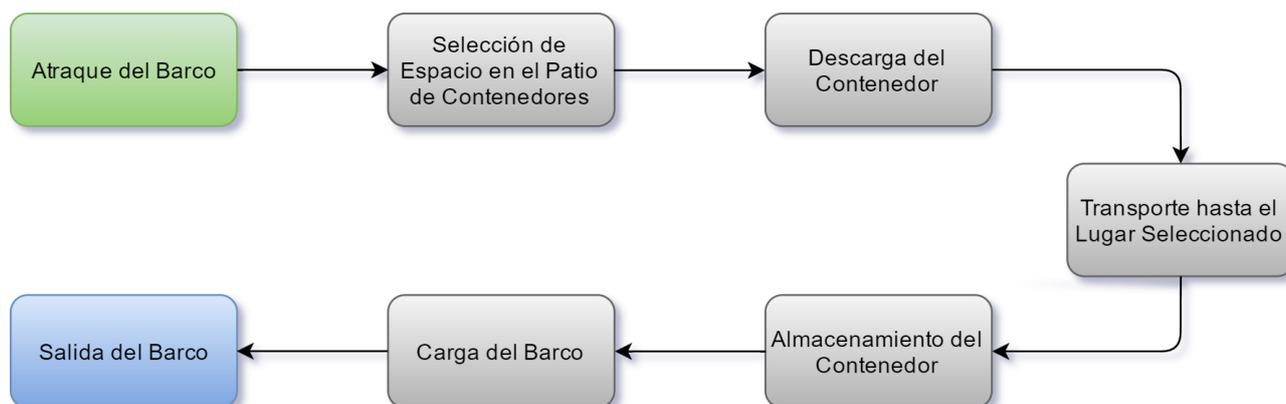


Figura 2.1: Diagrama de descarga de un barco

Cada barco portacontenedor debe llegar y atracar en el muelle. Una vez el barco se encuentra atracado en la terminal portuaria, éste permanece en la terminal un cierto periodo de tiempo definido mediante una función de distribución de probabilidad, la cual nos permite establecer un tiempo, que representará el rango de tiempo asociado a la estancia del barco en la terminal portuaria.

La función de distribución de probabilidad nos permite representar todo el proceso necesario para descargar un barco. Por lo tanto, esto conlleva que, en primer lugar se represente el atraque de un barco portacontenedores y posteriormente la selección del lugar en el patio de contenedores dónde se ubicarán los contenedores que han sido descargados del barco. A continuación, la descarga de los contenedores se realiza mediante grúas especializadas y se colocan en los vehículos de transporte interno de los que dispone la terminal portuaria para esta tarea. Seguidamente, se transporta el contenedor hasta la ubicación seleccionada en el patio de contenedores y este se deposita en su lugar. Y para concluir, la carga del barco portacontenedores, en caso de que requiera algún contenedor, de los que se encuentran almacenados en la terminal.

El patio de contenedores de la terminal portuaria es el lugar donde se almacenan temporalmente los contenedores que llegan a terminal hasta que éstos son entregados a los clientes.

Cuando un cliente solicita la retirada de un contenedor, dicho contenedor puede ser retirado de la terminal por vía terrestre, mediante camiones, o por vía marítima, mediante barcos. Cuando se requiere el transporte de algún contenedor, desde el patio a la terminal (por ejemplo para llevar a cabo la tarea de preparar un pedido), éste ha de ser cargado mediante grúas para ubicarlo en los vehículos internos de la terminal. Una vez colocado en estos vehículos es transportado hacia el lugar de atraque del

barco que ha solicitado dicho contenedor y finalmente, es cargado por una de éstas grúas de nuevo, para depositarlo en el barco portacontenedores.

Por otro lado, cuando la retirada es realizada por vía terrestre, el proceso a efectuar para completar la retirada, es el que se muestra en la Figura 2.2.

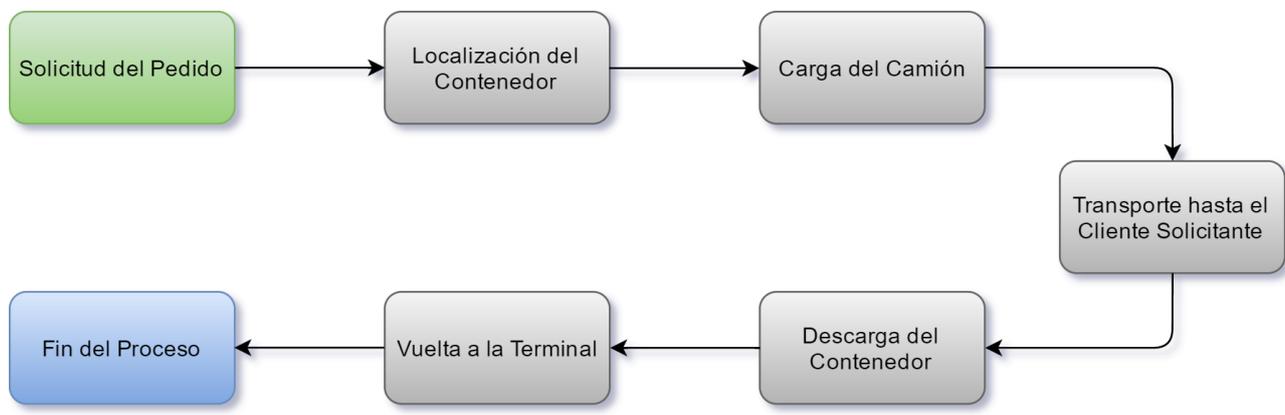


Figura 2.2: Diagrama de resolución de un pedido

En primer lugar, se recibe el pedido de alguno de los contenedores por parte de algún cliente, por lo que es preciso la retirada de dicho contenedor, que está almacenado en el patio de contenedores de la terminal portuaria. Esta petición trae asociada consigo un identificador único que permite diferenciar y discriminar el contenedor pertinente, que se ha de entregar en el pedido.

Seguidamente debe realizarse la búsqueda del contenedor en el patio de contenedores. Esta búsqueda es posible gracias al identificador que posee cada contenedor. El identificador de los contenedores es único, lo que permite diferenciarlos de todos los demás.

Una vez el contenedor está localizado, se envía un camión a la ubicación oportuna y dicho camión es cargado mediante las grúas de la terminal portuaria.

La preparación del pedido para su envío, es realizada por los operarios que trabajan en la terminal marítima. Éstos son los encargados de manejar las grúas y camiones pertinentes para poder preparar el pedido que se ha solicitado.

Cuando el camión ya se encuentra preparado con el pedido y éste ha sido confirmado, abandona la terminal portuaria rumbo al destino de la entrega del pedido (la ubicación del cliente que ha realizado dicho pedido). Una

vez el camión llega a su destino, se realizan las maniobras de descarga de la mercancía y una vez acabadas éste regresará nuevamente a la terminal portuaria, terminando así su ciclo de trabajo.

Con el modelado de este problema se pretende determinar el número necesario de recursos para que el coste del mantenimiento de los mismos sea razonable. Es por esto que se deben optimizar los recursos de los que dispone la terminal, como son:

- Camiones, encargados de transportar la mercancía vía terrestre.
- Operarios, encargados de preparar los pedidos que llegan a la terminal portuaria.

No se puede contar con una cantidad desproporcionada de camiones y/o operarios que permita que siempre estén disponibles los recursos para su utilización, ya que esto supondría la existencia de recursos sin utilizar la mayor parte del tiempo, lo que generaría una caída considerable de la eficiencia de la terminal portuaria.



Figura 2.3: Nueva base de contenedores Dársena del Este, S.C. de Tenerife

Para el desarrollo de este proyecto y Trabajo de Fin de Grado se ha seleccionado la terminal marítima de Santa Cruz de Tenerife, como ejemplo de terminal encargada de llevar a cabo la tarea de reparto de mercancía. Esta terminal portuaria se puede apreciar en la Figura anterior 2.3.

Se ha seleccionado también a las grandes empresas implantadas en la zona geográfica de Tenerife, como los clientes potenciales solicitantes de

mercancías. Estos clientes són listados a continuación: Alcampo, Carrefour, Centro Comercial La Villa, Aluminios Cortizo, Pinturas Cin Canarias, Coelca, Disa, El Corte Ingles, Electroauto Indrustial Güimar, Frigorífico del Polígono, Grupo Insular Océano, Hierros Tenerife, JSP, Kalise, La Oficina, Lidel, Mercadona, Pikolin, PPL, Prefabricados Metálicos Canarios, Cerveza Reina, Rocasa, Semaca y Unelco.

