



UNIVERSIDADE DA CORUÑA



Escola Politécnica Superior

**TRABAJO FIN DE MÁSTER
CURSO 2019/20**

*SIMULACIÓN Y PROGRAMACIÓN ESPACIAL EN
UN TALLER DE PREFABRICADO DE BLOQUES*

Máster Universitario en Ingeniería Industrial

ALUMNO

Iago Montes Flores

TUTOR

Diego Crespo Pereira

Marta Quiroga Pazos

FECHA

SEPTIEMBRE 2020

TÍTULO Y RESUMEN

Simulación y programación espacial en un taller de prefabricado de bloques

El contenido de este proyecto es el desarrollo e implementación de un algoritmo de programación de tareas espacial cuyo propósito es la optimización espacial y temporal de la ubicación de bloques en un taller de prefabricado, en la industria de la construcción naval.

Los bloques son subdivisiones del buque, de manera que previamente se construyen los bloques para su posterior ensamblaje y erección, formando así el navío.

El método de basa en la aplicación de una heurística, la cual es un conjunto de reglas de selección de ubicaciones, de selección de bloques, de actualización de las ubicaciones existentes y de derivación de nuevas ubicaciones a medida que se va produciendo la entrada de bloques al parque.

Para la verificación del método y la visualización de los resultados, se ha desarrollado un modelo de simulación que representa el proceso de ubicación de los bloques en el parque, en el cual se ha implementado el algoritmo.

Simulación e programación espacial nun taller de prefabricado de bloques

O contido deste proxecto é o desenvolvemento e implementación dun algoritmo de programación de tarefas espaciais cuxo propósito é a optimización espacial e temporal da localización de bloques nun taller de prefabricado, na industria da construción naval.

Os bloques son subdivisións do buque, de xeito que os bloques están previamente construídos para a súa posterior montaxe e erección, formando así o buque.

O método baséase na aplicación dunha heurística, que é un conxunto de regras para seleccionar localizacións, seleccionar bloques, actualizar as localizacións existentes e derivar novas localizacións a medida que se produce a entrada de bloques ao parque.

Para a verificación do método e a visualización dos resultados, desenvolveuse un modelo de simulación que representa o proceso de localización dos bloques no parque, no que se implementou o algoritmo.

Simulation and spatial scheduling of a block assembly workshop

This project carries out the development and implementation of a spatial scheduling algorithm in order to optimize the placement of blocks in the workshop spatially and temporally, within shipbuilding industry.

Blocks are subdivisions of the ship, which are previously built and then assembled and erected in order to form the ship.

The method consists in the application of a heuristic, which is a group of rules for selecting locations, selecting blocks, updating existing locations and creating new locations as blocks enter the work plate.

A simulation model has been developed in order to verify the method and visualize the results. The algorithm has been implemented in this model, which represents the process of placing the blocks in the work plate.

ÍNDICE GENERAL

DOCUMENTO I: MEMORIA

ANEXO I: DATOS



UNIVERSIDADE DA CORUÑA



Escola Politécnica Superior

**TRABAJO FIN DE MÁSTER
CURSO 2019/20**

*SIMULACIÓN Y PROGRAMACIÓN ESPACIAL EN
UN TALLER DE PREFABRICADO DE BLOQUES*

Máster Universitario en Ingeniería Industrial

Documento I

MEMORIA

Índice

1 Objeto	10
2 Alcance	11
3 Antecedentes	12
3.1 Introducción	12
3.2 Construcción Naval	13
3.2.1 La industria de la construcción naval	13
3.2.2 Reparto mundial de la industria naval	15
3.3 Evolución del sistema productivo en Navantia, Ferrol	19
3.4 Cuarta revolución industrial	22
3.4.1 Astillero 4.0	25
3.5 Programación de tareas	28
3.5.1 Programación de tareas en la empresa	29
3.5.2 Métodos heurísticos	31
3.5.3 Métodos metaheurísticos	33
3.5.4 Función objetivo	34
3.5.5 Programación de tareas espacial	37
3.6 Modelado y simulación	37
3.6.1 La utilidad de la simulación	38
3.6.2 Ventajas y desventajas de la simulación	40
3.6.3 Pasos del proceso de modelado y simulación	43
4 Estado del arte	45
5 Metodología	52
5.1 Descripción del problema y del proceso	52
5.1.1 Regla de localización	55
5.1.2 Regla de selección de bloques	56
5.1.3 Regla de actualización de áreas de localización	56
5.1.4 Regla de derivación de áreas de localización	57
5.1.5 Regla de derivación de áreas por salida de bloque	58
5.2 Modelo y simulación	58
5.2.1 Software de modelado y simulación, FlexSim	59
5.2.2 Modelo del parque de bloques	64
5.2.3 Ejecución del modelo	79
6 Experimentación	81
6.1 Optimización de la heurística	81
6.1.1 Resultados y análisis del primer experimento	82

6.2 Sensibilidad respecto a las dimensiones del parque	87
6.2.1 Resultados y análisis del segundo experimento.....	88
7 Conclusión	94
8 Trabajo futuro.....	96
9 Bibliografía	99
9.1 Construcción naval e industria 4.0	99
9.2 Programación de tareas y algoritmos heurísticos y metaheurísticos	99
9.3 Modelado y simulación	100
9.4 Estado del arte.....	100
9.5 FlexSim.....	101

Índice de tablas

Tabla 1. Entregas de buques nuevos por países / áreas en % anual (medido en GT) (Fuente: Lloyd's, Stopford, Clarkson y Wijnolst).	16
Tabla 2. Cifras de la industria naval en Europa 2017 – 2019 (Fuente: BRS Group). ...	18
Tabla 3. Cifras de la industria naval en Japón 2017 – 2019 (Fuente: BRS Group).	18
Tabla 4. Cifras de la industria naval en Corea del Sur 2017 – 2019 (Fuente: BRS Group).	18
Tabla 5. Cifras de la industria naval en China 2017 – 2019 (Fuente: BRS Group).	18
Tabla 6. Cifras de la industria naval en el resto del mundo 2017 – 2019 (Fuente: BRS Group).	19
Tabla 7. Escenarios diseñados para optimizar la heurística (Fuente: elaboración propia).	82
Tabla 8. Resultados obtenidos en el primer experimento (Fuente: elaboración propia).	82
Tabla 9. Escenarios diseñados para analizar la sensibilidad respecto a las dimensiones del parque (Fuente: elaboración propia).	87
Tabla 10. Resultados obtenidos en el primer experimento (Fuente: elaboración propia).	88

Índice de figuras

Figura 1. Comercio marítimo mundial, 2000 - 2019 (en miles de millones de toneladas-milla) (Fuente: UNCTAD).	17
Figura 2. Evolución de las embarcaciones construidas en Navantia (Fuente: A. Recamán Rivas, 2017).	22
Figura 3. Revoluciones industriales a lo largo de la historia (Fuente: Economipedia).	23
Figura 4. Fundamentos de la Industria 4.0 (Fuente: Boston Consulting Group).	25
Figura 5. Modernización del espacio del astillero de Ferrol (Fuente: A. Recamán Rivas, 2017).	28
Figura 6. Esquema de funciones del ERP (Fuente: Parada Visual).	30
Figura 7. Diagrama de los flujos de información en un sistema de producción (Fuente: Pinedo, 1995).	31
Figura 8. Representación conceptual de la simulación de procesos (Fuente: A. García del Valle).	39
Figura 9. Esquema del proceso de modelado y simulación (Fuente: A. García del Valle).	44
Figura 10. Layouts obtenidos con la aplicación del algoritmo del proyecto DAS (Fuente: K.J. Lee y J.K. Lee, 1996)	46
Figura 11. Proceso de ubicación de bloques con algoritmos de tipo bottom-left (Fuente: B. Kwon y G. M. Lee, 2015).	47
Figura 12. Esquema del proceso de ubicación de bloques (Fuente: J.G. Shin, O.H. Kwon, C. Ryu, 2008).	47
Figura 13. Esquema de actuación considerando las interferencias (Fuente: J.G. Shin, O.H. Kwon, C. Ryu, 2008).	48
Figura 14. Proceso de comprobación de interferencias (Fuente: J.G. Shin, O.H. Kwon, C. Ryu, 2008).	49
Figura 15. Comparación entre el método bottom-left fill y diagonal fill (Fuente: B. Kwon y G. M. Lee, 2015).	50
Figura 16. Ejemplo de estrategia de minimización del espacio inutilizable (Fuente: B. Kwon y G. M. Lee, 2015).	50
Figura 17. Representación tridimensional de la programación de bloques (Fuente: B. Kwon y G. M. Lee, 2015).	51
Figura 18. Orientaciones posibles de los bloques en este algoritmo (Fuente: B. Kwon y G. M. Lee, 2015).	52
Figura 19. Variables geométricas del problema (Fuente: D. Crespo).	53
Figura 20. Clasificación de los puntos de localización (Fuente: D. Crespo).	54
Figura 21. Posición de asignación del bloque (Fuente: D. Crespo).	57
Figura 22. Localización derivada en el lado del bloque (Fuente: D. Crespo).	57
Figura 23. Localización derivada en el lado opuesto al bloque (Fuente: D. Crespo). ..	57
Figura 24. Localización derivada en el lado opuesto al bloque (Fuente: elaboración propia).	58

Figura 25. Modelo 3D por defecto del elemento Source en FlexSim (Fuente: Apuntes de simulación de sistemas logísticos).....	59
Figura 26. Modelo 3D por defecto del elemento Queue en FlexSim (Fuente: Apuntes de simulación de sistemas logísticos).....	60
Figura 27. Modelo 3D por defecto del elemento Processor en FlexSim (Fuente: Apuntes de simulación de sistemas logísticos).....	60
Figura 28. Ejemplo del modelo 3D por defecto del task executer Transporter en FlexSim (Fuente: elaboración propia).....	61
Figura 29. Modelo 3D por defecto del elemento Sink en FlexSim (Fuente: Apuntes de simulación de sistemas logísticos).	62
Figura 30. Ejemplo de Process Flow (Fuente: elaboración propia).....	63
Figura 31. Ejemplo de representación de los datos de salida de la simulación (Fuente: Apuntes de simulación de sistemas logísticos).....	63
Figura 32. Modelo del parque de almacenamiento de bloques (Fuente: elaboración propia).....	64
Figura 33. Diagrama de flujo para la generación de los bloques en el modelo (Fuente: elaboración propia).....	65
Figura 34. Configuración actividad "Tiempo Inicio Disponibilidad" (Fuente: elaboración propia).....	65
Figura 35. Configuración de la actividad "Crear Bloque" (Fuente: elaboración propia).	66
Figura 36. Configuración de la actividad "Crear Bloque" (Fuente: elaboración propia).	67
Figura 37. Process flow que modela el comportamiento del almacén según la heurística diseñada (Fuente: Elaboración propia).	68
Figura 38. Configuración de la actividad "Asignar localización y tamaño" (Fuente: elaboración propia).....	69
Figura 39. Configuración de la actividad "Pull from List: Areas disponibles" (Fuente: elaboración propia).....	70
Figura 40. Configuración de la actividad "Posicionar" para la arista inferior (Fuente: elaboración propia).....	72
Figura 41. Configuración de la actividad "Posicionar" para la arista superior (Fuente: elaboración propia).....	72
Figura 42. Estado del parque en t = 200 h (Fuente: elaboración propia).	79
Figura 43. Estado del parque en t = 1100 h (Fuente: elaboración propia).	79
Figura 44. Estado del parque en t = 7000 h (Fuente: elaboración propia).	80
Figura 45. Estado del parque en t = 15000 h (Fuente: elaboración propia).	80
Figura 46. Estado del parque en t = 24800 h (Fuente: elaboración propia).	80
Figura 47. Evolución de la ocupación del parque del escenario 1 (Fuente: elaboración propia).....	83
Figura 48. Evolución de la ocupación del parque del escenario 2 (Fuente: elaboración propia).....	84

Figura 49. Evolución de la ocupación del parque del escenario 3 (Fuente: elaboración propia).....	84
Figura 50. Evolución de la ocupación del parque del escenario 4 (Fuente: elaboración propia).....	85
Figura 51. Evolución de la ocupación del parque del escenario 5 (Fuente: elaboración propia).....	85
Figura 52. Evolución de la ocupación del parque del escenario 6 (Fuente: elaboración propia).....	86
Figura 53. Evolución de la ocupación del parque del escenario 7 (Fuente: elaboración propia).....	86
Figura 54. Evolución de la ocupación del parque del escenario 8 (Fuente: elaboración propia).....	89
Figura 55. Evolución de la ocupación del parque del escenario 9 (Fuente: elaboración propia).....	89
Figura 56. Evolución de la ocupación del parque del escenario 10 (Fuente: elaboración propia).....	90
Figura 57. Evolución de la ocupación del parque del escenario 11 (Fuente: elaboración propia).....	90
Figura 58. Evolución de la ocupación del parque del escenario 12 (Fuente: elaboración propia).....	91
Figura 59. Evolución de la ocupación del parque del escenario 13 (Fuente: elaboración propia).....	91
Figura 60. Evolución de la ocupación del parque del escenario 14 (Fuente: elaboración propia).....	92
Figura 61. Evolución de la ocupación del parque del escenario 15 (Fuente: elaboración propia).....	92
Figura 62. Ejemplo de modelo del astillero. (Fuente: Unidad Mixta de Investigación, UMI).....	96
Figura 63. Ejemplo de modelo del astillero. (Fuente: Unidad Mixta de Investigación, UMI).....	97
Figura 64. Flujo de información entre el modelo virtual y el sistema real (Fuente: Siemens).....	97

1 OBJETO

El objetivo de este proyecto es el desarrollo de un algoritmo de programación de tareas espaciales para la optimización del *makespan* en un taller de prefabricación de bloques de construcción naval, teniendo en cuenta tanto restricciones temporales como geométricas. En adición a este objetivo, se implementará este algoritmo en un modelo de simulación para la evaluación de la programación obtenida, además de permitir la visualización del proceso de ubicación de bloques en el parque de almacenamiento.

2 ALCANCE

El alcance del presente proyecto es desarrollar un sistema basado en simulación que permita optimizar la programación espacial y temporal de la disposición de bloques de dimensiones heterogéneas en un taller de prefabricado de un astillero.

En primer lugar, se creará un algoritmo que permita la correcta secuenciación en la colocación de los bloques en el astillero, el cual tendrá en cuenta las restricciones correspondientes tanto para la ocupación del espacio como para el cumplimiento de los plazos de entrega.

Tras el proceso anterior, dicho algoritmo se implementará en un *software* de simulación, siendo FlexSim en este caso, en el que podremos realizar pruebas para la verificación del funcionamiento del algoritmo y una visualización del mismo. En este modelo de simulación de eventos discretos se representará la demanda de bloques y los procesos de entrada y salida del área de trabajo.

3 ANTECEDENTES

3.1 Introducción

La construcción de buques se enmarca dentro de las industrias pesadas que normalmente operan siguiendo una estrategia de construcción bajo pedido. En esta, se producen grandes estructuras caracterizadas por configuraciones complejas, sistemas de avanzada tecnología y con elevados tiempos de fabricación. Debido al gran tamaño que supone el cuerpo de la nave, esta se divide en un gran número de unidades llamadas bloques, contando con diversas formas y pesos. Entonces, en la construcción de buques de mediana a gran escala, la estrategia de construcción empleada es la construcción por bloques, la cual consiste en el ensamblaje en grada de las mencionadas divisiones prefabricadas del barco, de forma que existe un espacio en el astillero destinado principalmente al montaje de dichos elementos. Estos bloques se someten a una serie de operaciones, que incluyen el acondicionamiento previo, la pintura, el ensamblaje, la posible instalación de sistemas del buque, etc., como tareas a realizar antes de la erección de los mismos. Por tanto, la fabricación de estos buques se puede enmarcar en dos fases generales, siendo la primera la producción de los bloques fabricados por la misma empresa o por subcontrata; y la segunda consistiendo en el ensamblaje de los mismos.