Capítulo 3

Simulación y Optimización

En este capítulo se introduce la simulación y se habla de las herramientas principales de simulación. Se describe la optimización y su aparición en el problema de reparto de mercancías. También se describe el lenguaje de programación empleado para el desarrollo de este Trabajo de Fin de Grado.

3.1. Introducción a la Simulación

Las herramientas de simulación nos permiten la representación de problemas cotidianos, los cuales cuentan con cierto grado de incertidumbre en sus diferentes parámetros, lo que hace que la implementación de un modelo numérico de este problema se aleje de la realidad. Esto quiere decir que, en los modelos numéricos se necesita establecer con exactitud cada valor que se emplea a la hora de resolver el modelo, datos que en la realidad no conocemos, o que pueden variar dentro de un amplio margen, y es ahí donde las herramientas de simulación nos permiten abordar desde un punto de vista más realista estas situaciones.

Existen muchas herramientas de simulación, cada una de ellas enfocadas en su propio campo y para su uso concreto. Para este caso en particular había dos herramientas que prestaban los requisitos necesarios para poder dar solución al problema en cuestión, de las cuales hablaremos a continuación.

3.1.1. Arena Software Simulation

Arena es un potente software de modelado y simulación desarrollado por la compañía estadounidense Rockwell Automation ¹. Ha sido diseñado para

¹<https://www.arenasimulation.com/>

analizar el impacto que producen los cambios en diferentes situaciones reales. Dotado de gran flexibilidad y basado en el lenguaje de simulación SIMAN, permitiría perfectamente crear un modelo adecuado para el problema objeto de este Trabajo de Fin de Grado. Permitiendo incluir el análisis de contenedores, clientes y de la misma terminal portuaria.

Arena es un simulador intuitivo basado en la colocación y unión gráfica de diferentes módulos de procesos. En la Figura 3.1 se puede ver su interfaz.

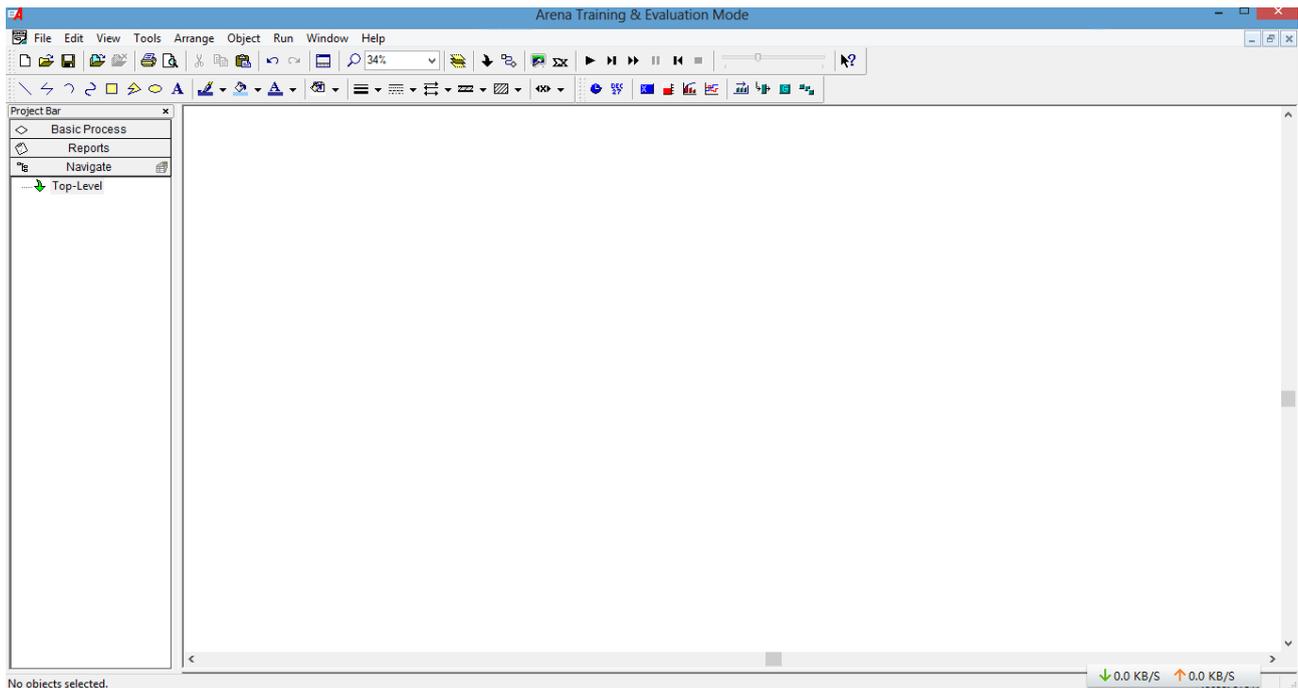


Figura 3.1: Interfaz Arena Software Simulation

Pese a sus grandes prestaciones y su facilidad de manejo, su elevado coste y su lenguaje de simulación poco común en comparación con Java, han hecho que Arena Software Simulation no haya sido el programa escogido para la resolución del problema objeto de este proyecto.

3.1.2. AnyLogic

Anylogic es una herramienta de simulación desarrollada por The AnyLogic Company². Es la herramienta que más métodos de simulación incluye. Desarrollado sobre el IDE Eclipse y ofreciéndonos gran potencia con su lenguaje de modelado gráfico, AnyLogic permite ampliar nuestros modelos mediante programación basada en Java. Todas estas características hacen

²<http://www.anylogic.com/>

de este software la elección correcta para abordar el problema objeto de este proyecto.

Podemos encontrarnos con diferentes versiones del software desarrollado por The AnyLogic Company, ofreciendo cada una diferentes ventajas sobre la anterior.

- *Free Anylogic® PLE*. Personal Learning Edition, es la versión gratuita del software y la cuál ha sido seleccionada para llevar a cabo el Trabajo de Fin de Grado. Pese a tener ciertas funciones capadas con respecto a las otras versiones, ofrece todo lo necesario para desarrollar un modelo realista y detallado del problema.
- *AnyLogic® University Researcher*. Versión de pago creada para instituciones educativas. Ofrece algunas características más como pueda ser un modo debug y soporte técnico.
- *AnyLogic® Professional*. Versión completa del software, la cual incluye todas las características del mismo.

Debido a que ha sido desarrollado sobre el IDE de Eclipse, la interfaz de AnyLogic 3.2, se asemeja mucho a la ofrecida por Eclipse, facilitando la adaptación si previamente se ha trabajado en Eclipse. Así como los conocimientos de programación necesarios para poder hacer un uso efectivo de la herramienta y aprovechar su potencial se centra en el conocimiento del lenguaje de programación Java, siendo éste un lenguaje bastante popular.

Como se puede apreciar en la Figura 3.2, la interfaz de AnyLogic se encuentra dividida en 3 secciones principales, y una sección más de consola de errores que aparecerá si existen fallos de compilación o ejecución. A continuación, se describe cada una de estas secciones:

- *Sección Izquierda*. En esta sección podremos encontrar dos pestañas diferentes, una primera pestaña denominada *Projects* donde se mostrarán los diferentes proyectos que tengamos abiertos, así como todos los componentes de cada uno de ellos, mostrados mediante una estructura de árbol. Por otro lado, una segunda pestaña llamada *Palette*, la cual incluye todos los elementos que podemos añadir al proyecto separando éstos en grupos por características comunes.
- *Sección Central*. En esta sección se pueden visualizar cada uno de los objetos que se crean en el proyecto.

- *Sección Derecha.* En esta sección se puede encontrar la pestaña *Properties*, la cual muestra las propiedades del elemento que se tenga seleccionado en ese preciso instante.
- *Sección Consola de Errores.* Esta sección aparece en la zona inferior de la sección central, pero solo cuando se haya compilado o ejecutado el proyecto y éste tenga fallos. Se muestra una consola con el listado de errores que contenga el proyecto.