Las dos etapas mencionadas anteriormente requieren de una integración de diversos sistemas y recursos de gran complejidad, que puede dificultarse aún más por la necesidad de coordinar la gestión tanto de recursos propios como ajenos, ya que en el caso de realizarse una subcontrata es frecuente la cesión de parte de las propias instalaciones a estas. Por este motivo, agregando además el gran capital que se invierte en este tipo de proyectos y su prolongado tiempo de consecución, la tarea de planificación de recursos humanos, equipos y materiales es un problema crítico que deberá ser tratado adecuadamente.

Para favorecer la reducción del tiempo de construcción del navío y la mejora de las condiciones de trabajo, es posible realizar el montaje de los diversos sistemas del buque durante el proceso de construcción de los bloques frente a realizar su instalación en el buque una vez se ha completado el ensamblaje de su estructura. Esta primera opción tiene la contrapartida de provocar un incremento del esfuerzo necesario en la planificación e ingeniería para evitar que durante el proceso de montaje de los bloques se deban realizar retrabajos para lograr un correcto ajuste, lo que suele conllevar retrasos en el cronograma de producción general, que a su vez desemboca en operaciones menos seguras y en resultados de peor calidad por apresurarse en dichas operaciones al intentar compensar los retrasos. Además, una planificación deficiente puede ocasionar fluctuaciones significativas de la carga de trabajo, traduciéndose en la aparición de operarios sin tareas asignadas o en operarios con una ocupación excesiva.

Para resolver el problema que supone la planificación, no es suficiente con el empleo de un *software* de planificación de proyectos, ya que la gran complejidad de las interrelaciones entre los diferentes componentes del proyecto sumadas a la importancia de las restricciones espaciales de los recursos hacen que esta planificación sea difícilmente abarcable por el software comercial existente.

Con el fin de emplear una disposición espacial dinámica de los recursos que permita realizar una correcta secuenciación de las actividades de montaje en la grada, es indispensable el uso de métodos de planificación espacial, de forma que se pueda minimizar el tiempo de erección de los bloques a la vez que se tienen en cuenta las restricciones espaciales pertinentes.

Por otro lado, en la fase de construcción de los bloques aparece un problema de programación de mayor importancia, en la que se deben considerar, además de las limitaciones de capacidad de los parques de almacenamiento donde se lleva a cabo el montaje, las relaciones de precedencia que el proceso de erección de bloques impone, debidas principalmente a criterios de estabilidad. Todos estos aspectos motivan la formulación de un problema de programación de tareas espacial complejo, el cual se resolverá con la aplicación de un algoritmo de planificación que consiste en un conjunto de reglas de despacho que aseguren el cumplimiento con las limitaciones espaciales y de capacidad de los recursos necesarios y minimicen el tiempo de producción.

Una vez creado el algoritmo con las reglas de despacho, para verificar la optimización de los tiempos de producción con las soluciones obtenidas mediante los algoritmos de planificación, una opción es emplear como herramienta la simulación de eventos discretos, lo que conlleva a la creación de un modelo digital del astillero o de la parte de este que se pretenda estudiar.

3.2 Construcción Naval

3.2.1 La industria de la construcción naval

Desde la antigüedad, la navegación ha formado parte de la humanidad siendo utilizada como medio de transporte de personas, de mercancías o bien como una fuerza militar. Esto, ha permitido el desarrollo de varias civilizaciones, favoreciendo el comercio e intercambio de bienes de forma global. Por ello, muchos de los grandes núcleos habitados surgieron en las orillas de océanos, mares, lagos y ríos importantes, gracias al gran abanico de posibilidades que ofrecía este medio.

Uno de los aspectos más destacables de la industria de la construcción naval es la naturaleza global de su mercado, por lo que surge una alta competencia a nivel internacional, motivando así la necesidad a las diferentes entidades navieras de estar a la orden del día en cuanto a aspectos técnicos y productivos. Frente a la industria de otros medios de transporte como la aviación, el ferrocarril o el transporte por carretera en los que la fabricación en serie está completamente implantada, la industria de la construcción naval no cuenta con este planteamiento, y en caso de acercarse a él, lo hace con series extremadamente cortas. Generalmente, a grandes dimensiones, no se produce un modelo para posteriormente introducirlo en el mercado, sino que en los astilleros se trabaja contra pedido produciendo buques singulares con un gran valor unitario bajo el contrato entre el astillero que lo va a construir y un armador, que es quien realiza el encargo del mismo.

El proceso completo para la construcción de un buque tiene un prolongado desarrollo en el tiempo, debido a las grandes dimensiones de este, a la complejidad de los diversos sistemas con los que cuenta, al diseño y planificación del barco y el proyecto, etc. En consecuencia, se trata de proyectos que cuentan con altísimas inversiones, de manera que el coste de la construcción es una cifra clave para el

astillero, quien con una mala gestión, planificación o actuación podría ocasionar pérdidas millonarias.

Desde la idea que esta industria planteaba en sus inicios hacia una mano de obra intensiva hasta la actual de capital y tecnología intensivos, la construcción naval ha atravesado una gran transformación hacia una industria de síntesis en la que la entidad que proporciona el navío al armador aporta entre el 20% y el 30% del valor del producto final, aunque su responsabilidad finalista responda de la totalidad. Estos porcentajes dependen del tipo de embarcación, ya que con sus diferentes finalidades su construcción será diferente y también lo serán los equipos y sistemas con los que cuente.

Con todo lo anterior, los departamentos de diseño e ingeniería cobran una vital importancia en el astillero, en los que se realiza una labor de unificación de equipos y componentes de características y tecnologías diversas aportados por diferentes suministradores. Entre los sectores participes en la producción de un navío podemos destacar los siguientes: la industria siderúrgica férrea y no férrea, la eléctrica y electrónica, la de equipos y maquinaria no eléctrica, la química, la maderera, la informática, entre otros.

A mayor complejidad y valor del navío, más crítico será la labor del astillero constructor en las ingenierías del proceso y del producto, dada su responsabilidad finalista en la entrega en cuanto al precio y al plazo acordado con el cliente. Por ello, la industria naviera se relaciona técnica y económicamente por medio de los suministros con un amplio abanico de importantes sectores industriales y de servicios. Debido a esto, las políticas industriales de diversas naciones se apoyaron en el sector de la construcción naval como elemento tractor del desarrollo industrial y de servicios, sobre todo cuando más ligado estaba a un gran empleo de mano de obra, y especialmente en países en los que la creación de empleo industrial era una política básica. Este escenario ha ido mutando con la progresiva aparición de productos más desarrollados y que requerían mayores inversiones para poder mantener la competitividad. De esta forma, el factor de la mano de obra ha disminuido proporcionalmente ya que la aplicación de tecnologías de última generación en todas las fases del proyecto, ya sea en la planificación, el diseño, la fabricación, etc. ha tomado un papel fundamental a la hora de ofrecer soluciones competitivas.

La imagen que se tenía años atrás sobre este tipo de industria, en tanto que se consideraba que era más propia de países aún en desarrollo y que se iba abandonando a medida que el sector servicios aumentaba su relevancia en el PIB, dista mucho de la realidad. Dicha industria naviera está ligada fuertemente a la demanda de transporte, cuyo comportamiento sigue los movimientos de los crecimientos económicos y los flujos de comercio tanto a nivel global como regional, en un mundo donde el mercado se encuentra cada vez más globalizado y en el que factor del transporte toma un papel muy relevante.

Por todas las características ya dadas, la manera en la que las entidades constructoras afrontan la predicción de la evolución del mercado para definir sus estrategias, tanto técnicas como comerciales, constituye un elemento muy importante para la planificación en el astillero. Es posible intentar realizar una estimación de la demanda de flota para años venideros, aunque en una industria a gran escala como esta no siempre se va a asemejar a la demanda real, ya que existen diversos factores que puedan afectar a la divergencia entre el escenario real y el ficticio. Situaciones como la aparición de crisis de manera repentina en los crecimientos económicos, variaciones de las rutas establecidas por causas ajenas a la navegación, crecimientos en las capacidades portuarias que divergen de los esperados, movimientos políticos, etc., pueden ocasionar la discordancia entre la demandada de flota que los técnicos de las organizaciones constructoras habían estimado y la demanda real que finalmente se

produce. Por razones como esta, mejorar la flexibilidad y eficiencia de la producción de estas empresas navales es vital.

Un ejemplo de situaciones como esta fue lo ocurrido tras la crisis de 2008. En 2007, la inversión total en nuevos contratos navieros se había incrementado hasta 263.100 millones USD. Sin embargo, con la crisis de 2008, esta cifra descendió a 166.000 USD en su primer año, y ya en 2009 llegó a su fondo con 30.000 millones USD (Pérez García, 2018), siendo este el peor año para el crecimiento económico mundial. Esa cifra se mantuvo más o menos estable con los años, sufriendo un leve crecimiento en los últimos años hasta la crisis actual ocasionada por el Covid-19.

De producirse un abultamiento significativo en las carteras de pedidos de las empresas, este retrasaría la entrega de un gran número de buques, por lo que algunas entidades constructoras están comenzando a aumentar su capacidad de producción. Por otro lado, también aparecen nuevas corporaciones que tratan de introducirse en este sector, buscando la producción del tipo de navíos que menores barreras de entrada presente, tanto a nivel técnico como constructivo, pudiendo producirse así un desequilibrio en el mercado. Una falta de experiencia puede derivar en retrasos en la fase de ejecución de las obras, causando así importantes desfases en los pagos del armador y rompiendo así la estructura piramidal financiera de la que dependía el constructor, basada en una continuidad en la adjudicación de nuevos contratos.

Por la consideración que tienen muchos países hacia esta industria de ser un factor estratégico y a pesar de que astilleros menores de propiedad privada o semiprivada desaparezcan en épocas en las que el crecimiento económico se estanque, las grandes corporaciones de construcción naval, cuya propiedad acostumbra ser del estado, no sufren estas consecuencias ya que se les otorgan cuantiosos pedidos producidos por compañías estatales, independientemente de la situación global del equilibrio oferta-demanda en el sector naval. El carácter estratégico surge de la intención de mantener una soberanía tecnológica sobre un factor vital para cada uno de los países, asegurándose el aprovisionamiento de recursos vitales y estratégicos que dependen del transporte marítimo. Por tanto, no se trata de que los armadores del país posean una gran flota, sino que en dicho país exista un desarrollo tecnológico y una capacidad de producir buques que puedan asegurar los suministros en épocas de incertidumbre que puedan aparecer de manera inesperada.

Para garantizar lo anterior, parece lógico que el objetivo de un país en ese aspecto sea mantener una industria naval y una flota con un desarrollo tecnológico que les permita competir en situaciones normales en el ámbito internacional. Sin embargo, este comportamiento no se ha dado nunca en la historia de la construcción naval, en la que se han impuesto proteccionismos, ayudas y subsidios que atenuaron las condiciones que se dan en cualquier otro mercado abierto, las cuales fomentan la competitividad de las empresas.

3.2.2 Reparto mundial de la industria naval

Resulta interesante realizar un análisis de la participación de distintas áreas geográficas o países en este sector, que como ya vimos, está muy ligado al desarrollo económico de los mismos, sobre todo en el segmento del transporte marítimo. En la siguiente tabla se representa la evolución de dicha participación en porcentaje de diferentes áreas, medida en toneladas de registro bruto GT, desde 1925 hasta 2016. En esta se manifiesta como Europa, comenzando con poseer prácticamente la

totalidad de la producción, ha ido quedando en segundo plano a medida que países orientales se implicaban en este sector.

Tabla 1. Entregas de buques nuevos por países / áreas en % anual (medido en GT) (Fuente: Lloyd's, Stopford, Clarkson y Wijjolst).

Año	Europa	Japón	Corea del Sur	China	EE.UU.	Otros
1925	90	3	0	0	5	2
1930	83	5	0	0	10	2
1935	86	11	0	0	2	1
1940	82	8	0	0	6	4
1945	15	0	0	0	80	5
1950	75	11	0	0	12	2
1955	85	12	0	0	1	2
1960	67	20	0	0	5	8
1965	55	42	0	0	2	1
1970	45	47	0	0	3	5
1975	42	48	2	0,5	2	5,5
1980	23	47	4	0,5	4,5	21
1985	17	52	17	2	0	12
1990	20	40	22	3	0	15
1995	18	40	27	5	0	10
2000	15	32	36	7	2	8
2005	12	26	43	8	2	9
2010	5	21	35	38	0,2	0,8
2015	2	19	36	37	0,5	5,5
2016	4	20	38	33,3	0,45	4,25

Como se observa en la tabla, la dominación europea en esta industria dura hasta 1940, coincidiendo con el inicio de la Segunda Guerra Mundial, momento en el que Estados Unidos irrumpió en este sector pasando a representar aproximadamente un 80% de las entregas de buques para transporte durante la duración del conflicto (no se contabilizan barcos con fines bélicos en esta tabla). Estos navíos eran responsables del flujo de recursos en el Pacífico y en el Atlántico, dada la extensión geográfica del conflicto.

Ya en la década de los años cincuenta, la industria naviera europea comenzó a recuperar cuota de mercado, mientras Japón aumentaba cada vez más su capacidad para competir en el sector, hasta sobrepasar a la europea, la cual entro en declive en la década de los 80 coincidiendo con el auge de la industria surcoreana. En el caso de china, gracias a las políticas industriales que adoptaron comenzado la segunda década de este siglo, han conseguido rápidamente llegar al liderazgo de la producción en el segmento del transporte.

Desde la Segunda Guerra Mundial, a medida que el mundo ha ido globalizando su economía con ascenso progresivo de esta, la demanda de buques a muy largo plazo también se ha mantenido ascendente. No es así para la demanda a corto plazo, la cual puede presentar oscilaciones significativas, como ha ocurrido con la crisis

financiera vivida recientemente, siendo las circunstancias de este tipo difícilmente predecibles desde la óptica de la industria naviera.

La diversidad de finalidades que tienen los productos que nacen en la construcción naval se traduce en que no sea adecuado hacer un tratamiento homogéneo en cuanto a los aspectos económicos y los factores que condicionan a cada una de las vertientes. De esta forma, la producción de buques para el transporte de productos energéticos como el petróleo y sus derivados, gases licuados, carbón, etc., responde a las demandas y precios de las distintas energías, de igual manera que condiciona a los buques, artefactos y estructuras relacionados con la explotación de los fondos marinos y los de producción de energía, como pueden ser las plataformas de energía eólica *off-shore*. Respecto a la construcción de buques para gráneles sólidos, ya sea el carbón que ya hemos citado o minerales como el hierro, su demanda dependerá del crecimiento de industrias que empleen dichos recursos. En cuanto a los productos elaborados o semielaborados, estos son transportados principalmente en buques portacontenedores o buques para cargas rodadas, y su demanda es cada vez mayor de acuerdo con la globalización del comercio que acontece actualmente. Cabe destacar que esta rama del transporte marítimo ha condicionado mucho la ampliación de las capacidades de los astilleros. Debido a los beneficios del efecto de escala y al aumento de los combustibles, el cual supuso una reducción de la velocidad a la que se desplazaban los buques para lograr un menor consumo por milla navegada, el tamaño promedio de estos navíos portacontenedores sufrió un importante aumento. Por ello, diversos astilleros tuvieron que tomar medidas al respecto por la limitación de sus capacidades físicas, ya sea con ampliaciones, una mejor planificación, etc, además de afectar al método de construcción de los buques, como se verá posteriormente.