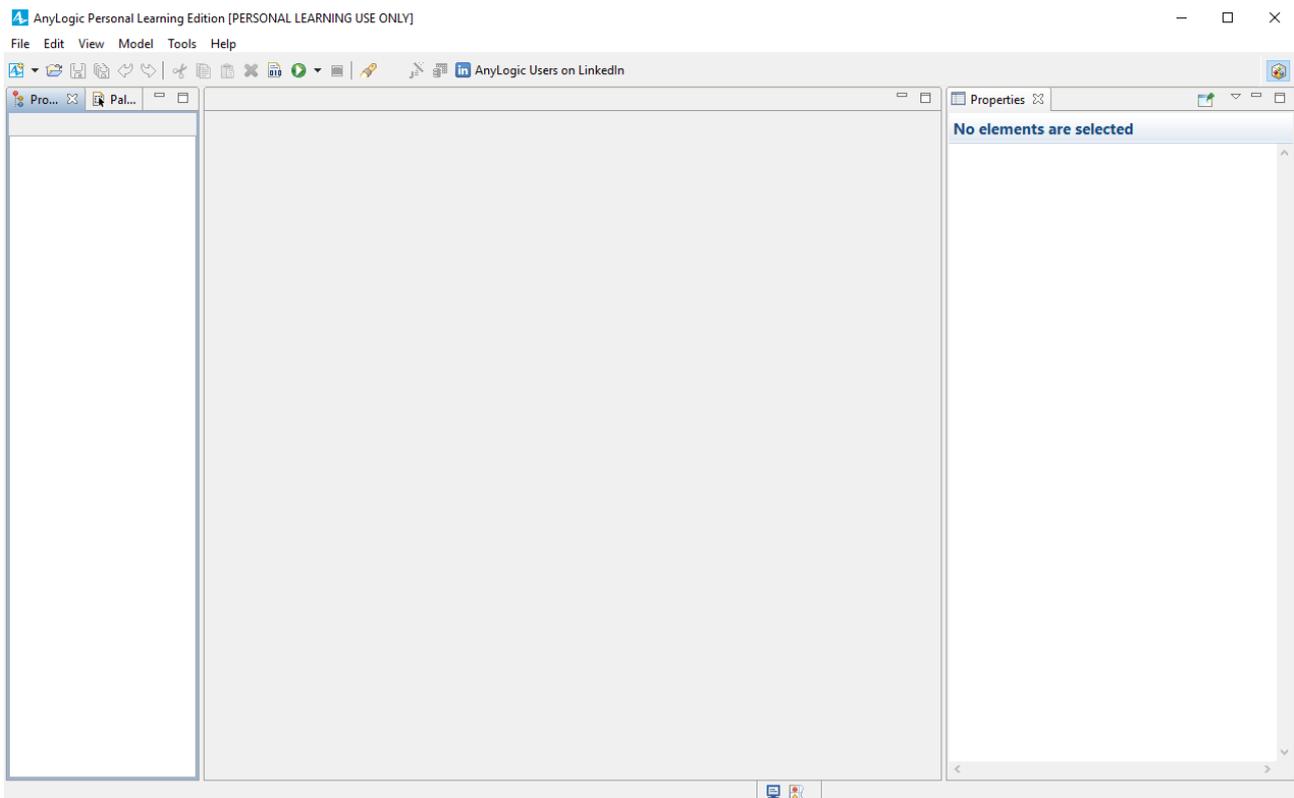


Figura 3.2: Interfaz AnyLogic

Todo esto hace de AnyLogic una herramienta para crear modelos y para simular muy potente y con una curva de aprendizaje asequible, ya que tanto el lenguaje de simulación como la interfaz son sencillas y habituales en el mundo de la programación.

3.2. Introducción a la Optimización

La optimización consiste en determinar los valores óptimos para los recursos que existen en un sistema. Establecer estos valores permite que el resultado sea el mejor posible. Esto se entiende como cambiar los valores

de los recursos obteniendo así un mayor rendimiento, ya sea haciendo que algo trabaje de manera más eficiente o utilizando un menor número de recursos.

Todas las empresas, ya sean pequeñas, medianas o grandes; cuentan con una serie de recursos a su disposición, los cuales son los encargados de realizar las funciones de esta empresa, y sin los que no se podría llevar a cabo las tareas de la empresa. Por esto es necesario un buen empleo de los recursos de los que disponemos. Ya que el objetivo no es solo realizar la tarea propuesta en el menor tiempo posible, sino que también tenemos que tener en cuenta que esta tarea sea realizada con el mínimo de recursos posibles. Por esto debe existir un equilibrio entre el tiempo necesario para desarrollar la tarea, y la cantidad de recursos que se emplean para la misma.

Es necesario establecer los objetivos, así como las tareas que hay que realizar y cuáles son los recursos que se utilizarán. Esto debe definirse de manera clara, para poder dar paso a la optimización. Ésta permite aumentar la productividad, mejorar los servicios y regular la cantidad de recursos, lo que posibilita en última instancia, una reducción de los costos en el desempeño de la actividad a desarrollar.

Dada esta primera idea global sobre la optimización, podremos centrarnos ya sobre la optimización en el problema que nos ocupa.

En el problema de reparto de mercancía vía terrestre, el objetivo principal es optimizar la utilización de los camiones necesarios para realizar esta entrega, al igual que los operarios encargados de preparar los pedidos de mercancía de la terminal portuaria. Estos camiones son un recurso del que dispone nuestra terminal, y por tanto, se deben optimizar, de manera que cuando nos llegue un pedido a la terminal, el cliente que realiza el pedido no tenga que esperar una cantidad excesiva de tiempo hasta que se hace efectivo este envío, pero tampoco que el número de camiones que espera en nuestra terminal sea desproporcional al número de pedidos que se deben cubrir.

Esto se debe a que es esencial encontrar un equilibrio entre el número de pedidos realizados por los clientes y la flota de camiones de la que dispone la empresa. Ya que no es interesante, ni asumir los costos del mantenimiento y almacenamiento de un número excesivo de camiones en la empresa, ni generar clientes insatisfechos debido a la ineficacia en el envío de los pedidos. Al igual que se debe optimizar todo lo referente a los camiones, es vital hacer lo propio con los operarios del patio de

contenedores, encargados de verificar qué contenedor pertenece a cada pedido y de cargarlos en los respectivos camiones para su entrega. Por lo tanto, el número de operarios debe ser suficiente para que esta tarea se pueda realizar de manera fluida y en consecuencia no se demoren demasiado en terminar un pedido y atender el siguiente y tampoco que estos operarios estén inactivos por un largo periodo de tiempo.

La optimización de estos recursos de los que se disponen en nuestro modelo, será realizada una vez tengamos la simulación montada por completo. AnyLogic proporciona las herramientas necesarias para optimizar estos recursos, y nos permite ver en tiempo de ejecución los porcentajes de utilización de cada uno de los recursos implementados en el modelo, permitiendo ajustarlos sobre la marcha y comprobar inmediatamente el impacto que suponen dichos cambios en nuestro sistema.

3.3. Lenguaje de Programación

Esta sección se basa en el lenguaje de programación Java. Java es el lenguaje en el cual está basado la herramienta de simulación *AnyLogic* que se ha utilizado para llevar a cabo el desarrollo de este Trabajo de Fin de Grado.

Como ya se mencionó en la sección de AnyLogic, el empleo de la herramienta del lenguaje de programación Java es una de las principales causas que han motivado la elección de este software. Java es un lenguaje bastante común dentro del ámbito de la programación, se trata de un lenguaje orientado a objetos muy potente que permite a los desarrolladores no tener que recompilar el programa para cada dispositivo, sino que una vez compilado puede ser ejecutado en cualquier otra máquina.

Desarrollado en unos inicios por *Sun Microsystems* y publicado en 1995, Java pertenece en la actualidad a la compañía *Oracle Corporation*³ la cual en 2009 adquirió a Sun Microsystems.

Una de las principales características de Java es que es un lenguaje orientado a objetos. Esto da lugar a poder crear entidades genéricas que permiten la reutilización de código posteriormente, a la par que facilita la estructuración del programa separando cada entidad que se necesite en un objeto único que la represente.

Java también cuenta con independencia de la plataforma, lo que permite que las aplicaciones escritas en Java puedan ejecutarse en cualquier tipo

³<http://www.oracle.com/technetwork/java/index.html>

de máquina, pues estas aplicaciones no dependen del hardware para ejecutarse. Esto se consigue al compilar el código, ya que se genera un archivo que es posteriormente interpretado por una máquina virtual Java.

Otra de las características más importantes de Java es su recolector de basura. Éste es el encargado de liberar la memoria. Así como un programador es el encargado de la creación de los objetos, es el propio Java el que decide cuándo un objeto debe ser destruido para liberar espacio, evitando así fugas de memoria.