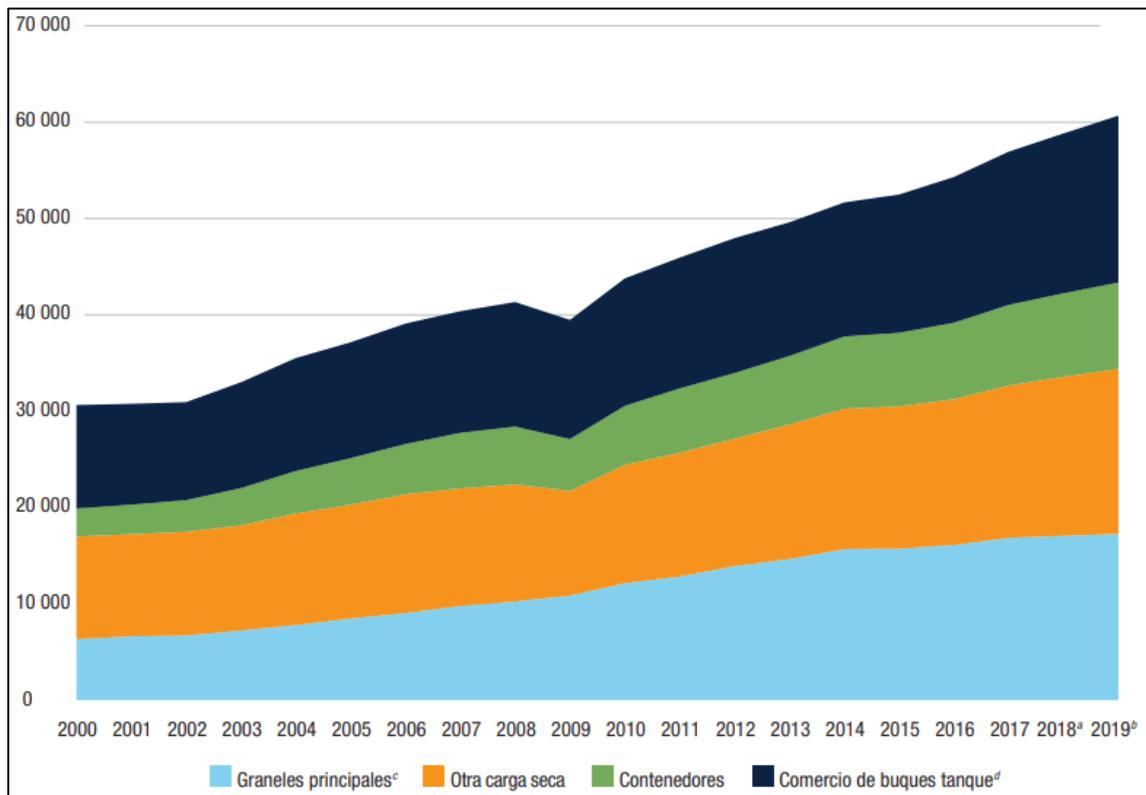


Figura 1. Comercio marítimo mundial, 2000 - 2019 (en miles de millones de toneladas-milla) (Fuente: UNCTAD).

Dadas las condiciones cíclicas de industrias como la construcción naval mercante o la que se dedica a la explotación de recursos marinos, en tanto que presenta altibajos respecto a su oferta y la demanda, los países cuyo sector naviero no se ha adaptado a las nuevas condiciones van perdiendo relevancia dentro del mismo, dejando hueco para los que sí han realizado un esfuerzo en mejorar sus capacidades, como en avances tecnológicos, en mejoras continuas en la ingeniería de procesos, en la productividad y sostenibilidad, etc.

Ya en los últimos años, las cifras de producción naval de EE.UU continuarían descendiendo frente al crecimiento en este sector de otros países como Filipinas, Vietnam, etc. La situación de los países y áreas productoras principales es la reflejada en las siguientes tablas:

Tabla 2. Cifras de la industria naval en Europa 2017 – 2019 (Fuente: BRS Group).

Europa	2017		2018		2019	
	mill. TPM	Buques	mill. TPM	Buques	mill. TPM	Buques
Cuota de mercado	1.60%	237	7.70%	288	8.70%	285
Cartera de pedidos	3.4	237	12.1	288	12.6	285
Contratos	1.1	77	2.8	103	2.9	80
Entregas	1.7	89	1.9	55	2.4	83

Tabla 3. Cifras de la industria naval en Japón 2017 – 2019 (Fuente: BRS Group).

Japón	2017		2018		2019	
	mill. TPM	Buques	mill. TPM	Buques	mill. TPM	Buques
Cuota de mercado	25.10%	750	24.30%	741	21.90%	625
Cartera de pedidos	54.0	750	54.9	741	44.1	625
Contratos	11.2	167	22.1	308	13.7	239
Entregas	20.2	337	20.1	313	24.5	355

Tabla 4. Cifras de la industria naval en Corea del Sur 2017 – 2019 (Fuente: BRS Group).

Corea del Sur	2017		2018		2019	
	mill. TPM	Buques	mill. TPM	Buques	mill. TPM	Buques
Cuota de mercado	24.40%	396	27.80%	460	28.10%	483
Cartera de pedidos	52.4	396	62.7	460	56.6	483
Contratos	32.0	193	30.4	254	26.3	243
Entregas	30.8	272	19.0	178	32.4	220

Tabla 5. Cifras de la industria naval en China 2017 – 2019 (Fuente: BRS Group).

China	2017		2018		2019	
	mill. TPM	Buques	mill. TPM	Buques	mill. TPM	Buques
Cuota de mercado	43.70%	1234	43.00%	1309	45.40%	1206
Cartera de pedidos	93.9	1234	97.1	1309	91.4	1206
Contratos	34.0	441	38.7	555	30.5	421
Entregas	38.5	528	34.7	510	36.2	524

Tabla 6. Cifras de la industria naval en el resto del mundo 2017 – 2019 (Fuente: BRS Group).

Otros	2017		2018		2019	
	mill. TPM	Buques	mill. TPM	Buques	mill. TPM	Buques
Cuota de mercado	5,20%	267	3,30%	226	2,60%	187
Cartera de pedidos	11,3	267	74	226	5,2	187
Contratos	4,8	122	1,7	52	1,3	45
Entregas	4,9	87	4,6	95	3,5	84

De lo expuesto anteriormente se manifiesta lo poco repartida que esta la producción naval en el mundo, siendo tres los países que abarcan la mayoría de esta producción, Japón, China y Corea. Estos países asiáticos supieron adaptarse a los requisitos actuales y tomaron políticas industriales específicas para el crecimiento y mantenimiento de una industria fuerte. Por la contra, los países europeos no supieron adoptar las medidas adecuadas, creyendo que era necesario mantener una competitividad a nivel global, a pesar de que los armadores europeos eran libres de establecer un contrato con empresas navales de cualquier parte del mundo con ayudas de estado europeas. Para evitar esta situación, lo idóneo sería alcanzar una situación puntera en el mercado internacional en todos los aspectos, ya sea en el diseño o la fabricación como en la explotación. Es importante tener presente que la producción naval de defensa es un componente importante desde el punto de vista estratégico, produciéndose un transferencia de tecnologías tanto del propio barco como de la producción entre ambos sectores, que en una gran parte de los casos comparte las instalaciones de fabricación.

Pese a la situación que se ha presentado, Europa tiene competencias suficientes para reimpulsar su industria naval, por ser una sociedad comprometida con el desarrollo económico y social y con la calidad de vida y el bienestar. Esta situación podría ser alcanzada si establecen políticas adecuadas y se invierte y se aplican nuevas tecnologías como ya se hace en industrias de otros sectores, considerando aspectos como la protección del medio ambiente, el empleo de energías renovables y menos contaminantes, la aplicación de la inteligencia artificial, nuevas técnicas de fabricación, el uso de tecnologías tipo *block-chain*, etc., en definitiva, dando paso a la adaptación de los astilleros a la industria 4.0, que se presentará posteriormente.

3.3 Evolución del sistema productivo en Navantia, Ferrol

Ferrol cuenta desde hace más de dos siglos con unas instalaciones dedicadas a la construcción naval, en las que actualmente opera Navantia. Durante todo el periodo de su existencia la metodología empleada en su proceso productivo ha ido sufriendo distintas transformaciones, de acuerdo con nuevas técnicas y tecnologías que surgieron a lo largo de los años. A continuación, se expondrá una revisión de esta evolución en los últimos años, desde mediados del siglo XX hasta la actualidad, periodo en el cual el avance tecnológico ha ido acelerándose cada vez más.

Actualmente, las fases de fabricación de un buque se siguen ajustando a las tradicionales. Sin embargo, los aspectos que sí ha experimentado un gran cambio han

sido la complejidad de sus estructuras, los materiales empleados, los componentes y sistemas del buque, las técnicas aplicadas en su construcción y, en definitiva, todo el proceso constructivo del navío, ya sea con finalidades militares o de transporte.

El proceso productivo de un buque abarca desde que el armador realiza la contratación para la construcción de un buque hasta que este es entregado al mismo. Durante todo este proceso se debe cumplir con una planificación para satisfacer los requisitos del cliente, tanto en el producto final como en precio y plazo, siendo responsable la entidad constructora. Por ello, a la vez que las necesidades de los clientes han ido avanzando, también lo ha hecho conjuntamente el proceso productivo.

En los inicios de la década de los años cincuenta, la Empresa Nacional Bazán en Ferrol operaba bajo las directrices de producción designadas por los encargos del Gobierno con el objetivo de renovar la flota de defensa. Por otro lado, se continuaba con la construcción de varios navíos militares de un proyecto iniciado a finales de los años cuarenta, en el que surgieron algunos contratiempos que ralentizaron la producción de los mismos. La producción de barcos con fines bélicos no se extendió mucho más en estos años, ya que la actividad de los astilleros se dirigió hacia la construcción de petroleros y cargueros desde 1953, debido a la firma de un acuerdo de cooperación con Estados Unidos y las necesidades energéticas de España. Durante la década de los sesenta se mantuvo predominante la construcción de buques con dichas finalidades, destacando este astillero por ser responsables de la construcción del mayor petrolero en España hasta ese momento.

En este periodo, el método de construcción empleado seguía las líneas más tradicionales, siendo un sistema de construcción gradual, por etapas y en la propia grada, de forma que el buque crecía en altura y amplitud progresivamente en dichas gradas siguiendo las fases de construcción, condicionadas por el ritmo de trabajo de las diferentes áreas del astillero y por la llegada de los diferentes componentes de barco desde los talleres. Aun siguiendo una metodología tradicional, se contaba con la incorporación de algunas nuevas tecnologías de la época, siendo el primer astillero del país en emplear un sistema de prefabricación y montaje de divisiones del buque ya soldadas eléctricamente, proceso que se iría desarrollando con los años en los astilleros de España, influenciados por las innovaciones que introducía Japón en su industria naval.

El planteamiento del proyecto dependía del tipo de buque que demandaba el cliente, que en este caso era el Estado, ya sea por el Ministerio de Marina para renovar la flota o por la Empresa Nacional Elcano para el transporte de petróleo. Los navíos que tenían una finalidad bélica presentaban una mayor complejidad en cuanto a su construcción, incorporando los sistemas que requería el diferente equipamiento militar. Por otro lado, los buques destinados al transporte marítimo, pese a plantear menos dificultades técnicas, demandaban igualmente un trabajo de calidad para conseguir una estanquidad y estabilidad adecuadas a la carga que portarían.

Ya en esta época la oficina de proyectos constituía el pilar central en la construcción de los barcos, comenzando en sección técnica el proceso productivo con el diseño del proyecto. Completado esto, el departamento de delineación ejecutaba los planos necesarios para que la sección de trabajos realizase una planificación de las tareas que serán necesarias para la ejecución del barco. A partir de los planos, se comenzaba el proceso de la propia construcción con la solicitud del material necesario para ello. Este material llegaba a los talleres de fundición, en los que preparaban el acero para iniciar la construcción del casco. Tras esto, las diferentes áreas de trabajo del astillero ejecutaban las tareas pertinentes. Cada una de estas áreas contaba con un responsable, quienes se hacían cargo de la organización de las tareas que le correspondía a su sección y de su correcta ejecución por los trabajadores, para lo que se comenzó con la implementación de algunos controles de tiempo y calidad.

Finalizado el ensamblaje de todos los elementos preconstruidos, el montaje de algunos elementos y el pintado, el buque era botado. Ya a flote, se continuaba con la instalación de los motores, de los sistemas e instrumentos que el tipo de buque requería, de tuberías y equipos eléctricos y de los distintos elementos para la habitabilidad del navío. Previamente a la entrega al cliente, se realizaban una serie de pruebas para verificar el correcto funcionamiento de los sistemas instalados.

Desde 1964, el equipo directivo del astillero fue incorporando cambios en el sistema de fabricación, dada la mayor complejidad de los contratos y al avance de las técnicas de construcción naval. Esto impuso una adaptación de las infraestructuras del astillero y de la organización del trabajo para cumplir con los plazos en la producción de los nuevos buques, que como ya se ha mencionado se trataba principalmente de cargueros y petroleros, aunque también se produjo la renovación de algunos navíos militares. La producción comenzó a centrarse en la etapa de prearmamento, de forma que cada vez se realizaban más trabajos sobre los bloques, los cuales iban creciendo en tamaño y complejidad. Las mejoras en el método de producción requerían de una mejor organización del trabajo, adaptándola a las exigencias impuestas por los técnicos estadounidenses.

En los últimos años de la década de los sesenta se produjo una primera transformación importante en el sistema de producción causada tanto por la vía militar como por la de transporte marítimo, que para poder mantener una competitividad en ambos sectores, fue necesaria una importante renovación en aspectos técnicos de la producción. Por una parte, se estaba ideando desde el Gobierno un programa para actualizar los buques de defensa nacionales en colaboración con Estados Unidos; y por otra parte, tras el cierre del Canal de Suez se incrementaban la necesidad de nuevos buques para el transporte de petróleo.

De dicha colaboración con EEUU surgió una actualización y ampliación de las tres áreas del astillero para poder cumplir con las exigencias del acuerdo de transferencia tecnológica desde Estados Unidos para las fragatas españolas. Esto también supuso la necesidad de formar a los trabajadores de todas las áreas del astillero para adaptarse a las nuevas técnicas de trabajo y al control constructivo de los técnicos norteamericanos, que obligaría posteriormente a aplicar un control de tiempos y en la productividad más exhaustivo. Es destacable que, además de la renovación de los talleres, también se creó un nuevo taller para la producción de los bloques que posteriormente compondrían el barco, de acuerdo con el nuevo enfoque productivo en el que se pretendía aumentar las tareas que se harían sobre estos bloques y disminuir las que se ejecutan una vez se realiza su ensamblaje.

El método de construcción por bloques continuaría perfeccionándose entre los años setenta y ochenta, aplicándose en todas las áreas de la producción, resultando en unos bloques cada vez más completos y de mayor tamaño. Años más adelante, el método seguiría con su consolidación con la construcción de las fragatas F100 y el buque de asalto anfibio Rey Juan Carlos, el cual fue el mayor buque militar fabricado en España. Estos proyectos impulsaron el reconocimiento del astillero de Ferrol en la producción de buques altamente tecnificados, que facilitó la llegada de nuevos contratos importantes durante la primera década del siglo XXI como la producción de cinco fragatas para la marina de Noruega.