La sintaxis de Java heredó características empleadas por el lenguaje de programación C++, eliminando aquellas que eran demasiado complejas.

En lo que respecta al rendimiento de Java. Algunas de sus características, como puede ser el recolector de basura entre otros, hace que éste sufra una penalización con respecto a lenguajes como C o C++. Por lo que en lo que respecta al tiempo de ejecución las aplicaciones de Java son algo más lentas que las de otros lenguajes. Esta pérdida de rendimiento viene principalmente del hecho de que Java es interpretada por una máquina virtual y ésto es más lento que utilizar el código de máquina interno de la plataforma dónde se ejecutará.

Como se ha comentado anteriormente Java es un lenguaje popular entre los programadores y ésto hace que exista una gran comunidad detrás de él y que existan IDE creados para facilitar el desarrollo de aplicaciones creadas en Java.

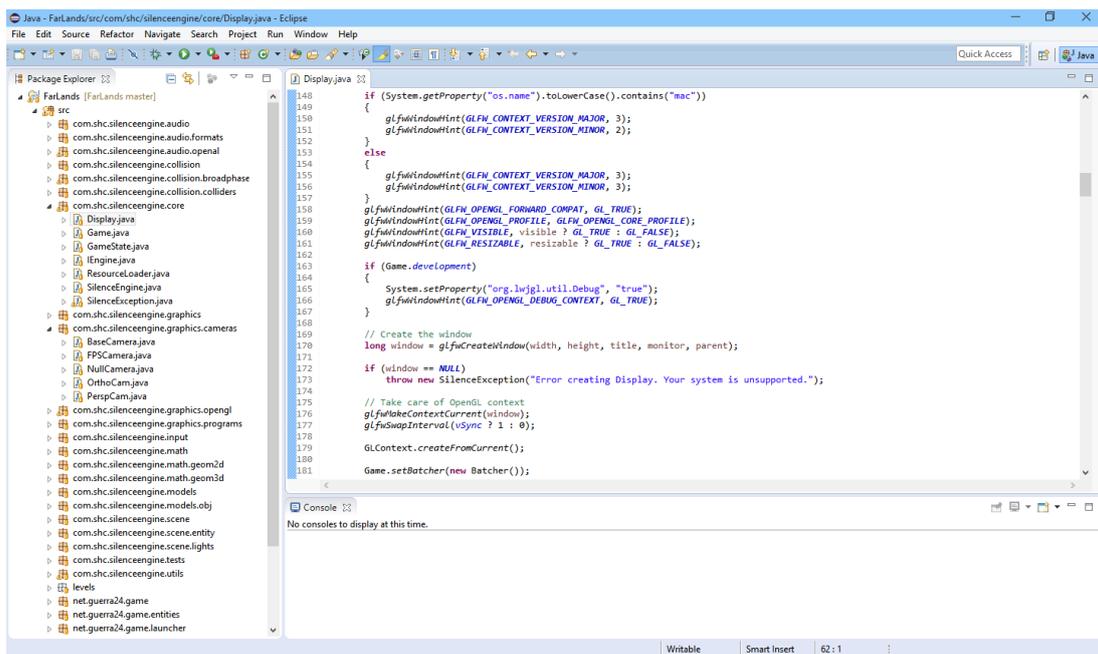


Figura 3.3: Interfaz Eclipse IDE

Unos de los IDE más famosos es Eclipse ⁴ y es el IDE sobre el que está creada la herramienta que se emplea en el desarrollo del proyecto. Cómo se puede apreciar en la Figura 3.3 la interfaz que presenta Eclipse es muy similar a la de AnyLogic 3.2 (que se muestra en apartados anteriores de este proyecto).

No obstante, cabe destacar que Eclipse no solo es empleado para desarrollar aplicaciones en Java, sino que también nos permite la utilización de otros lenguajes como por ejemplo: ANSI, C, C++, JSP, etc.

⁴<https://eclipse.org/home/index.php>

Capítulo 4

Implementación

En este capítulo se describe el desarrollo del trabajo realizado para la implementación del modelo de simulación. Se muestran los diferentes elementos cuya implementación ha sido necesaria para completar correctamente el modelo, como son los clientes, el puerto y el resto de elementos para el correcto funcionamiento de la simulación.

4.1. Estructura del Modelo

En este apartado se muestra de forma esquemática la estructura que presenta el proyecto, para ayudar en los apartados siguientes a una mejor comprensión de los mismos.

El modelo ha sido creado con la herramienta de simulación AnyLogic, bajo el nombre de *Puerto*. Siendo las clases implementadas en el modelo las mostradas en la siguiente Figura 4.1. Como se puede apreciar, las clases contienen un nombre autodescriptivo para facilitar la tarea de desarrollo.

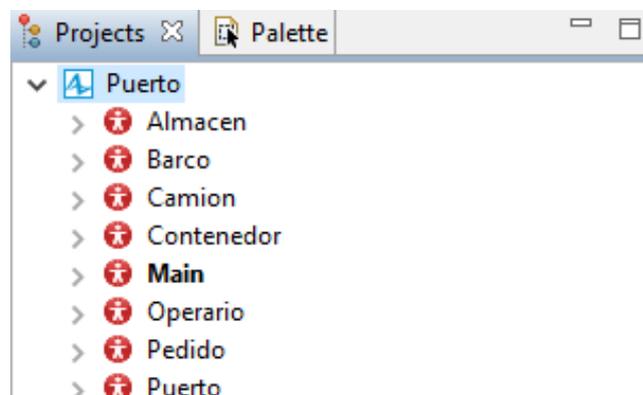


Figura 4.1: Listado de clases del proyecto

Las clases de la Figura 4.1 se listan a continuación junto con una breve descripción de las mismas:

- *Almacen*. Las instancias de esta clase representan a los diferentes clientes.
- *Barco*. Los barcos portacontenedores son instancias de esta clase.
- *Camion*. Clase que se instancia para dar vida a los camiones.
- *Contenedor*. Clase la cual nos representa a cada contenedor de la terminal.
- *Main*. Clase principal del proyecto que contiene el grueso del modelo.
- *Operario*. Clase que se instancia para crear a los operarios de la terminal.
- *Pedido*. Esta clase hace las veces de mensaje entre los almacenes y el puerto.
- *Puerto*. Encargado de gestionar la descarga de los barcos, así como la carga de los camiones y el envío de mercancía.

4.2. Clase Principal *Main*

Para poder resolver el problema que nos ocupa, se ha implementado un modelo formado por la clase principal, que llevara el peso de la simulación, y a ésta se le ha ido añadiendo el resto de clases y parámetros necesarios para completar el modelo.



Figura 4.2: Mapa georreferenciado de simulación

Esta clase es llamada *Main*. Alberga el mapa georeferenciado que podemos ver en la Figura 4.2 sobre el cual se basa nuestra simulación.

También es la encargada de gestionar la llegada de los barcos portacontenedores al muelle. En la siguiente Figura 4.3 se muestra la lógica empleada para el tratamiento de los barcos portacontenedores en el proyecto.

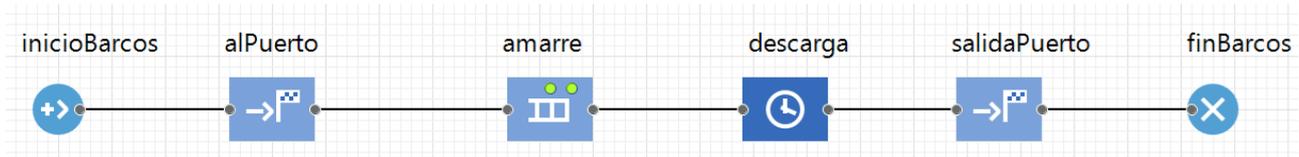


Figura 4.3: Lógica del funcionamiento de los barcos

Los barcos son creados mediante una función que nos permite determinar su frecuencia y cambiarla si es necesario, son enviados mediante las rutas marítimas, que proporciona el mapa, hacia el puerto. Allí pasan a ser amarrados para su posterior descarga, el tiempo de amarre y descarga es definido mediante una función que puede ser cambiada si es necesario para adaptarla mejor a las necesidades. Durante la descarga se ejecuta la función que añade nuevos contenedores al patio de carga, podemos ver un pseudocódigo de esta función a continuación 1.