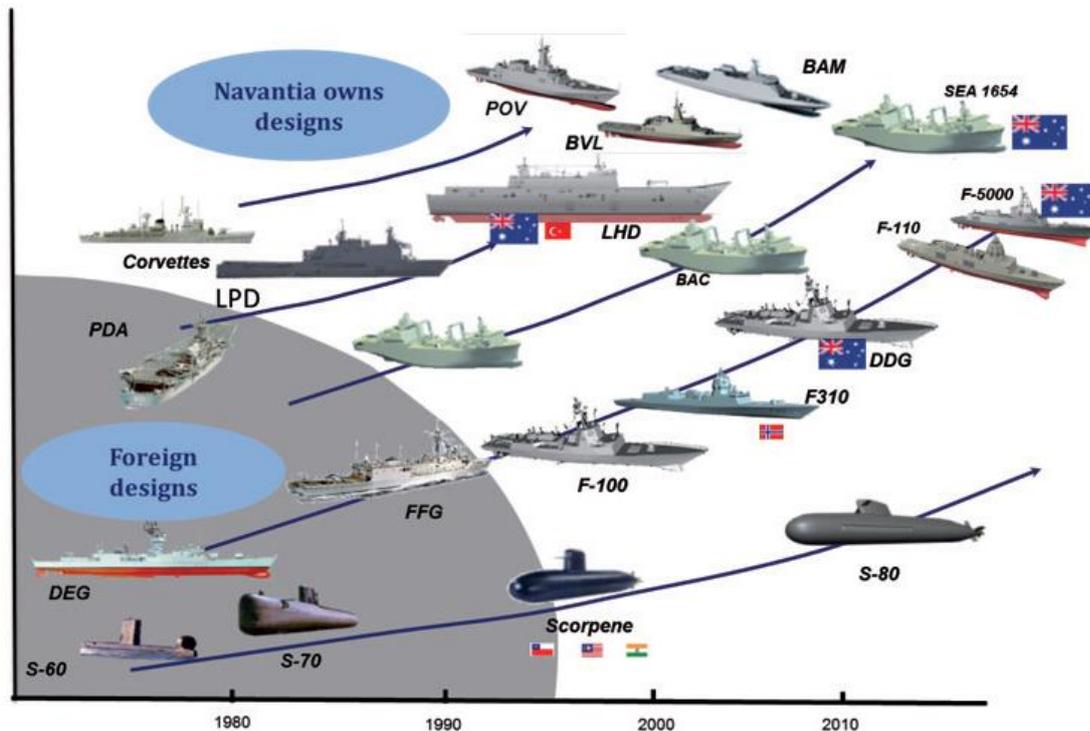


Figura 2. Evolución de las embarcaciones construidas en Navantia (Fuente: A. Recamán Rivas, 2017).

En las tres últimas décadas se produjeron buques que presentaban una alta complejidad tecnológica para cumplir con los requisitos de los clientes. Esto, unido a los rígidos plazos de entrega impuestos por los contratos, obligaba al Navantia a incorporar los nuevos avances técnicos y tecnológicos, mejorando la eficacia y eficiencia de su proceso productivo e incrementando la labor de planificación y gestión de proyectos y la coordinación de las diferentes áreas y departamentos de la entidad, dando lugar a un sistema de construcción integral y entendiendo el buque como un conjunto de productos intermedios. Ya en los últimos años de esta década, la filosofía de la entidad se acerca cada vez más a la industria 4.0, con la evolución hacia lo que denominan como astillero 4.0.

3.4 Cuarta revolución industrial

Durante la Edad Contemporánea se han producido tres procesos de revolución tecnológica, los cuales tuvieron un gran impacto en la manera en la que se producían los bienes.

La primera revolución industrial comenzó a finales del siglo XVIII, afectando a diversos aspectos de la sociedad, ocasionándose una transformación económica, social y tecnológica. El gran promotor de este cambio fue la invención de la máquina de vapor, desplazando así una gran parte del trabajo manual y el uso de tracción animal.

La segunda revolución industrial aparición a finales del siglo XIX. Esta trajo consigo el uso de nuevas fuentes de energía como el gas, el petróleo o la electricidad, nuevos materiales y nuevos sistemas de transporte como el avión o el automóvil con la invención de los motores de combustión interna. Un suceso destacable, además de la

aparición de la energía eléctrica, fue la aparición de la producción en serie, provocando un cambio de paradigma en el ámbito industrial buscando así una producción masificada más eficiente.

La tercera revolución industrial se apoya en las nuevas tecnologías de la información y la comunicación que surgieron a mediados del siglo XX. Un avance de gran relevancia es la implantación de procesos automatizados en la industria, el cual mantiene la filosofía de aumento de la eficiencia y la productividad de la industria.

En el momento actual, se está produciendo una situación similar a la de estos tres cambios mencionados, debidos principalmente a una integración entre los elementos de producción y sistemas de información y comunicación. Dicha integración produce un espacio ciber-físico conocido como Internet de las Cosas (*IoT*), donde se producen comunicaciones entre máquinas, entre máquinas y personas, entre máquinas y bases de datos y todas las combinaciones posibles.



Figura 3. Revoluciones industriales a lo largo de la historia (Fuente: Economipedia).

Esta nueva etapa industrial afecta tanto a la estructura de la propia industria como a los consumidores, para los cuales las empresas ofrecen nuevas soluciones y servicios digitales. Desde una perspectiva operacional, las nuevas tecnologías digitales se emplean para reducir tiempo y trabajo en los procesos, lo que supone un ahorro en cuanto a los costes y una gran mejora de la productividad y la eficiencia. A esta nueva forma de industria se le conoce como Industria 4.0, la cual se asienta sobre las siguientes bases (Lorenz, Rübmann, Waldner y Gerbert, 2015):

- **Big Data y análisis de datos.** Se conoce como Big Data a grandes conjuntos de datos o combinaciones de conjuntos de datos que pueden estar estructuradas o no. Por su gran tamaño, complejidad y velocidad de crecimiento superan las capacidades de las tecnologías tradicionales en cuanto a su captura, gestión procesamiento o análisis. Por ello, surgen nuevas tecnologías para el procesamiento de estos que nos permiten realizar una gestión y análisis de los mismos para realizar una optimización de procesos industriales, mejorar el consumo energético y la calidad de la producción. Estos

datos pueden venir de diferentes fuentes, como bases de datos que una empresa acumula a lo largo de su existencia, la implantación de sensores en una planta industrial, información sobre el mercado y los consumidores, etc.

- **Robots autónomos.** El desarrollo ocasionado en los últimos años en el campo de la robotización ha dado lugar a la creación de robots con altas competencias y un cierto grado de inteligencia artificial que les permite trabajar largos periodos sin supervisión humana. En general, esta nueva generación de robots presentan la capacidad de aprender y adaptarse a medida que experimentan nuevas situaciones en el desarrollo de una tarea.
- **Simulación de procesos.** La simulación es una herramienta de representación de procesos industriales que se apoya en el uso de modelos. Con esta podemos reproducir virtualmente el funcionamiento de una planta, estudiar su comportamiento ante diferentes hipótesis y cambios, y extraer conclusiones para realizar una planificación, optimización o comparación de alternativas. Este será uno de los principales pilares en los que se apoyará este trabajo.
- **Sistemas para la integración vertical y horizontal.** Esto supone una integración de la gestión de la información tanto dentro de los departamentos o áreas de la propia empresa como entre la empresa, proveedores, y clientes desarrollando así una estrategia de actuación compenetrada.
- **IoT (Internet de las cosas).** La idea básica de este concepto es la intercomunicación que se da entre distintos dispositivos como móviles, ordenadores, sensores, actuadores, satélites, etc. Gracias a la mejora en cuando a la calidad, velocidad y posibilidades de conexión, con la interacción y cooperación entre estos elementos se puede lograr una toma de decisiones en tiempo real en el contexto del proceso de fabricación.
- **Ciberseguridad.** Con los aspectos ya mencionados se manifiesta la gran dependencia que tiene la industria a estar en constante conexión con sistemas externos. Esto, si no se impone alguna barrera de protección de por medio, puede dar lugar al robo de datos de vital importancia para la empresa, la entrada de software malicioso, o cualquier otro tipo de ataque informático. Por este motivo, la ciberseguridad es un aspecto fundamental que debe tener en cuenta el nuevo modelo de industria.
- **Cloud computing.** Este concepto hace referencia al uso de datos compartidos entre diferentes servidores, incluidos servidores externos a la compañía. Esto ayuda a reducir costes y tiempo, resultando en un proceso de manejo de datos más eficaz y eficiente.
- **Fabricación aditiva.** La aplicación de los nuevos métodos de fabricación que aparecieron en los últimos años es otro de los ejes clave de la Industria 4.0. Técnicas como la fabricación por adición de materiales fomentan la creación de nuevos composites, que pueden suponer grandes ventajas en diversos sectores por las mejores propiedades que presentan estos nuevos materiales. Otra aplicación interesante es la fabricación de prototipos de productos, reduciendo significativamente el tiempo de creación de prototipos frente a otros métodos.

- **Realidad virtual y realidad aumentada.** Pese a ser uno de los campos con menos desarrollo de la Industria 4.0, son múltiples las aplicaciones de la realidad virtual y la aumentada en la industria, además de otras ramas como la medicina o la aviación.



Figura 4. Fundamentos de la Industria 4.0 (Fuente: Boston Consulting Group).

3.4.1 Astillero 4.0

Es de interés para cualquier país desarrollado el mantener las capacidades industriales y de innovación como motores de su economía. Entendiendo que no hay crecimiento económico sostenido sin una industria fuerte e innovadora, existe un enfoque global para mejorar el desempeño de la industria aprovechando las nuevas tecnologías, conocido como la Cuarta Revolución Industrial (Industria 4.0), que ya presentamos anteriormente. Como resultado de los avances tecnológicos en digitalización y conectividad, nace el concepto de la fábrica del futuro atendiendo a los siguientes aspectos (A. Recamán Rivas, 2017):

- Arquitectura de estandarización y referencia.
- Gestión de sistemas complejos.
- Infraestructura de banda ancha integral para la industria.

- Seguridad y protección.
- Organización y gestión del trabajo.
- Formación y desarrollo profesional continuo.
- Eficiencia de recursos.
- Marco normativo.

Con el fin de cumplir con estos requisitos de mejora de la productividad y de reducción de los costes empresariales, Navantia ha implantado un modelo de Astillero 4.0. Este modelo tiene como objetivo aplicar y optimizar las tecnologías de la Cuarta Revolución Industrial para aplicaciones de construcción naval. Las bases sobre las que se asienta esto son las siguientes (A. Recamán Rivas, 2017):

- Creación de una industria de construcción naval sostenible.
- Modernización de las instalaciones, ofreciendo así productos competitivos a nivel internacional.
- Modernización de las infraestructuras de TIC (Tecnologías de la Información y la Comunicación) que servirá de base para la implantación de gemelos digitales del astillero.
- Creación de nueva mano de obra cualificada capaz de afrontar los nuevos retos industriales.

Teniendo estas pautas en mente, el desarrollo de las tecnologías para el nuevo astillero 4.0, en concordancia con los nueve fundamentos ya presentados de la cuarta revolución industrial, se apoya en los siguientes avances (A. Recamán Rivas, 2017):

- **Digitalización de toda la maquinaria de los talleres.** Esto permitirá una integración vertical con los sistemas de información, facilitando la optimización tanto del consumo energético como de las labores de mantenimiento en las máquinas.
- **Robots de apoyo.** Implementar robots permitirá la automatización que un gran número de tareas, desde las más difíciles y repetitivas, hasta tareas de control de calidad o incluso de carácter administrativo.
- **Fabricación aditiva.** El empleo de nuevas técnicas de fabricación 3D dan paso a que la fabricación de piezas complejas desde un modelo 3D previo creado digitalmente sea algo tan sencillo como imprimir una hoja de papel.
- **Realidad virtual.** En concordancia con la creación de un “gemelo digital”, esto nos permitirá explorar completamente el modelo antes de la construcción física. Por otro lado, la realidad aumentada proporciona la ventaja de facilitar el acceso a toda la información de un componente específico para los operarios.
- **Minería de datos.** Eso facilitará la extracción de información relevante de las grandes cantidades de datos que se generaran dada la integración vertical y horizontal en el Astillero 4.0.

- **Internet de las Cosas.** Esta red conectará a todas las partes interesadas y participes del astillero como personas, productos e instalaciones, ya sea dentro del propio astillero o fuera de este, permitiendo una conectividad global. Esto podría incluir la conexión de toda una cadena de suministro geográficamente dispersa o la conexión entre distintos astilleros.
- **Almacenamiento en la nube.** Los sistemas de almacenamiento de información en la nube eliminarán las fronteras para almacenar, computar e intercambiar la información existente.
- **Ciberseguridad.** Dadas las redes de gran alcance que se implementan como el internet de las cosas o el almacenamiento en la nube, será necesaria la aplicación de la ciberseguridad garantizando la protección de la información.
- **Modelos virtuales.** Eso permitirá la optimización previa de las configuraciones de los barcos y sus sistemas, además de poder simular los procesos de producción necesarios para su fabricación, como se realizará en este trabajo con el procesado de los bloques.

Las instalaciones físicas del Astillero 4.0 evolucionarán junto con los sistemas de información y comunicación, permitiendo la conectividad entre personas, productos y máquinas, el Internet de las Cosas. El uso de sensores en los procesos de producción, la inserción de robots especializados, así como el *big data* y la minería de datos, permitirán la identificación más temprana de defectos y errores que se corregirán por adelantado. A su vez, el empleo de modelos digitales permitirá la simulación de diferentes escenarios de actuación, facilitando así la planificación de los procesos. Estos aspectos mejorarán en gran medida la garantía de calidad, además de permitir un flujo continuo de información y materiales en el astillero y sus clientes y colaboradores como pueden ser los proveedores. De este modo, se integra la cadena de valor desde la fase de innovación hasta el diseño y la fabricación, y continúa a través del servicio que se da durante la vida del producto.

Este nivel de integración horizontal permite la colaboración de entidades como las universidades, centros científicos y tecnológicos, además de la red de proveedores más tradicionales, ya que dicha integración permite mejorar reglas de participación, basándose en un nivel de transparencia y alcance previamente impensable. Además, la aplicación efectiva de las nuevas tecnologías requiere la contribución de la ingeniería en todos los aspectos del Astillero 4.0, en los ejes vertical y horizontal y a través de la vida de los barcos. De esta forma, los sistemas de ingeniería se utilizarán de forma rutinaria como una parte integral del proceso de construcción de buques.

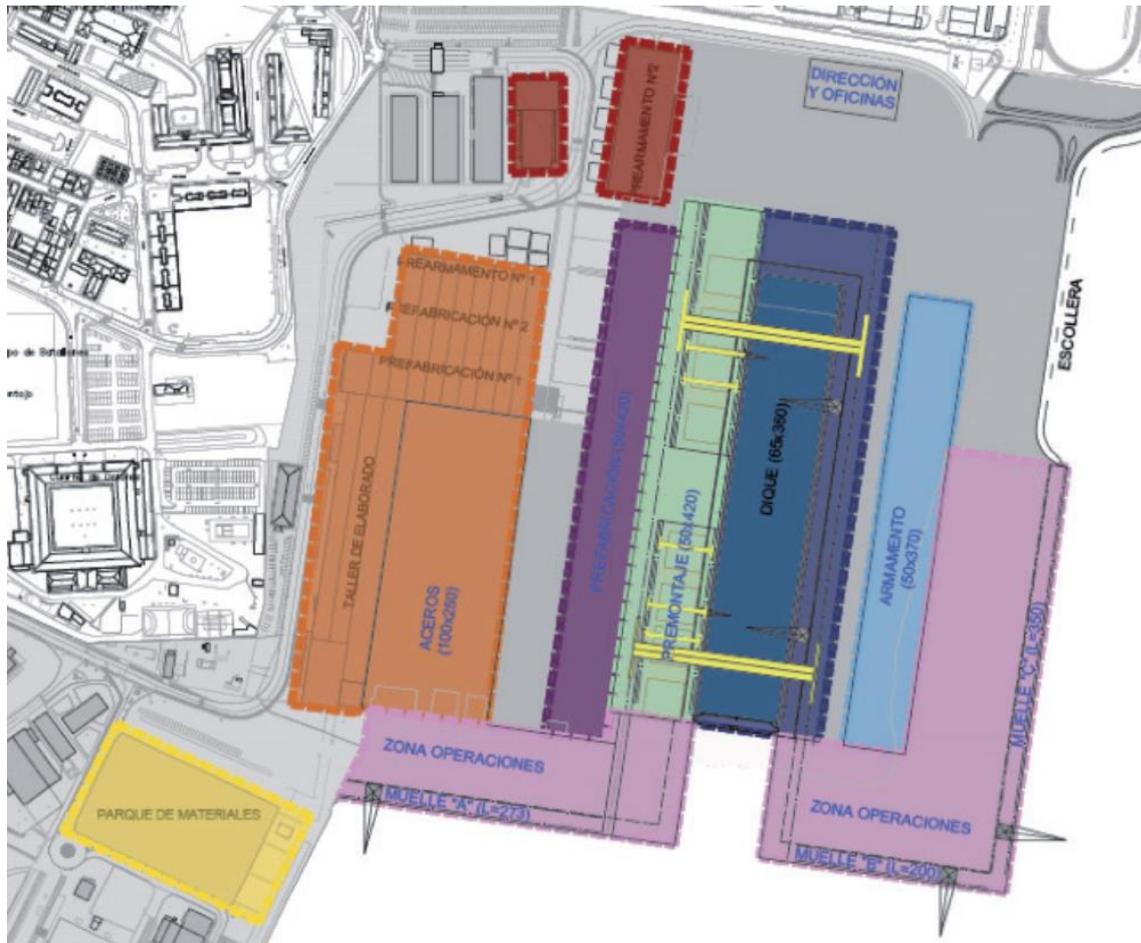


Figura 5. Modernización del espacio del astillero de Ferrol (Fuente: A. Recamán Rivas, 2017).

En este contexto, este trabajo trata de mejorar la eficiencia de uno de los procesos que se dan a cabo en el astillero en cuanto a la producción naval, concretamente en cuanto a la programación de tareas, empleando una filosofía que se adecúa a la de la Industria 4.0. Esto se apoyará en un modelo virtual para elaborar una mejor planificación del proceso en cuestión, que desarrollará posteriormente en este trabajo.

3.5 Programación de tareas

Antes de comenzar con la programación de tareas espacial que se tratará en este proyecto, conviene dar un marco teórico general del concepto de la programación de tareas. Esta programación, también conocida como *scheduling* está ampliamente implantada en diversos procesos y servicios industriales, cuyo objetivo es optimizar la asignación en el tiempo de un conjunto de actividades y recursos limitados, que en función del área en la que se aplique pueden tratarse de tareas y recursos muy diferentes (Pinedo, 1995). Por ejemplo, en el caso de la programación de vuelos, las tareas podrían ser los despegues y aterrizajes de los aviones y los recursos las pistas donde se realizan los mismos; en el de una empresa dedicada a la construcción las

tareas serían las diferentes etapas del proceso constructivo y los recursos podrían ser la maquinaria de la que se dispone o los operarios; en el de una empresa dedicada a los sistemas de procesamiento de información, las el procesamiento de datos se podría identificar como tareas y las unidades de procesamiento con los recursos, etc. En el caso que se presenta en este proyecto, a pesar de ser un tipo de programación de tareas concreto, el espacial, sigue también un esquema como este, dónde la asignación de los bloques serían las tareas y las divisiones del espacio disponible los recursos. Problemas de estas características aparecen en la gran mayoría de procesos productivos, que si son gestionados de manera correcta, pueden suponer una reducción de los costes y del tiempo de los procesos en cuestión, incrementando la eficiencia de la planta con un mejor aprovechamiento de los recursos. La implantación de sistemas de organización de la producción como el *just in time* obligan al perfeccionamiento de las empresas en la cuestión de la programación de tareas. Todos estos problemas de programación de tareas presentan un número de tareas finito, igualmente que el de agentes o recursos, que se describen como máquinas en el proceso más general (D. Morillo, L. Moreno y J. Díaz, 2014). De entre todos, el más similar al que se va a tratar en este documento es el de programación de máquinas en paralelo, el cual consiste en la secuenciación de varios trabajos y la distribución de estos entre varias máquinas en paralelo.

Los problemas que se engloban en esta área suelen ser problemas de optimización combinatoria, del tipo NP-hard, que a su vez es un subconjunto de la clase NP (problemas en los que no es posible obtener una solución en tiempo polinomial en todas sus instancias) (Suárez, 2011). La gran complejidad del *scheduling* es de carácter computacional, donde el espacio de las soluciones viene dado por la cardinalidad recursos por tareas por tiempo, por lo que dependiendo del tamaño del problema resulta más o menos costos, resultando irresoluble en algunos casos (Matfeld, 1996). Sumado a lo anterior, se establece restricciones en la mayoría de los casos, como el tiempo máximo para la fabricación de algún producto, limitaciones técnicas de algunas tareas, escasez de recursos, etc., las cuales dificultan aún más la programación

El concepto de la programación de tareas comenzó a ser tratado a principios del siglo XX en trabajos como los de Henry Gantt. Sin embargo, las publicaciones relacionadas con el *scheduling* tardaron hasta mediados de siglo en salir en la literatura de investigación de operaciones. Ya en los años sesenta, se realizaron numerosos estudios en diferentes casos, en los que se aplicaba la programación dinámica y la programación entera. En las siguientes décadas, se continuaría con el desarrollo de este campo, apareciendo nuevos métodos y enfoques para el modelado y la resolución de estos casos, algunos más eficaces que otros. Algunos ejemplos de estas metodologías son las siguientes: sistemas expertos, programación matemática, reglas de despacho, algoritmos genéticos, redes neuronales o búsqueda tabú, etc., siendo la mayoría métodos heurísticos o metaheurísticos, conceptos que engloban también al método que se desarrolla en este trabajo y que se explicarán posteriormente.

3.5.1 Programación de tareas en la empresa

Dentro de una organización, la aplicación de la programación de tareas supone una interacción con las diferentes secciones de la misma, que a menudo ocurren dentro de un sistema de información empresarial (Pinedo, 1995). Lo más habitual es que las fábricas u organizaciones de servicios modernas cuenten con un complejo sistema de información, con una base de datos que puede estar en servidores propios o no, formando una red de comunicación empresarial con los ordenadores o

estaciones de trabajo, terminales de entrada de datos, etc. conectados con un ordenador central. Todo este sistema está controlado por un *software* denominado sistema de planificación de recursos empresariales (ERP). Este, hace de vía de transporte para la información que atraviesa la empresa en todos los niveles organizativos, de manera que las decisiones y bases de datos de las diferentes partes de una empresa sean integradas de forma que las consecuencias de las decisiones tomadas en un área de esta se reflejen en los sistemas de planificación y control del resto de la empresa. Esta información se actualiza en tiempo real, estando disponible para todo el personal conectado con el sistema ERP.



Figura 6. Esquema de funciones del ERP (Fuente: Parada Visual).

En la planificación de la producción, que abarca una programación a medio y largo plazo, se realiza una labor de estimación de los recursos necesarios, niveles de inventarios, etc., la cual afecta a al problema que estamos tratando. Este problema de programación de tareas también se comunica con entradas del control en planta, debido a situaciones inesperadas que puedan ocasionarse en el proceso que obliguen a reestructurar la programación para poder mitigar los efectos negativos como posibles retrasos. En la siguiente figura se representa esquemáticamente los flujos de información que se generan entorno al programa maestro de producción, el cual engloba a la programación de tareas.

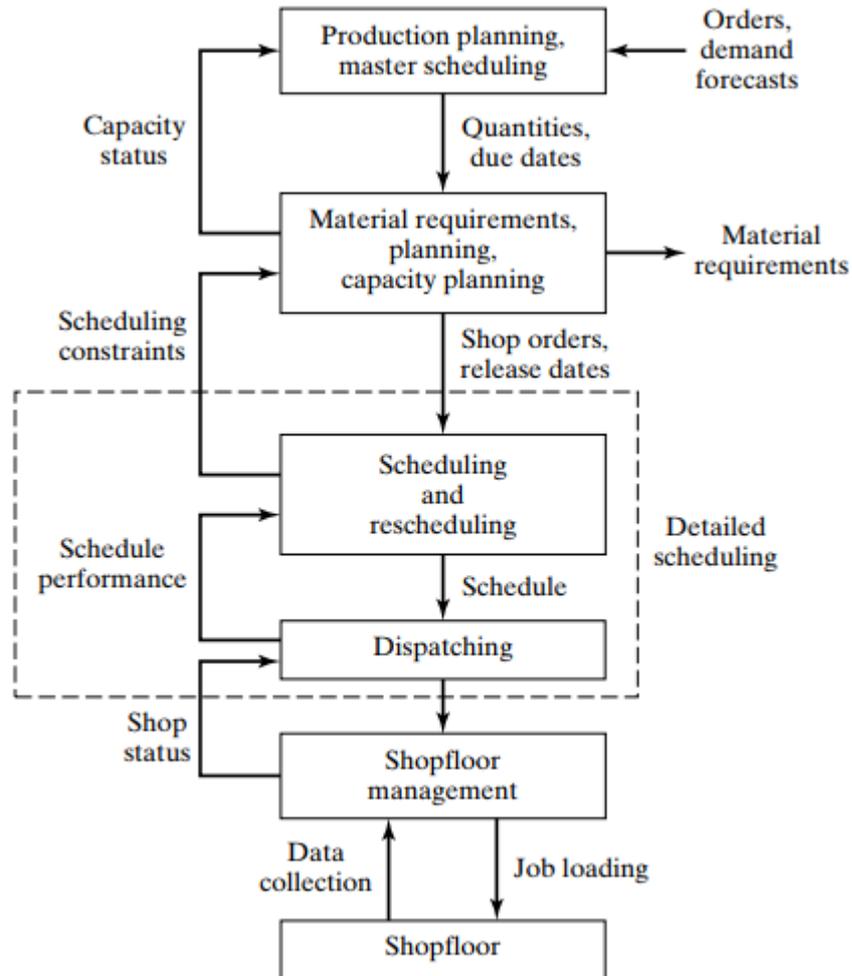


Figura 7. Diagrama de los flujos de información en un sistema de producción (Fuente: Pinedo, 1995).

3.5.2 Métodos heurísticos

La definición que se le puede dar a método heurístico es la de ser una técnica que busca una buena solución con un coste computacional razonable sin garantizar la obtención de la solución óptima, pero sí tratara de acercarse al óptimo (Rayward-Smith, 1996). Estos, se emplean en problemas donde por su amplitud o complejidad sería imposible intentar resolverlos de forma por algoritmos de otro tipo, ya que supondrían un coste computacional o un tiempo de cálculo desproporcionado, como puede ser el caso de la programación de tareas espacial.

Generalmente, el proceso de estas técnicas es un algoritmo iterativo, de manera que en cada iteración se realiza una búsqueda para alcanzar una solución que puede ser mejor que la de la interacción anterior, de manera que se va acercando cada vez más al óptimo. Tras un número de iteraciones razonable habremos alcanzado una solución suficientemente buena. En general, estos métodos se basan en reglas de sentido común sobre como buscar una buena solución, las cuales son adaptadas para ajustarse al tipo de problema que se quiere resolver. Por ello, la mayoría de los

métodos heurísticos son métodos *ad hoc*, ya que la heurística se diseña para un problema en concreto no para una gran variedad de aplicaciones.

3.5.2.1 Aplicación de los métodos heurísticos

Es adecuado el desarrollo y aplicación de un método heurístico ante un problema de optimización determinístico si se cumple alguna de las siguientes pautas (R. Martí):

- Debido a la naturaleza del problema, no existe ningún método exacto para su resolución.
- Se conoce un método exacto para el problema, pero su empleo supone un alto coste computacional.
- La heurística ofrece más flexibilidad que el método exacto existente, permitiendo la incorporación de condiciones de difícil modelización.
- El modelo matemático que describe el problema es excesivamente grande o complejo desde el punto de vista lógico.
- La toma de suposiciones o aproximaciones que simplifiquen el problema descompone estructuras del modelo que son vitales en el contexto del mundo real, resultando en una solución no viable.
- La heurística se emplea como parte de un procedimiento global que garantiza la solución óptima de un problema, bien porque este método proporciona una buena solución inicial de partida o bien porque el método heurístico sea un paso intermedio del procedimiento.

Entonces, en la aplicación de estos métodos nos centraremos en dos objetivos. Por un lado en obtener una buena solución inicial y por otro lado dada una adecuada solución inicial, el método heurístico parte de esta como un procedimiento intermedio que busca mejorar dicha solución. Como cuando se resuelve un problema mediante heurísticas no se garantiza encontrar el óptimo, se debe medir la calidad de los resultados, evaluando la eficiencia del método para poder determinar su capacidad frente a otros. Por ello, es deseable que el algoritmo heurístico incorpore las siguientes características para ser considerado un buen método (R. Martí):

- **Eficiencia.** El coste computacional debe ser razonable para la obtención de una buena solución.
- **Bondad.** Las soluciones generadas por el método no deben alejarse en promedio de la solución óptima del problema.
- **Robustez.** La probabilidad de que nos alejemos de dicho óptimo, es decir, de encontrar una mala solución, debe ser mínima.

3.5.2.2 Clasificación de los métodos heurísticos

A lo largo de los años se han diseñado diversos métodos heurísticos para una gran variedad de problemas, ya que era difícil realizar una generalización de estos que

permitiese una aplicación de alguna heurística ya desarrollada a un problema diferente. Sin embargo, es posible agrupar estos algoritmos por características similares en las siguientes clases (R. Martí):

- **Métodos de descomposición.** Estos tratan de dividir el problema inicial en varios subproblemas que sean más sencillos de resolver, sin olvidar que todos ellos pertenecen al mismo problema.
- **Métodos inductivos.** Este tipo de métodos se basan en generalizar de versiones más pequeñas o sencillas al caso completo. Identificando las propiedades o técnicas aplicadas en los casos menores más fáciles de analizar para poder emplearlas en el problema completo.
- **Métodos de reducción.** Su objetivo es identificar propiedades que se cumplen mayoritariamente en las soluciones buenas que se fueron obteniendo. Una vez identificadas, se fija su valor como una restricción del problema, acotando así el espacio de las soluciones y simplificando el problema. Tiene la contrapartida de que puede dejar fuera soluciones mejores por haber restringido ciertas variables.
- **Métodos constructivos.** Tratan de ir construyendo paso a paso una solución desde cero. Habitualmente son procesos deterministas que suelen estar basados en mantener la mejor elección que se ha aplicado en cada iteración.
- **Métodos de búsqueda local.** Frente a los anteriores, este método parte de una solución inicial, la cual va mejorando progresivamente. El proceso consiste en realizar en cada iteración un desplazamiento de una solución a otra cuyo valor sea mejor que la actual, de manera que el proceso finaliza cuando no es posible desplazarse a una solución mejor.