Algoritmo 1: Función de descarga

```

1 Data: Integer nContenedores
2  $nContenedores \leftarrow \text{rand}(\text{minContenedores}, \text{maxContenedores})$ 
3 for  $i = 0; i < nContenedores; i++$  do
4   |  $\text{int } id = \text{rand}() \notin \text{PatioContenedores}$ 
5   |  $\text{PatioContenedores.add}(\text{new Contenedor}(id))$ 

```

Donde $nContenedores$ representa el número de contenedores que descargará el barco, $minContenedores$ y $maxContenedores$ son el número mínimo y el número máximo de contenedores que puede descargar un barco respectivamente. El entero id será el id representativo de cada contenedor, el cual no puede existir ya en el *PatioContenedores*.

Por último los barcos salen del puerto, y los objetos que los representan son destruidos, para evitar que se colapse la memoria.

La clase *Main* mantiene todos los parámetros organizados y permite consultarlos desde cualquiera de sus subclases. En la Figura 4.4 se puede

ver una lista de los diferentes objetos, con nombres autodescriptivos, que se han creado para realizar la correcta implementación del modelo propuesto.

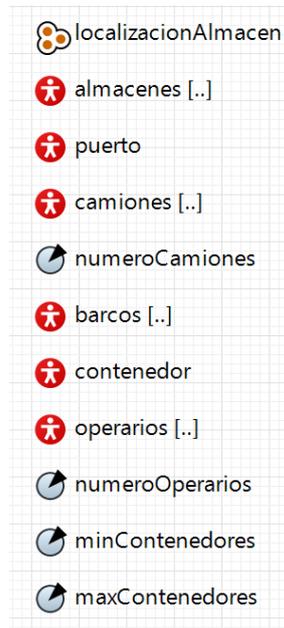


Figura 4.4: Listado de objetos de la clase Main

La creación de estas clases y estos parámetros en la clase *Main*, permite que se pueda acceder posteriormente desde las subclases de esta misma.

4.3. Clientes

Los clientes son representados en el proyecto mediante la clase denominada *Almacen*. Existe una lista de clientes la cuál se puede incrementar o decrementar según se crea necesario. Esta clase es la encargada de enviar peticiones de mercancía al puerto, en un periodo de tiempo ajustable seleccionado por el programador. En la Figura 4.5 se muestra la estructura que presenta cada cliente, para permitir una mejor descripción de como está organizado cada cliente.

En primer lugar arriba a la izquierda se puede ver un hangar de color amarillo, el cuál es la representación gráfica que tomará cada almacén durante la simulación del proyecto. Estos se encontrarán repartidos a lo largo de todo el territorio de la isla de Tenerife.

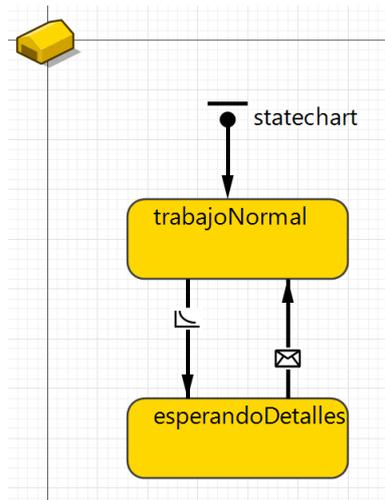


Figura 4.5: Estructura interna de los objetos *Almacen*

El funcionamiento de los almacenes consta de dos estados diferentes. Un primer estado de trabajo normal por parte del almacén y un estado de esperando mercancía, la transición del primer estado al segundo viene definida por una función de tiempo, que se puede variar en conveniencia. Una vez se pasa al segundo estado, quiere decir que el almacén ha enviado una petición, solicitando mercancía al puerto, y no se saldrá de este estado hasta que el puerto mande la mercancía pertinente al almacén. La creación del pedido por parte del almacén es representada en el pseudocódigo 2.

Algoritmo 2: Función de solicitud de mercancía

- 1 **Data:** *Integer id*
 - 2 $id \leftarrow rand()$
 - 3 $Pedido\ pedido = new\ Pedido(this, id)$
 - 4 $send(pedido, puerto)$
-

Una vez la mercancía llega al almacén, se abandona el estado de esperando mercancía y se vuelve al estado normal.

4.4. Barcos Portacontenedores

Los barcos portacontenedores, están representados en el proyecto mediante la clase *Barco*. Su función principal es aportar la representación gráfica durante la simulación del modelo. Son creados y destruidos, durante la ejecución del sistema. En la Figura 4.6 podemos ver el aspecto que presentan éstos durante la simulación.

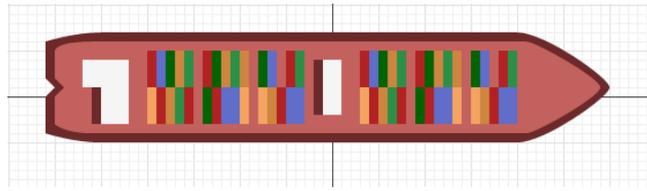


Figura 4.6: Representación gráfica de los barcos

Durante su movimiento, los barcos siguen las rutas marítimas establecidas por el mapa georreferenciado.

4.5. Transporte Terrestre de Contenedores

El transporte terrestre de los contenedores es realizado mediante los camiones, y éstos son creados por la clase denominada *Camion*. Esta clase es la encargada de representar gráficamente los camiones en la simulación. En la siguiente Figura 4.7 se muestra su contenido.

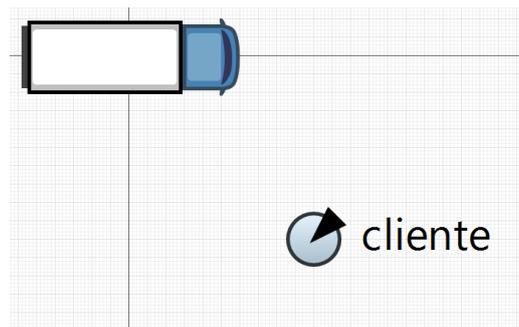


Figura 4.7: Estructura de la clase *Camion*

Por una lado, tenemos la imagen del camión que recorrerá las carreteras del mapa para entregar la mercancía y por otra parte, tenemos el parámetro *cliente*, el cual nos sirve para guardar el nombre del cliente que solicita la mercancía, así cada camión sabe cuál es su destino a seguir.

4.6. Contenedores

Los contenedores, son la mercancía que hay que repartir en el problema que nos ocupa. Para representarla se emplea la clase *Contenedor*. Esta clase no tiene representación gráfica, pues el elevado número de objetos de esta clase, colapsaría la representación gráfica.



Figura 4.8: Estructura de la clase *Contenedor*

Como se puede apreciar en la Figura anterior 4.8, pese a no contar con la representación gráfica como se ha dicho anteriormente, si que cuenta con un parámetro *id*, el cuál sirve para identificar cada contenedor, pues este identificador es único, es decir, cada contenedor cuenta con su propio identificador diferente al del resto. Este parámetro nos sirve para poder diferenciar los contenedores y saber que contenedor llevar a cada cliente.

4.7. Operarios de la Terminal

Los trabajadores de la terminal son representados mediante la clase *Operario*. Éstos son los encargados de verificar qué contenedor pertenece a cada pedido y de cargarlos en los respectivos camiones. No cuentan con representación gráfica pero su definición es necesaria para mantener una estructura ordenada en el proyecto.

4.8. Pedidos

La clase *Pedido* sirve para enlazar las solicitudes de mercancía entre almacén y puerto. No tiene un apartado visual a la hora de la simulación pero es muy importante para el funcionamiento del modelo. En la Figura 4.9 podemos ver la estructura que presenta esta clase.



Figura 4.9: Estructura de la clase *Pedido*

Como se puede ver en la Figura 4.9, la clase está compuesta por dos parámetros. Un primer parámetro llamado *representante*, que es el que almacena el nombre del almacén que realiza el pedido y un segundo parámetro denominado *identificador*, que guardar el identificador del contenedor solicitado por el almacén. Ambos parámetros son necesarios para que la clase *Puerto* pueda procesar correctamente los envíos.

4.9. Terminal Marítima de Contenedores

La terminal marítima está representada en el proyecto mediante la clase llamada *Puerto*. Esta clase contiene todo el esqueleto de funcionamiento del trabajo. En la Figura 4.10, mostrada a continuación, se ve la estructura completa de la clase. Como se puede apreciar, junto con la clase *Main* ésta es la clase con mayor contenido, pues es la encargada de gestionar los pedidos, los camiones y los operarios. Haciendo uso de estos recursos para poder realizar las entregas de la mercancía solicitada.