3.5.3 Métodos metaheurísticos

Frente a las heurísticas, las ideas de los métodos metaheurísticos tienen un carácter más genérico, proporcionando una estructura general y una orientación para la construcción de una heurística para un problema particular. En la actualidad, los metaheurísticos son una de las técnicas más importantes entre los métodos cuantitativos que se aplican en ingeniería, proporcionando una guía general para crear nuevos algoritmos híbridos, combinando diferentes conceptos de diversas áreas como la inteligencia artificial, la evolución biológica y los mecanismos estadísticos. En definitiva, estos tratan de abstraer ideas generales para crear algoritmos que con pocas variaciones puedan ser aplicados en un gran número de problemas, a diferencia de las heurísticas que son mucho más específicas para cada caso. Este tipo de técnicas son muy habitualmente empleadas en la resolución de problemas de optimización combinatorias, por ser uno de los mejores métodos para aplicar en estos casos.

En estos mecanismos se persigue una serie de propiedades que favorezcan el interés práctico y teórico de estos, aunque no es posible mejorar todas las

características dado que algunas son parcialmente contrapuestas. Estas son las siguientes (Suárez, 2011):

- **Simple.** El método debe basarse en un principio sencillo, claro y fácil de comprender.
- **Precisa.** La formulación de las reglas a seguir debe ser concreta.
- **Coherente.** Los elementos del algoritmo deben deducirse naturalmente de sus principios.
- **Eficaz.** El método debe presentar una alta probabilidad de alcanzar una buena solución.
- **Eficiente.** La carga computacional causada por la aplicación del algoritmo no debe ser elevada.
- **General.** Se debe poder aplicar a un amplio abanico de problemas, presentando un buen rendimiento en su desempeño.
- **Adaptable.** Debe poder adaptarse a diferentes contextos o modificaciones importantes del modelo.
- **Robusta.** La metaheurística no debe ser muy sensible ante pequeñas alteraciones del modelo o del contexto de aplicación.
- **Interactiva.** El método debe permitir que el usuario pueda aplicar sus conocimientos para mejorar el procedimiento.
- **Múltiple.** Debe proporcionar diferentes soluciones alternativas de calidad.
- **Autónoma.** El procedimiento debe estar libré de parámetros, o que al menos se puedan establecer automáticamente.

Para la evaluar estos mecanismos se debe tener presente tanto la eficiencia de los recursos como la eficacia, en tanto que alcanza una buena solución. Para verificar la primera, se contrasta experimentalmente con el empleo de un tiempo computacional moderado; para la segunda, se comprueba si resuelve con éxito problemas de un banco de casos para los que ya se conocen las soluciones.

3.5.4 Función objetivo

En el caso de la programación de tareas, toda heurística empleada para la está ligada a una o varias funciones objetivo, las cuales pueden ser de minimización del tiempo de flujo de una tarea, de maximización de la utilización de un recurso etc. Previamente a la descripción de estos objetivos, es necesario definir los siguientes conceptos (A. García del Valle):

- Fecha de finalización C_i . Es el instante en el que se termina de procesar la última operación de un trabajo i .
- Tiempo de flujo F_i . Se define como el tiempo transcurrido entre la llega y la finalización de un trabajo i .

$$F_i = C_i - a_i$$

Siendo a_i la fecha de entrada del trabajo i en el sistema.

- Tiempo de espera W_i . Este es el tiempo que un trabajo i está en colas esperando a que los trabajos que le preceden sean procesados. Junto con el tiempo de procesamiento p_i , forman el tiempo de flujo.

$$F_i = C_i - a_i = W_i + p_i$$

- Demora o retraso L_i . Es la diferencia de tiempo entre la fecha establecida para la entrega de un trabajo i y el momento en el que se produce la entrega realmente.
- Tardanza T_i . Si un trabajo i se termina tras su fecha de entrega se denomina tardanza.

$$C_i > d_i$$

Siendo d_i la fecha establecida para la entrega del trabajo i .

- Prontitud E_i . En cambio, si un trabajo i , se termina antes de su fecha de entrega se denomina prontitud.

$$C_i < d_i$$

Conocido esto, los objetivos más frecuentes empleados en la programación de tareas son los siguientes (Pinedo, 1995):

- **Makespan**. Este concepto hace referencia al tiempo de finalización del último trabajo que entra en el sistema, buscando la minimización de este.

$$C_{max} = \max_i(C_i)$$

- **Tiempo de flujo ponderado**. Con este objetivo se persigue minimizar el tiempo que las tareas permanecen en el sistema en función de su importancia w_i .

$$F_{wt} = \sum_i w_i F_i$$

- **Demora ponderada**. Trata de reducir el margen de tiempo negativo o positivo que las tareas exceden sobre su fecha límite, con el fin de evitar una penalización w_i .

$$L_{wt} = \sum_i w_i L_i$$

- **Tardanza ponderada.** Este objetivo es similar al anterior pero únicamente se contabilizan los excesos de tiempo positivos, penalizando este hecho de acuerdo con la importancia relativa de las tareas en el sistema w_{Ti} .

$$T_{wt} = \sum_i w_{Ti} T_i$$

- **Tiempo de flujo máximo.** Trata de minimizar el tiempo que las tareas permanecen dentro del sistema.

$$F_{max} = \max_i(F_i)$$

- **Demora máxima.** Con este objetivo se intenta disminuir el margen que las tareas exceden de su fecha límite, tanto positivo como negativo.

$$L_{max} = \max_i(L_i)$$

- **Tardanza máxima.** Trata de disminuir los márgenes de tiempo positivos que las tareas exceden de su fecha límite.

$$T_{max} = \max_i(T_i)$$

- **Número de tareas tardías.** En este objetivo se persigue la minimización del número de tareas que se retrasan,

$$N_{wt} = \sum_i W_i F_i \rho(T_i)$$

$$\rho(x) = \begin{cases} 1, & \text{si } x > 0 \\ 0, & \text{si } x \leq 0 \end{cases}$$

- **Just in time.** En este tipo de organización de la producción se intenta minimizar el almacenaje de materias primas que se usarán para la fabricación de un producto en la planta, de manera que los aprovisionamientos lleguen en el momento en el que se van a emplear y se evite por tanto el coste asociado a mantener un nivel de inventario. Por ello, en este objetivo se penaliza tanto por entregas prematuras como por entregas tardías.

$$\varepsilon T_{wt} = \sum_i (w_{E_i} E_i + w_{T_i} T_i)$$

En el caso tratado en este documento se realizará una optimización basándonos en la reducción del *makespan*.

3.5.5 Programación de tareas espacial

Como ya se ha presentado anteriormente, en industrias a gran escala como la construcción naval u otras como la aeronáutica, las subdivisiones de lo que se pretende construir suelen presentar unas grandes dimensiones y un peso considerable, ocupando así grandes áreas de trabajo. Para la manipulación de estos bloques se emplean recursos y equipos de alto coste, como pueden ser las grúas. Por tanto, el espacio que dispone de estos equipos en la gran mayoría de casos se encuentra limitado, pudiéndose generar cuellos de botella que pueden intervenir negativamente en el cumplimiento de los plazos. Con esto, la programación que considera estos recursos espaciales, además de los que se suelen considerar como son la mano de obra y la maquinaria, se denomina programación de tareas espacial.

El problema de la programación de construcción de bloques espacial consiste básicamente en la optimización de la asignación de un conjunto de recursos, que en este caso es el espacio de los parques de construcción de bloques, a un conjunto de órdenes de producción que vienen dadas por la cartera de pedidos de la empresa y, como ya se ha visto, su secuenciación está condicionada por la etapa de erección de los bloques en grada. En este, a diferencia de otros problemas en los cuales los recursos de almacenamiento se simplifican a una cierta capacidad, por las dispares dimensiones de los distintos bloques es necesario realizar un estudio en profundidad para optimizar este almacenamiento. Cabe destacar, en la gran mayoría de casos puede tratar bidimensionalmente el problema, tanto en la definición de los bloques como en la de los parques se omitirá la altura ya que generalmente no habrá obstáculos que interfieran con dicha dimensión.

Este caso se agrupa dentro de problemas de programación de tareas espacial, los cuales son problemas de complejidad computacional NP-completos (englobados dentro de los NP-hard), este, a su vez, está compuesto por otros dos problemas NP-completos, la programación de tareas y la optimización de layout. Por tanto, al pertenecer a esta clase de complejidad, verificar si una solución es válida no resulta complicado, pero sí se dificulta la obtención de la solución óptima. Por esto, la mayor parte de los estudios en este campo se centran en la obtención de algoritmos heurísticos como los que hemos presentado anteriormente con el fin de obtener soluciones próximas al óptimo mediante una serie de reglas, sin el empleo de métodos de resolución exactos.

3.6 Modelado y simulación

En este trabajo, la simulación y el modelado nos permitirán comprobar las soluciones obtenidas con la heurística que se ha desarrollado para la ubicación de bloques en el parque de almacenamiento. A continuación se expondrá una introducción a esta disciplina, la cual constituye una de las bases de la industria 4.0.

Comenzando por el concepto de modelo, este es una representación de un sistema compuesto por evento y/o elementos los cuales pueden ser reales o ideados. De manera general, un sistema puede ser definido como un conjunto de componentes interdependientes organizados para un propósito (Coyle, 1996). Estos, pueden ser agrupados en las siguientes cuatro clases (Chechland, 1981):

- **Sistemas naturales:** son aquellos originados por causas físicas o químicas del propio universo, por ejemplo el átomo, el sistema solar, el clima, etc.
- **Sistemas físicos:** se trata de sistemas tangibles creados por el ser humano, como puede ser una casa, un coche o una planta de producción.
- **Sistemas abstractos:** sistemas intangibles creados por el hombre. Las matemáticas o la literatura son un ejemplo de esta clase.
- **Sistemas de actividad humana:** son aquellos que el ser humano crea en el ejercicio de su actividad, de manera consciente o inconsciente, como puede ser una familia, una ciudad, un sistema político, etc.

Cualquiera de estos sistemas es susceptible de ser modelado y simulado, aunque en este trabajo nos centraremos en los sistemas físicos. Aplicados a un entorno industrial, este vendría a representar los elementos que participan en el proceso productivo, como pueden ser las máquinas, los operarios, los elementos de transporte interno, etc. Este, en función del objetivo que se persiga en el caso a estudiar, podrá estar más o menos simplificado, pero tratará de representar el sistema de manera adecuada matemáticamente.

A su vez, la simulación es una metodología aplicada para describir el comportamiento del sistema a estudiar, ya sea empleando un modelo matemático o simbólico. De manera simplificada, la simulación es la imitación del funcionamiento de un proceso o sistema real en un intervalo de tiempo. Esto nos servirá para la toma de decisiones, pudiendo extraer información como la detección de cuellos de botella en un proceso, la capacidad máxima necesaria en una cola, el sobredimensionado algunos elementos, etc.

3.6.1 La utilidad de la simulación

La mayoría de los sistemas que se pretenden estudiar son demasiado complejos como para realizar una predicción por métodos analíticos, debido a la variabilidad que presentan estos sistemas, la gran interconexión entre sus elementos, etc. La simulación posibilita la representación de estos incluyendo dicha variabilidad e interconexión, de forma que nos permite realizar predicciones del funcionamiento del sistema. De forma general, es conveniente emplear como herramienta en la simulación si se dan las siguientes pautas (Shannon, 1975):

- No existe una formulación matemática del problema o los métodos analíticos para ello no están suficientemente desarrollados para resolver el modelo matemático.
- Las hipótesis de simplificación que hay que asumir para aplicar los métodos analíticos existentes desvirtúan las soluciones obtenidas.
- La gran complejidad de los procedimientos numéricos que proponen los métodos analíticos existentes resulte en un proceso excesivamente laborioso y costoso.

- Es necesario obtener una historia simulada de un proceso dentro de un horizonte natural dado para poder estimar ciertos parámetros.
- No sea posible experimentar en el proceso real por cualquier causa
- Es necesario realizar una comprensión temporal para estudiar cómo evoluciona un sistema a largo plazo.

La gran potencia de esta disciplina es que nos permite realizar diferentes hipótesis y ponerlas a prueba, obteniendo así una información veraz sobre las consecuencias de cada uno de los escenarios planteados teniendo en cuenta la variabilidad existente. En la mayoría de los casos, estas hipótesis no pueden ser probadas sobre un proceso real, ya sea por el coste y tiempo que supondrían o por las consecuencias negativas que tendrían algunos escenarios, por ello, gracias a la simulación podremos imitar con un ordenador el funcionamiento que tendría un sistema o proceso real en dichos escenarios.

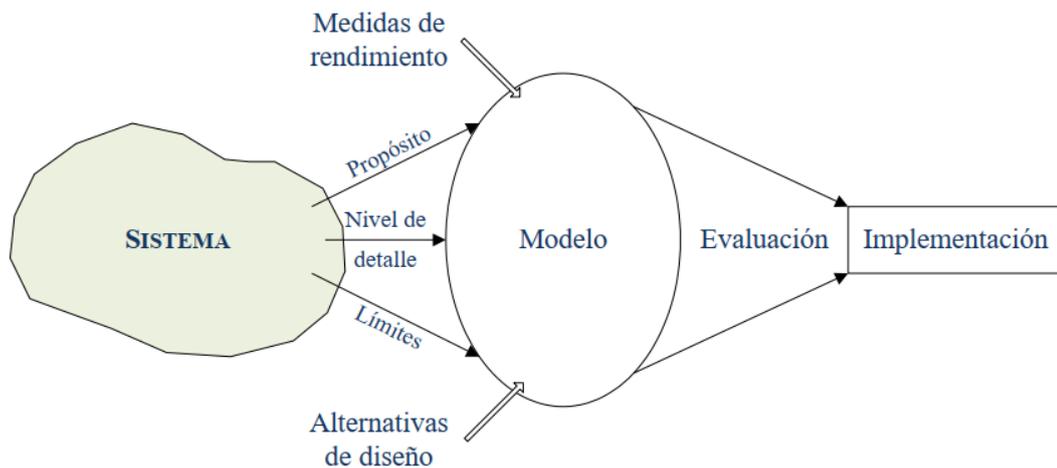


Figura 8. Representación conceptual de la simulación de procesos (Fuente: A. García del Valle).

En el caso de la industria y el negocio, lo más frecuente es emplearla en la simulación de sistemas con colas de espera. Estos, son aquellos que se componen por objetos que se procesan en una serie de etapas, de forma que entre estas se produce una acumulación del inventario en proceso por la incapacidad de procesar tantos elementos aguas abajo. Bajo este esquema si pueden englobar muchos sistemas, ya que esto se puede producir tanto por esperas de personas, de objetos o de información. Los principales campos en los que se aplica (Banks, 1996) son los siguientes:

- Procesos de fabricación
- Sistemas públicos como la medicina, el militar, etc.
- Redes de transporte
- Procesos de construcción
- Restauración y entretenimiento
- Industria alimenticia
- Sistemas de computación

3.6.2 *Ventajas y desventajas de la simulación*

Para el análisis de sistemas y la optimización de estos existen diversos métodos, siendo la simulación uno de ellos. Por ello, es conveniente remarcar cuales son las ventajas que ofrece la simulación frente a otros métodos la justificar su uso.

Además de otros métodos de modelado, cabe la posibilidad de emplear el propio sistema real vía para la experimentación (Pidd, 1998), así que comenzaremos haciendo una comparación con esto. De efectuarse dichos cambios en él, deben considerarse los siguientes aspectos que favorecen el empleo de la simulación (Robinson, 2004):

- **Coste.** Las pruebas en un sistema real frecuentemente suponen un coste en el ámbito de la producción, no solo por el importe de los propios cambios que se aplique como en el caso de añadir una estación de trabajo o un elemento de transporte interno, sino que además en múltiples ocasiones se deben interrumpir las tareas que se realizan rutinariamente en la industria, lo que disminuye el ritmo de producción y puede llegar a ocasionar retrasos. En cambio, la simulación ofrece la posibilidad de ver los resultados de estos cambios sin influir en el proceso real.
- **Tiempo.** El experimentar las diferentes hipótesis en un sistema real puede prolongarse mucho en el tiempo, tanto por el preparamiento para cada una de las pruebas como el tiempo que tardan en manifestarse los efectos a largo plazo. La simulación también requiere el empleo de un cierto tiempo para crear un modelo correcto, pero permite probar diferentes casuísticas rápidamente. De cualquier manera, el orden de magnitud de los tiempos que se manejan con la simulación son significativamente menores que con una experimentación en el sistema real.
- **Control de las condiciones de experimento.** En numerosos casos las variables con las que se pretende experimentar están fuera del alcance de los técnicos en sistemas reales, como por ejemplo la llegada de pacientes en un hospital. La simulación nos permite establecer muchas más condiciones de las que podemos con el sistema real.
- **No existe el sistema real.** En este caso, claramente es inviable aplicar la experimentación en el sistema real, ya que construir una serie de sistemas reales alternativos para cada experimento es altamente ineficiente.
- **Situaciones extremas.** Un modelo nos permite llevar al límite las capacidades del sistema. Aplicar hipótesis con condiciones extremas en un sistema real resulta impensable por las consecuencias que podría tener o mismamente por la imposibilidad de reproducir esas condiciones en un proceso real, como por ejemplo crear una sobrecarga de trabajo que ocasione un gran cuello de botella en el proceso o incrementar ampliamente la plantilla de operarios respectivamente.

En cuanto al empleo de otros modelos, que abarcan desde algo tan simple como cálculos a mano hasta modelos matemáticos complejos o métodos como la

programación dinámica, algoritmos genéticos, etc., la simulación ofrece una serie de ventajas sobre estos, que exponemos a continuación (Robinson, 2004):

- **Modelado de la variabilidad.** La mayoría de los métodos alternativos no son capaces de implementar la variabilidad de la misma forma que en la simulación. Cuanto mayor es la variabilidad que se puede producir en el sistema, más beneficioso es el emplear la simulación. En una comparación entre un análisis estático (sin variabilidad) y uno mediante simulación en una fábrica, el primero de ellos concluía con que se llegaba a la meta de producción requerida, en cambio, mediante la simulación se mostraba que ninguno de los diseños era válido (Robinson and Highton, 1995). Esto se debía la forma simplificada en la que el modelo estático consideraba la variabilidad respecto a las paradas por fallos en las máquinas, ya que se contabilizaba aumentando los tiempos de ciclo medios de los procesos.
- **Suposiciones restrictivas.** La simulación generalmente no requiere suposiciones, aunque algunas veces sí se den con motivo de simplificar el modelo o por falta de datos. Sin embargo, otros enfoques sí requieren de aplicar ciertos supuestos, como por ejemplo los que se toman en la teoría de colas para los tiempos de llegada y de servicio, en los que se asume una distribución en particular que no siempre es la más adecuada. La simulación en cambio nos permite representar el comportamiento a través de cualquier distribución.
- **Transparencia.** La visualización más intuitiva que ofrece la simulación permite que personal de cualquier área comprenda el funcionamiento del modelo y confíe en los resultados que ofrece, frente a modelos matemáticos y hojas de cálculos que son más complicadas de entender.

Frente a todos estos aspectos que hemos mencionado, la simulación también presenta una serie de contras que hay que valorar a la hora de decidir si es apropiado emplear esta metodología (Robinson, 2004):

- **Coste del software.** Los programas dedicados a esta rama suelen tener un precio significativo, además de que desarrollar el modelo también incurre en costes que hay que considerar para saber si compensa aplicar este método frente a otros más simples.
- **Tiempo de aplicación.** El desarrollo de un modelo es un proceso que puede consumir un tiempo importante, por lo que se debe evaluar si el aplicar la simulación va a aportar un valor significativo de cara a los resultados frente a otro tipo de modelos.
- **Toma de datos.** Para la realización de los modelos se necesitan una gran cantidad de datos de los procesos, que en la mayoría de casos no se cuenta con registros previos. Por ello, la tarea de toma de datos puede resultar en una actividad que prolonga aún más en el tiempo el proceso de simulación y obtención de resultados.

- **Experiencia necesaria.** La simulación requiere habilidades respecto a modelaje, estadística, gestión de proyectos, trabajo cooperativo, etc., no es algo tan simple como utilizar un programa de ordenador.
- **Exceso de confianza.** La visualización tan realista en la que se representa el funcionamiento de un proceso nos puede llevar a pensar a que todo lo que estamos observando en el modelo es correcto. Esto hay que tomarlo con precaución ya que puede haber fallos ocultos o simplificaciones erróneas en el modelo que estén dando unos resultados que no predicen correctamente la realidad.

Por otro lado es interesante también la aportación que tiene la simulación en el contexto de la gestión de proyectos y las habilidades humanas. En este sentido, podemos destacar los siguientes beneficios (Robinson, 2004):

- **Fomento de la creatividad.** La simulación permite que cualquier hipótesis pueda ser probada en el modelo sin riesgos, lo que invita a que los participantes del proyecto aporten nuevas ideas sin miedo a las consecuencias de una equivocación.
- **Creación de conocimiento.** El desarrollo y manejo de un modelo de simulación obliga a las personas a pensar en cuestiones que de otro modo no se habrían considerado, ya el modelo requiere la imitación de diversos detalles del proceso que habitualmente se pasan por alto. El modelador realiza diversas tareas como la búsqueda de información, la solicitud de datos e incluso cuestiona suposiciones, que acaban resultando en una mejor comprensión y conocimiento de las características del sistema que se está simulando. Tanto es así, que la propia creación del modelo, sin llegar a la experimentación, pueden generar tanto conocimiento del sistema como para realizar una mejora en el proceso real (Shannon, 1975).
- **Visualización y comunicación.** La facilidad de mostrar los resultados de este método a los superiores, clientes o responsables de este método evita que no se pierdan buenas ideas por no poder tener una demostración fácil, siendo entonces la simulación un buen apoyo en la comunicación de dichas ideas.
- **Consenso de opiniones.** En todos los ámbitos, a menudo surgen diferentes opiniones por los participantes de un proceso. Por ejemplo, los trabajadores de un taller pueden pensar que los recursos de los que se dispone no son suficientes a diferencia de un gerente que piensa que sí se cuenta con una cantidad adecuada. Modelando y simulando dicho proceso se puede extraer una información contrastable de si es necesario o no un aumento de los recursos, sin emplear como justificación las opiniones.

3.6.3 Pasos del proceso de modelado y simulación

Los pasos a seguir para el modelado y la simulación solo los siguientes (A. García del Valle):

1. Definición del problema y planificación del estudio.

Esto implica definir claramente los objetivos del proyecto de simulación. Primero, se debe tener claro el estado actual del sistema y sus causas, así como el estado deseado. El conocer los elementos y las variables que componen el sistema, además de su comunicación e interacción, nos permitirá definir correctamente los objetivos.

2. Recogida de datos.

Los objetivos del estudio establecen gran parte de la clase de datos que deben ser obtenidos y la cantidad de detalles necesaria. Estos pueden ser obtenidos de diversas formas si el sistema es existente, ya sea por observación del propio sistema, por registros históricos del mismo, etc. En el caso de ser un sistema inexistente será necesario realizar estimaciones para su obtención. Es una fase de importancia vital, en la que es necesario que se involucre personas que conozcan bien el proceso real (operadores, supervisores, etc.). En proceso industrial, los datos típicos que se suelen obtener son las condiciones y parámetros del proceso de producción, personal que labora y las acciones que realizan, materiales o materias primas utilizadas, maquinaria o equipos, esquemas, planos, entre otros. Gracias a estos datos se podrán obtener los valores de los parámetros estadísticos de las funciones de probabilidad por las que se rigen los sucesos, los cuales serán utilizados durante la programación del modelo.

3. Formulación del modelo.

En esta etapa se definirán los aspectos del sistema que serán representados en el modelo y cuáles no, de forma que la complejidad del modelo final no exceda los requerimientos que cumplan con el propósito para el cual se está creando. Es importante tener en mente que un nivel de detalle excesivo no ofrece mejores resultados que un modelo que no representa a tanto nivel de detalle, pero que está bien ideado. Como se trata de diseñarlo a nivel conceptual, se pueden emplear herramientas como los diagramas de flujo.

4. Construcción del modelo.

En el inicio de la creación del modelo se debe considerar la magnitud y el nivel de detalle necesario. En un sistema pequeño y simple la modelación es rápida y certera si se cuenta con información completa y correcta, en cambio, si el sistema es complicado o demasiado extenso, es posible recurrir a la fragmentación de este en módulos más pequeños y simples con el fin de realizar una evaluación por separado previamente a integrarlos en un modelo único. Por otra parte, la idealización del sistema debe ser programada en un formato que reconozca un ordenador, teniendo en cuenta los siguientes aspectos: la información de entrada y las condiciones iniciales, que garantizarán la certeza de que el modelo representará correctamente el sistema real; el software a emplear, que debe adecuarse al campo en el que lo vamos aplicar, existiendo múltiples opciones de software de simulación; y las habilidades y experiencia del programador, donde su creatividad e ingenio servirán de base para planear, organizar, dirigir y controlar un proyecto de simulación exitoso.

5. Verificación y validación del modelo.

Tras completar el modelo, se debe realizar una verificación ejecutando el modelo para comprobar que represente con precisión los parámetros, sucesos y lógica del sistema real, quedando expuesto si la información que se ha utilizado era correcta. Por otro lado, la validación trata de asegurar que el modelo es suficientemente bueno como para que se puedan tomar decisiones con sus resultados sobre el sistema real. Por tanto, el modelo será válido si representa adecuadamente al sistema que se pretende estudiar.

6. Diseño de los experimentos de simulación.

Es necesario idear y definir los experimentos que se van a llevar a cabo sobre el modelo. En estos se consideran aspectos como las condiciones bajo las que se hará la simulación, la duración de tiempo de la simulación o el número de simulaciones requeridas para que los datos sean estadísticamente confiables.

7. Ejecución de los experimentos.

En esta etapa se ejecutan los escenarios creados con las condiciones de interés establecidas en el paso anterior. Tras esto, se obtendrá un gran número de datos del modelo que se analizarán posteriormente.

8. Análisis de los resultados.

Mediante la interpretación de la información de salida de los experimentos ejecutados el modelo, podemos extraer conclusiones sobre el sistema real que nos permitirán tomar las decisiones adecuadas para mejorar el sistema.

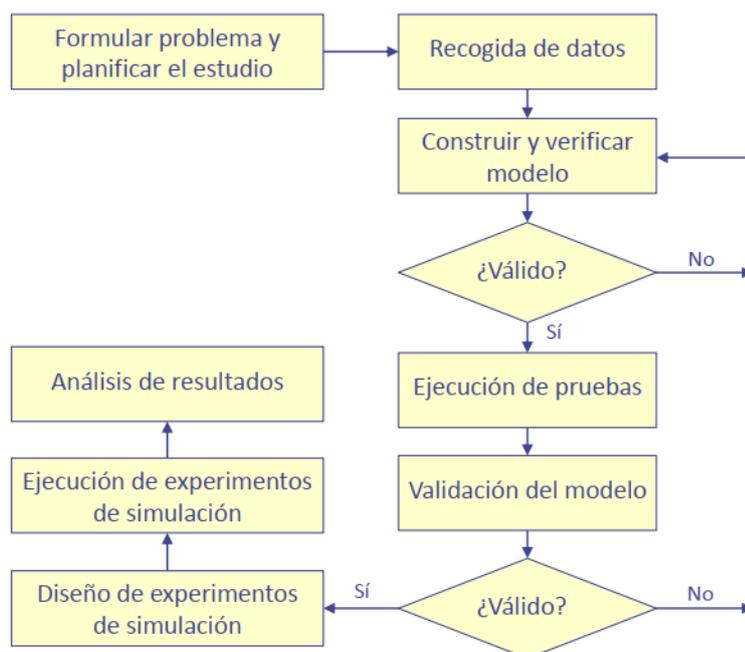


Figura 9. Esquema del proceso de modelado y simulación (Fuente: A. García del Valle)

4 ESTADO DEL ARTE

La programación de tareas espacial tiene un amplio abanico de áreas de aplicación, es por ello que existen trabajos de investigación al respecto en diferentes sectores. En el contexto de la construcción de buques, una primera publicación importante que se centra en la programación de tareas espacial de los parques de bloques es *A Spatial Scheduling System and its Application to Shipbuilding: DAS-CURVE* (1996), en la que se expone la ineficiencia con la que ha sido tratada esta cuestión anteriormente, donde la tarea de programación espacial era realizada por humanos, que aun siendo expertos, se requería mucho tiempo y esfuerzo para obtener una programación satisfactoria. Esta dificultad se debe a necesidad de considerar la geometría de los diferentes bloques, sus tiempos de trabajo, instalaciones requeridas, etc. En la práctica, la programación espacial con una duración superior a seis meses es inabarcable sin la ayuda de métodos como los que se van a tratar. Por ello, manifiesta la necesidad de realizar una programación espacial en el proceso de construcción de bloques de forma automatizada, para lo que propone un algoritmo heurístico para la ubicación de los bloques en el contexto del Proyecto DAS. En dicho proyecto, los objetivos con los que debe cumplir el algoritmo son también minimizar el *makespan* y mantener las cargas equilibradas entre las diferentes etapas de las operaciones de montaje. Por otro lado, los recursos con los que trabaja son la capacidad de la grúa, el espacio del área de trabajo y los operarios.

Para la aplicación de dicho método, primero simplifica bidimensionalmente los bloques a polígonos convexos, ya que a pesar de que algunos presenten algunas zonas con cierta concavidad, esta no podrá ser aprovechada para la colocación de otros elementos en la mayoría de los casos. Otra simplificación en la que se apoya el método propuesto es la orientación de los bloques. En construcción naval, la disposición de un elemento esta prefijada a cuatro orientaciones alternativas (0° , 90° , 180° y 270°) para garantizar la estabilidad de las operaciones con la grúa, por ello, limita la disposición de los bloques a una orientación fija, ya que el problema con múltiples orientaciones finitas puede ser considerado como una extensión del problema con una orientación (J.S. Park en su tesis de 2004 *Optimal nesting for structural elements of ship blocks using differential evolution* muestra que considerar esas cuatro orientaciones es suficiente para obtener buenas soluciones ya que los componentes que formarán el buque no tienen geometrías demasiado complejas). Con esto, este algoritmo se basa en la asignación de posiciones en el parque a bloques en base a criterios de espacios libres tras la inserción de un bloque. La secuencia seguida para la colocación de los bloques no se recalcula dinámicamente, sino que está determinada a priori de acuerdo a un ordenamiento por tiempos límite de producción o duraciones. Además, publicaciones posteriores (Shin, J.G., Kwon, O.H. y Ryu, C., 2008) destacan la dificultad de empleo de su planificación por el usuario y la inflexibilidad de este algoritmo ante situaciones inesperadas que pueden ocurrir en su aplicación real. Por esto, el método de planificación propuesto en este documento, pese a aumentar la eficiencia de la programación espacial en este campo, podrían acercarse más al óptimo mejorando algunas aspectos, como el mismo artículo propone.