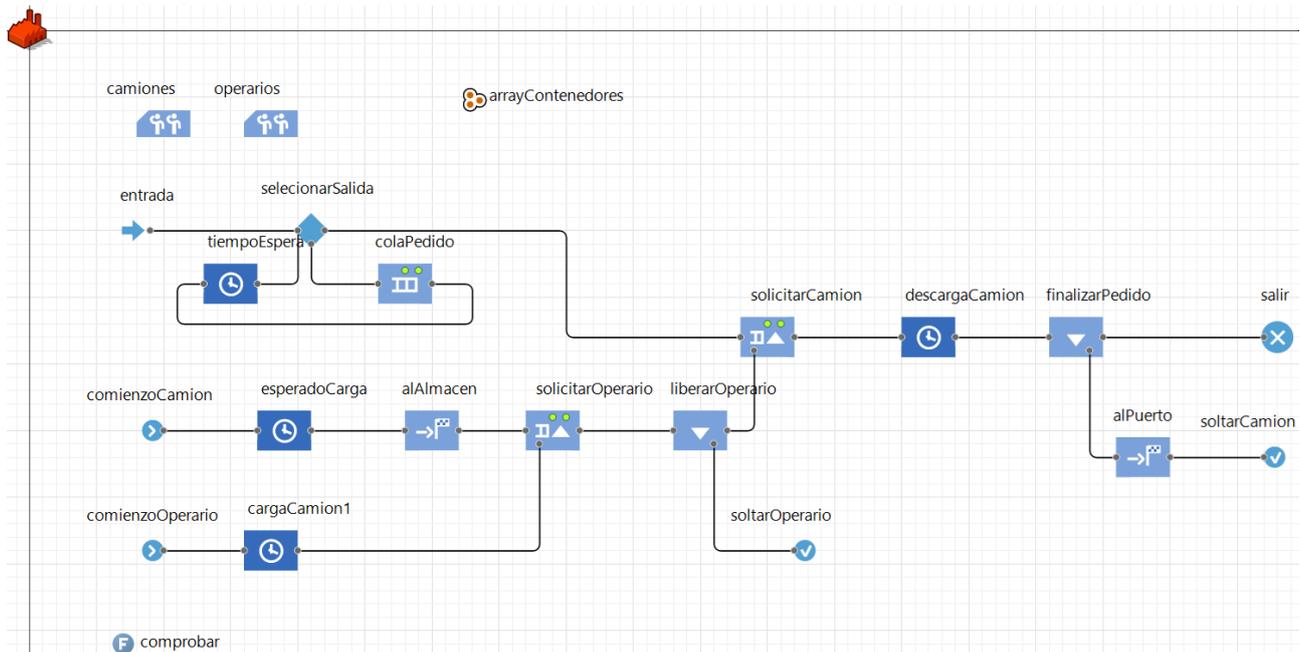


Figura 4.10: Estructura de la clase Puerto

- En la esquina superior izquierda se puede apreciar una estructura de fábrica, que dará vida gráficamente al puerto durante la simulación del proyecto.
- Se dispone de un conjunto de camiones, los cuáles serán utilizados durante la simulación para el transporte de la mercancía.
- También hay un conjunto de operarios, que serán los encargados de preparar y enviar los pedidos que llegan a la terminal portuaria.
- Se cuenta con un array de contenedores, que almacenará todos los contenedores que irán llegando a la terminal marítima hasta su posterior retirada.

- Cuenta con una función denominada *comprobar*, que se encargará de realizar la verificación cuando llega un pedido, de si el contenedor solicitado en dicho pedido, está almacenado en la terminal o nó.

El resto de contenido mostrado en la Figura 4.10 es el esquema lógico del comportamiento que sigue la terminal portuaria al recibir un pedido. Éste es detallado a continuación.

El puerto gestiona los pedidos generados por los almacenes. Cuando llega un pedido a la terminal, éste es introducido por el bloque *entrada*. Posteriormente, se envía el pedido al bloque *seleccionarSalida*, y es éste es el encargado de llamar a la función *comprobar* para verificar si el contenedor solicitado se encuentra en nuestra terminal, o hay que esperar a que llegue en un futuro. El procedimiento de la función *comprobar* es el mostrado en siguiente pseudocódigo.

Algoritmo 3: Función de Comprobación

```

1 Data: Integer id
2 Result: Boolean S
3 if PatioContenedores found id then
4   | S ← True
5 else
6   | S ← False
7 return S

```

Una vez se ejecuta la función, ésta puede devolver dos valores, *True*, implicando que el contenedor solicitado está almacenado en el patio de contenedores de la terminal, o *False*, lo que implica que el contenedor solicitado en el pedido aún no ha llegado a la terminal.

En el caso de que el contenedor no este aún en la terminal, el pedido pasa a un estado de espera definido mediante una función programable, que nos permite determinar durante cuanto tiempo tendremos el pedido esperando hasta que volvamos a realizar la comprobación, para saber si el contenedor que se solicita ya ha llegado a la terminal. Los pedidos que no pueden ser resueltos se almacenan en una cola, que presenta un comportamiento FIFO (First In First Out), esto permite que los pedidos que más tiempo llevan en la cola sean los primeros en volverse a comprobar. Cuando se cumple el tiempo de espera determinado los pedidos son comprobados de nuevo.

En caso contrario, es decir que el contenedor ya se encuentra localizado en el patio de contenedores, se pasa a la preparación de envío del mismo.

El pedido llega desde el *selecOutput* hasta el bloque *seize* donde se espera hasta que haya un camión disponible para preparar el contenedor. Cuando tenemos un camión libre, éste pasa al estado de *esperandoCarga* mientras los operarios de la terminal lo cargan, una vez acaba la carga del camión por parte de los operarios, finaliza también el estado de *esperandoCarga* del camión. Este tiempo que se tarda en cargar el camión está definido mediante funciones de distribución de probabilidad, para permitir ajustar este periodo al tiempo que se desee. Cuando finaliza éste tiempo el camión es enviado al cliente que ha solicitado el contenedor, y los operarios encargados de preparar el pedido son liberados de su tarea, estando disponibles de nuevo para preparar nuevos pedidos.

El camión viaja desde la terminal portuaria hacia su destino, utilizando las rutas que ofrece el mapa georreferenciado. Cuando llega a su destino pasa al estado de *descargaCamion*, este estado al igual que el encargado de la carga es totalmente programable. Una vez descargado el camión, se le envía un mensaje al cliente y se le informa de que el pedido que ha solicitado, ya ha sido entregado. Ésto permite que la terminal pueda cerrar la petición y enviar de vuelta a la terminal portuaria al camión. Cuando éste llega, es liberado y pasa a estar de nuevo disponible para realizar nuevas entregas.

Capítulo 5

Modelo de Simulación

En este capítulo se describen los resultados obtenidos de la ejecución del modelo propuesto en el capítulo 4.

Para la ejecución se ha tenido que dar valor a todas las variables existentes en el modelo. Estos valores pueden ser cambiados en cualquier momento. En este caso en concreto, los valores elegidos se listan a continuación:

- *Número de pedidos realizados:* Función de distribución de probabilidad triangular con un mínimo de 5, un máximo de 9 y una moda de 7 pedidos a la semana por cada cliente.
- *Número de barcos atracados:* Función de distribución de probabilidad uniforme con un mínimo de 3 y un máximo de 5 al día.
- *Número de operarios necesarios:* 2 operarios por pedido.
- *Número máximo de contenedores que se pueden almacenar en la terminal:* 1000 contenedores.
- *Número de contenedores que descarga cada barco:* Función de distribución de probabilidad uniforme con un mínimo de 10 y un máximo de 20 contenedores por barco.
- *Tiempo de esperar para volver a comprobar si un contenedor ha llegado:* Función de distribución de probabilidad uniforme con un mínimo de medio día y un máximo de un día completo.
- *Tiempo de espera para carga y descarga de los camiones:* Función de distribución de probabilidad uniforme con un mínimo de 2 y un máximo de 3 horas por camión.

- *Tiempo de espera para la descarga de los barcos:* Función de distribución de probabilidad triangular con un mínimo de 3, un máximo de 5 y una moda de 4 horas por barco.
- *Número de clientes existentes:* 25 clientes.

La Figura 5.1 muestra la imagen que presenta el modelo durante la ejecución del mismo con los parámetros especificados anteriormente.

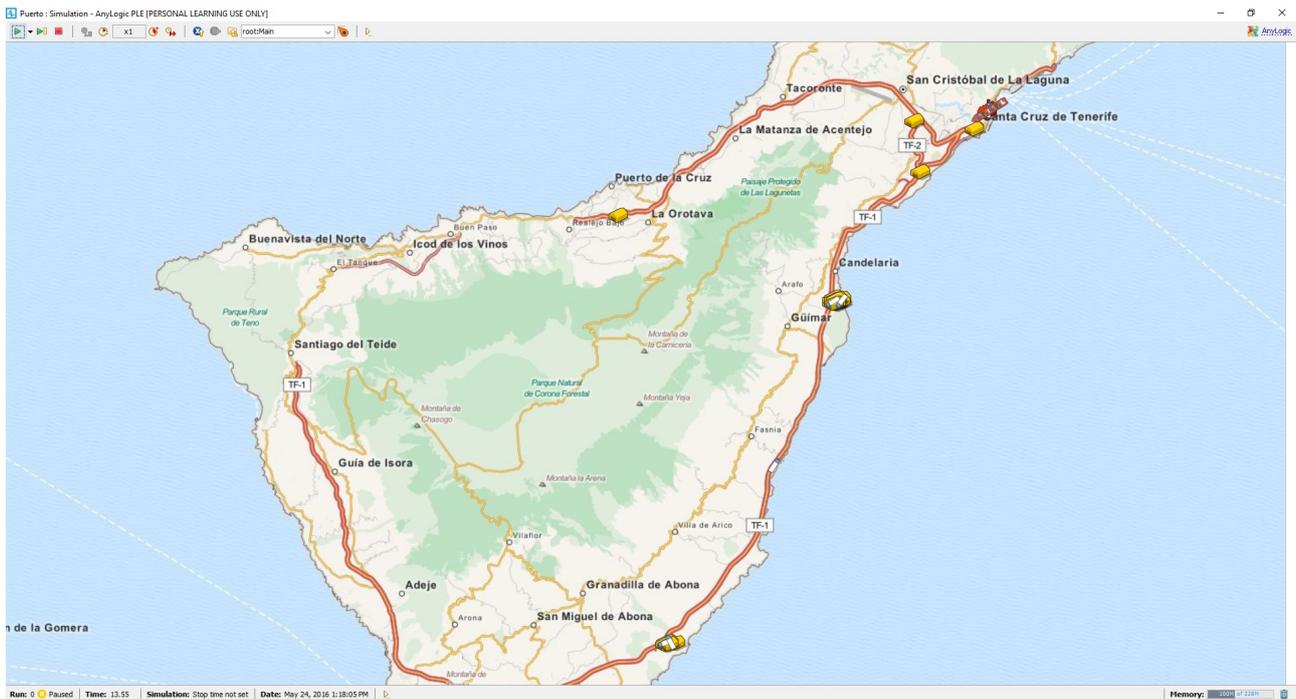


Figura 5.1: Vista gráfica de la ejecución

Se puede apreciar el atraque de los barcos en la terminal al igual que se ve a los camiones descargando sus mercancías en los distintos clientes, representados mediante los almacenes amarillos. También se pueden ver algún camión circulando por la autopista. Pero aunque el modelo funcione correctamente éste puede no ser eficiente, pues se puede estar haciendo un mal uso de los recursos de los que dispone. Es por ésto por lo que es necesaria la optimización del mismo.

Una vez establecidos los parámetros de modelo, se debe pasar a la optimización. En este apartado hay ciertos criterios que debemos establecer para poder obtener unos valores favorables de los recursos de los que dispone la terminal.

A continuación aparecen detallados todos los apartados necesarios que se tienen que implementar para poder comenzar con la optimización de recursos.

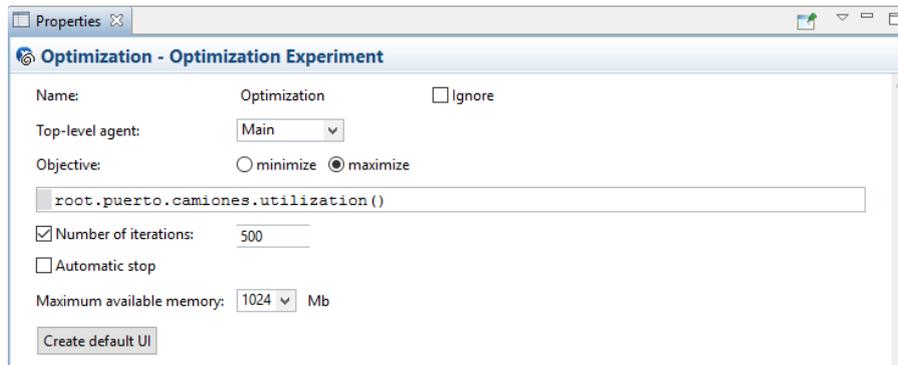


Figura 5.2: Objetivo principal a optimizar

Como se muestra en la Figura 5.2, en primer lugar se selecciona el objetivo principal de nuestra optimización, el cuál será de *maximizar*, y se indica que es lo que se quiere optimizar, en este caso, la utilización de los camiones de la terminal.

Parameters:						
Parameter	Type	Value				Suggested
		Min	Max	Step		
numeroCamiones	discrete	5	15	1		
numeroOperarios	discrete	4	14	1		
minContenedores	fixed	1				
maxContenedores	fixed	1000				

Figura 5.3: Parámetros de la optimización

Luego en el apartado de parámetros (Figura 5.3) dentro de las propiedades de optimización, se pueden seleccionar de los diferentes parámetros del proyecto cuáles son los que tienen que variar para lograr la optimización. En este caso esos parámetros son los camiones y los operarios de la terminal. Para cada uno de ellos se selecciona un rango dentro del cual oscilarán los valores posibles de los parámetros y también se selecciona como incrementaran ambos en cada iteración.

Para los camiones se ha seleccionado un valor mínimo de 5 camiones hasta un máximo de 15 e incrementarán de uno en uno. Para los operarios se ha seleccionado un mínimo de 5 operarios llegando hasta un máximo de 20 e incrementarán también de uno en uno. El resto de parámetros del modelo no es necesario modificarlos, por lo que se les deja con su valor natural.

Requirements			
Requirements (are tested after a simulation run to determine whether the solution is feasible):			
Enabled	Expression	Type	Bound
<input checked="" type="checkbox"/>	root.puerto.camiones.utilization()	<=	0.7
<input checked="" type="checkbox"/>	root.puerto.operarios.utilization()	<=	0.7

Figura 5.4: Requisitos a cumplir en la optimización

Por último en la pestaña de requerimientos (Figura 5.4) se añade que tanto la utilización de los camiones, como la utilización de los operarios no supere el 70 %. Con esto se consigue que los pedidos no estén parados en la terminal por culpa de que no disponemos de recursos, haciendo así que los clientes no tengan que esperar demasiado, a la misma vez que se logra mantener los recursos en un buen índice de trabajo.

Al ejecutar la optimización con todo configurado obtendremos los resultados mostrados en la Figura 5.5

Puerto : Optimization

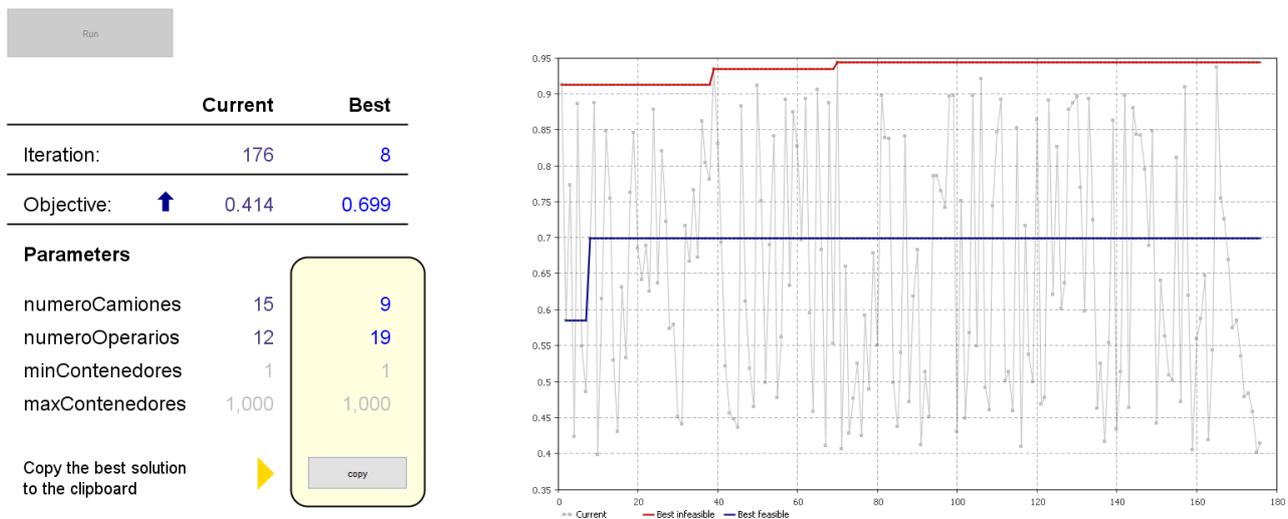


Figura 5.5: Resultados tras la optimización

El tiempo de ejecución de la optimización puede ser elevado dependiendo del ordenador donde se ejecute la misma, y del rango de valores dados a los parámetros. En este caso el tiempo aproximado empleado para completar la optimización fue de 17 minutos. Estos resultados han sido obtenidos en un PC con las especificaciones técnicas siguientes:

- *Microprocesador:* Intel i5-2300 a 2,80Ghz y 6MB Cache.
- *Ram:* Kingston DDR3 4GB 1333.

Los resultados muestran que para cumplir todos los requerimientos solicitados y mantener un trabajo eficiente, el número de camiones del que debe disponer la terminal es de un total de 9 y el número de operarios ha de ser de 19.

Una vez determinados los valores idóneos para los recursos de la terminal, estos son cambiados y se procede a la ejecución de la simulación con los valores ya corregidos. Mirando los porcentajes de utilización mostrados en la Figura 5.6 podemos comprobar que el porcentaje de trabajo de los recursos no supera el 70 %.



Figura 5.6: Porcentajes de utilización de los recursos de la terminal

Por lo tanto se verifica que los valores devueltos como *mejores* por la optimización cumplen los requerimientos solicitados. Obteniendo así unos resultados satisfactorios. No obstante, se puede cambiar la configuración de la optimización para tratar de conseguir otros resultados aceptables. Esto podría ser cambiando por ejemplo el porcentaje de trabajo de los operarios y de los camiones, elevando este porcentaje, para lograr un mayor uso de estos recursos. Esta modificación causaría una variación de los resultados obtenidos, los cuales serían válidos para el modelo, y se ajustarían a los nuevos requerimientos solicitados. Estos cambios se pueden realizar con cualquiera de los parámetros que conforman la optimización.

Capítulo 6

Conclusiones y líneas futuras

En este trabajo se ha abordado el problema del reparto de mercancías en terminales marítimas de contenedores, cuyo objetivo principal es optimizar los recursos empleados por la terminal para repartir la mercancía. Este problema tiene un alto impacto en los términos económicos que conciernen a las terminales portuarias. Para aportar una solución al problema se ha creado un modelo para su simulación y optimización.

En conclusión, se ha logrado alcanzar los diferentes objetivos propuestos para completar el proyecto. Para hacer esto posible, en primer lugar, se realizó un estudio del funcionamiento de las terminales marítimas portuarias así como de todos los elementos que conciernen a las mismas.

Una vez terminado el proceso de investigación, se pasó al aprendizaje de la herramienta de simulación con la que se desarrolló el proyecto, ésta es *AnyLogic*.

Después de un breve periodo de aproximación a la herramienta de simulación, se comenzó con la implementación de todas las partes necesarias para el correcto funcionamiento del sistema. Empezando por añadir un mapa georreferenciado sobre el que basar la simulación.

El siguiente paso fue diseñar los clientes, que serían los encargados de solicitar la mercancía a la terminal portuaria, así como la implementación de la propia terminal que recibirá estos pedidos.

Ya con los dos elementos principales incluidos en el modelo, se pasó a implementar el sistema de transporte de mercancías, tanto los camiones, para el transporte terrestre, como los barcos, para el marítimo.

Por último se implementó el patio de contenedores de la terminal, y con éste funcionando correctamente se pudo pasar a la optimización de los recursos de la terminal.

El siguiente pasos a seguir después de todo este trabajo realizado podría ser estudiar como afectaría a la terminal un modelo más detallado del funcionamiento del patio de contenedores, incluyendo tanto grúas como vehículos internos. También se podría abordar el problema de gestionar envíos de contenedores en barcos hacia otras terminales portuarias. Como último punto a valorar para futuros desarrollos, se podría desarrollar una interfaz que contenga todos los parámetros del proyecto, y que éstos puedan ser cambiados desde ahí, sin tener que entrar en el modelo para hacerlo.

Capítulo 7

Summary and Conclusions

This project tackles the goods' delivery issue in maritime container terminals. The main purpose of this problem is to optimize means used by the port terminal. This problem has a huge economic impact which concerns the port terminals. To give a solution to this issue a simulation and optimization prototype has been created. To sum it up, the different proposed objectives have been achieved to complete the project.

To make this possible, first of all, a study of the functioning of maritime terminals as well as all their components of these has been carried out. When the research process finished, it started the learning process of the simulation tool AnyLogic, with which the project has been developed. Afterwards a brief nearness period of time to this simulation tool, the implementation of the whole necessary parts to the right system running began. The next step was the desing of the customers, who are responsible for requesting goods to the port terminal, as well as the implementation of the terminal, which will receive the requests. Afterwards, it was implemented the goods delivering system, which consists of trucks for the ground transportation and ships for the sea transportation. Finally, the yard of the port terminal was implemented and the resources optimization phase was performed.

The next steps after all this work would include the study of the effect of a more detailed model of the operations of the container yard, including cranes and internal vehicles. It could also be addressed the problem of managing container shipments on boats to other port terminals. As a last point to value for future developments, it could be the development of an interface containing all the parameters of the project that allows modifying them without entering the the model.

Capítulo 8

Presupuesto

En la tabla 8.1 se muestra un presupuesto aproximado sobre lo que costaría el desarrollo de este proyecto. En la tabla no se incluye el software de simulación empleado, pues se ha utilizado la versión gratuita del mismo para el desarrollo. Por lo que para medir el presupuesto se ha hecho una estimación de las horas de trabajo empleadas y del ordenador utilizado para llevar a cabo el proyecto.

Concepto	Precio
Horas de trabajo	300 Horas x 15€ = 4500€
Equipo de trabajo	500€
TOTAL	5000€

Tabla 8.1: Presupuesto aproximado del proyecto

Capítulo 9

Bibliografía

- [1] *AnyLogic Help*: <http://www.anylogic.com/anylogic/help/>
- [2] *Puertos Tenerife*: <http://www.puertosdetenerife.org/index.php/es/>
- [3] *Banco Mundial de Datos*: <http://www.bancomundial.org/>
- [4] *Polígono Industrial de Güímar*: <http://polguimar.net/>
- [5] *Polígono Industrial de Granadilla de Abona*: <http://www.polgran.com/>
- [6] LÓPEZ ANSORENA, IÑIGO, *Criterios de calidad del servicio prestado en el ámbito de la línea de atraque de las terminales portuarias de contenedores* Tesis Doctoral, Madrid, 2013.
- [7] ALCEDO MOMOITIO, IÑAKI, *Las estrategias competitivas de los operadores de transporte marítimo de contenedores en línea regular*, Servicio Editorial de la Universidad del País Vasco/Euskal Herriko Unibertsitatearen Argitalpen Zerbitzua, País Vasco, 2014.
- [8] CASTELLANOS RAMÍREZ, ANDRÉS, *Manual de la gestión logística del transporte y distribución de mercancías*, Ediciones Uninorte, Barranquilla, Colombia, 2009.
- [9] RICARD, MAR; J. DE SOUZA, ADAMIR; MARTÍN, JUAN; DE RODRIGO, JAIME, *El transporte de contenedores: terminales, operatividad y casuística*, Edicions UPC, Edicions de la Universitat Politècnica de Catalunya, Barcelona, 2003.
- [10] RÚA COSTA, CARLES: *"Los puertos en el transporte marítimo"*, EOLI: Enginyeria d'Organització i Logística Industrial, Enero 2006.