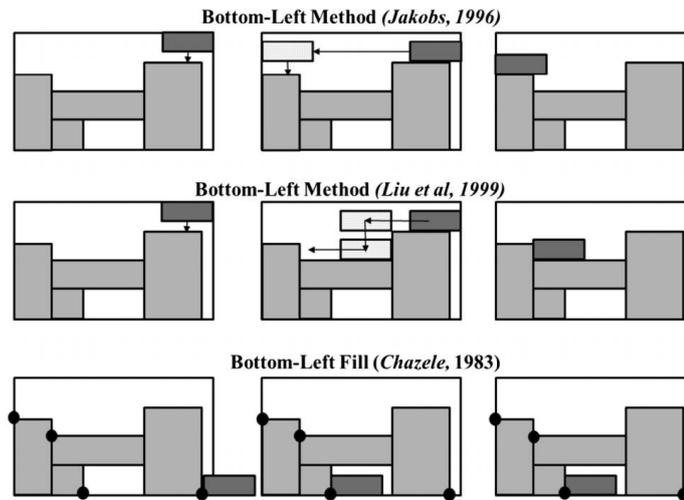


Figura 11. Proceso de ubicación de bloques con algoritmos de tipo bottom-left (Fuente: B. Kwon y G. M. Lee, 2015).

Una limitación de este algoritmo (J.G. Shin, O.H. Kwon, C. Ryu, 2008) es que trabaja con los bloques en una determinada orientación, por ello, este método de dos etapas se combina posteriormente con un algoritmo de evolución diferencial que optimiza la orientación de los bloques, además de optimizar la secuencia de asignación.

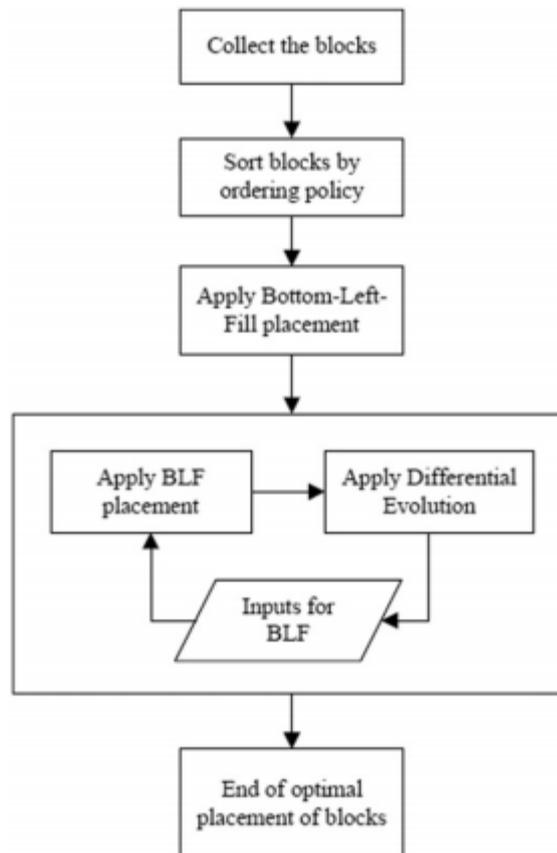


Figura 12. Esquema del proceso de ubicación de bloques (Fuente: J.G. Shin, O.H. Kwon, C. Ryu, 2008).

En la misma publicación, es interesante también la consideración que realizan respecto a las posibles interferencias que puedan surgir durante el proceso de ubicación de los bloques. Mientras se busca el emplazamiento de un bloque, se inspecciona la posible interferencia con otros elementos, como pueden ser otros bloques, instalaciones, etc. Para resolver una interferencia, los bloques se desplazarán hacia arriba o a la derecha, como aparece representado en las siguientes figuras.

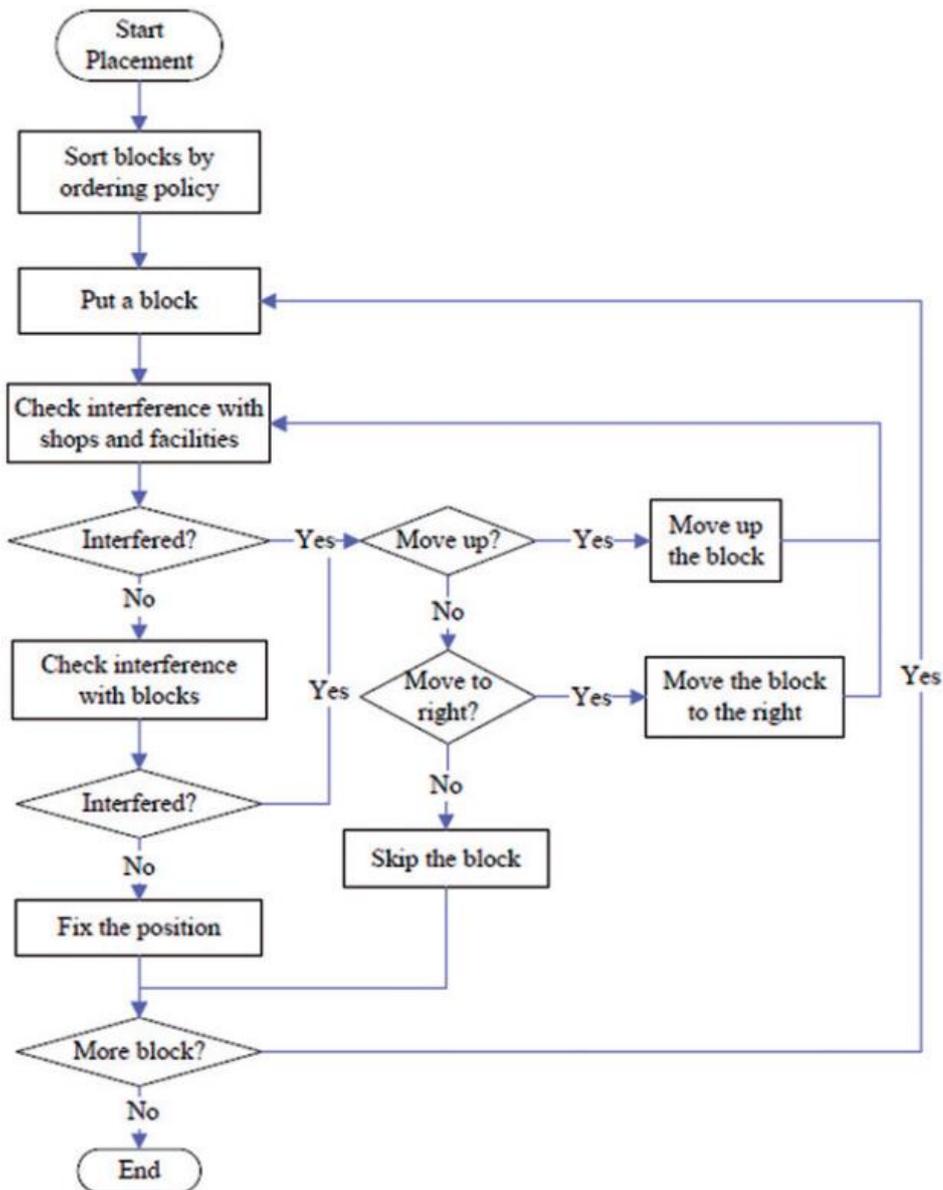


Figura 13. Esquema de actuación considerando las interferencias (Fuente: J.G. Shin, O.H. Kwon, C. Ryu, 2008).

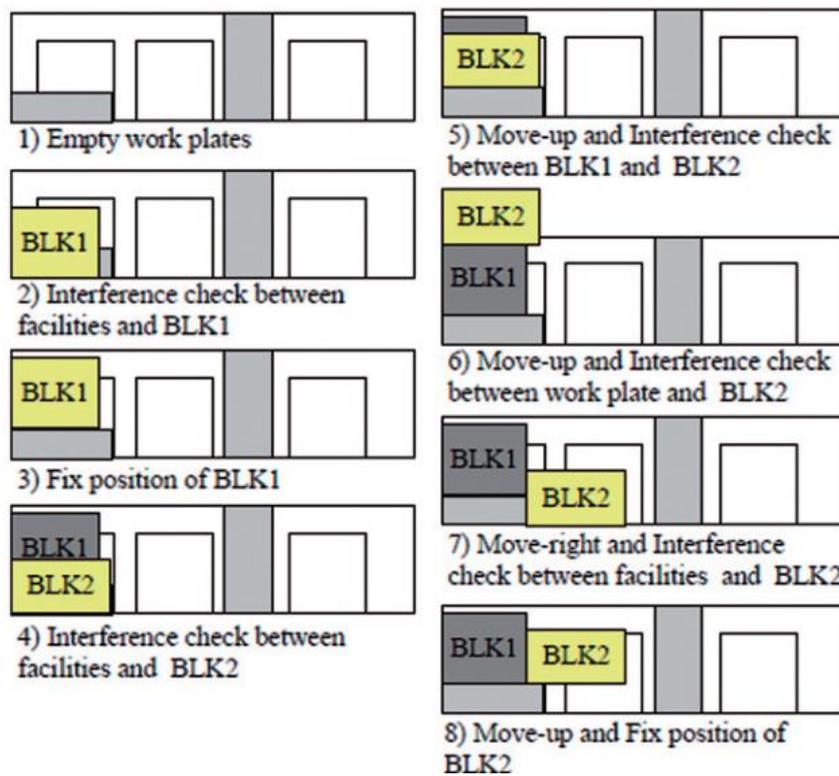


Figura 14. Proceso de comprobación de interferencias (Fuente: J.G. Shin, O.H. Kwon, C. Ryu, 2008).

Posteriormente, B. Kwon y G.M. Lee con su artículo *Spatial scheduling for large assembly blocks in shipbuilding* (2015), presentan un algoritmo heurístico también en dos etapas, una consistiendo en reglas de prioridad para despachar los bloques y la segunda siendo una modificación del método BLF al que han nombrado como *diagonal fill* (DL). En este se consideran cuestiones como la rotación de los bloques, el equilibrado de la ocupación de las diferentes áreas de trabajo, la facilidad de elaborar modificaciones ante situaciones imprevistas, etc. En la primera etapa de este algoritmo tienen en cuenta el tiempo mínimo y el tiempo máximo de procesamiento, de forma que en este caso han decidido que el tamaño de los bloques no tenga relevancia en el orden de asignación. En la segunda etapa emplean el método DL para la ubicación de los bloques, el cual consiste en considerar tanto los vértices de abajo a la izquierda como los de arriba a la derecha. Como los bloques van entrando y saliendo del parque continuamente por sus diferentes tiempos de procesamiento se puede generar la situación en la que la ubicación de un nuevo bloque sea retrasada hasta que se resuelva una interferencia ocasionada por los bloques ya posicionados en el parque, por lo que este mecanismo supone una opción más eficiente en cuanto a la utilización del parque ya que coloca los bloques lo antes posible. Este hecho se observa en la siguiente figura.

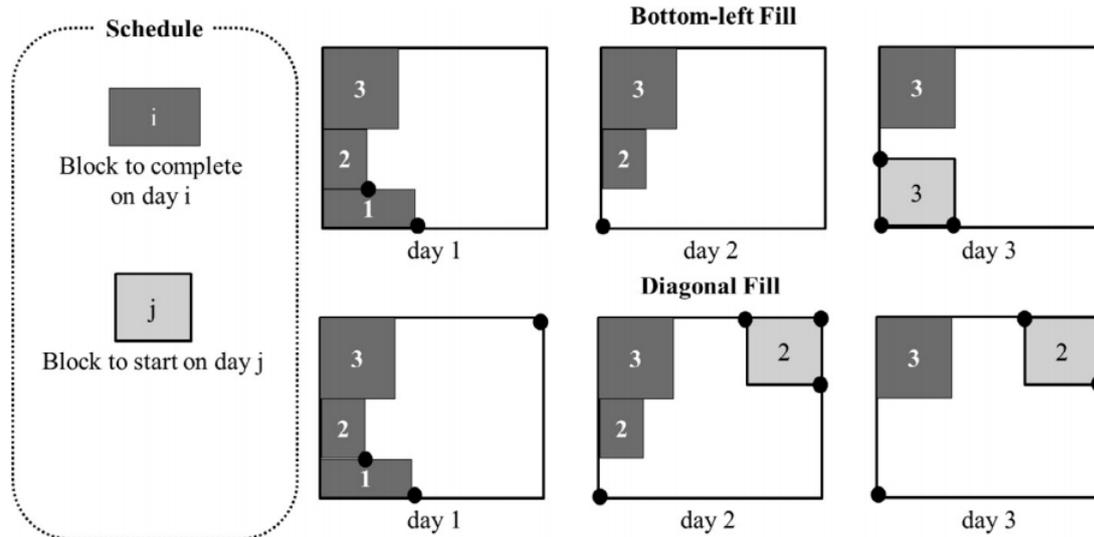


Figura 15. Comparación entre el método bottom-left fill y diagonal fill (Fuente: B. Kwon y G. M. Lee, 2015).

Uno de los grandes inconvenientes que puede presentar el mecanismo DL es la aparición de espacios inutilizables, para lo que desarrollan otra estrategia que minimice este factor. Esta consistirá en realizar una comprobación del tamaño que tendrá el espacio inaprovechable combinando una rotación alternativa del bloque y la posición en puntos de localización de los bloques vecinos.

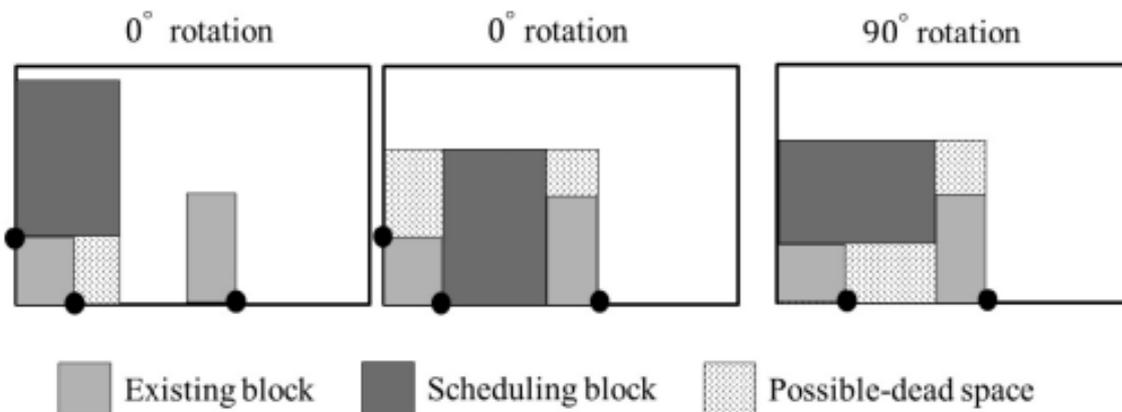


Figura 16. Ejemplo de estrategia de minimización del espacio inutilizable (Fuente: B. Kwon y G. M. Lee, 2015).

Resulta interesante también la manera en la que se puede tratar la dimensión temporal como se explica en algunas de estas publicaciones. En artículos como el primero que hemos mencionado del proyecto DAS o este último definen el tiempo como una dimensión espacial más, convirtiendo así un problema bidimensional y temporal en un problema de distribución tridimensional con ciertas restricciones. Con esta transformación, los rectángulos que representaban a los bloques serán extruidos, de forma que tomarán una cierta altura representando la duración del trabajo que recibirán el en parque. Se debe tener en cuenta que los bloques tienen una franja de tiempo en la que se puede realizar dicho trabajo, la cual viene dada por el tiempo

mínimo y el tiempo máximo de trabajo en parque, por lo que los bloques en el problema de espacio tridimensional tendrán un cierto intervalo del eje Z en el que podrán ser posicionados.

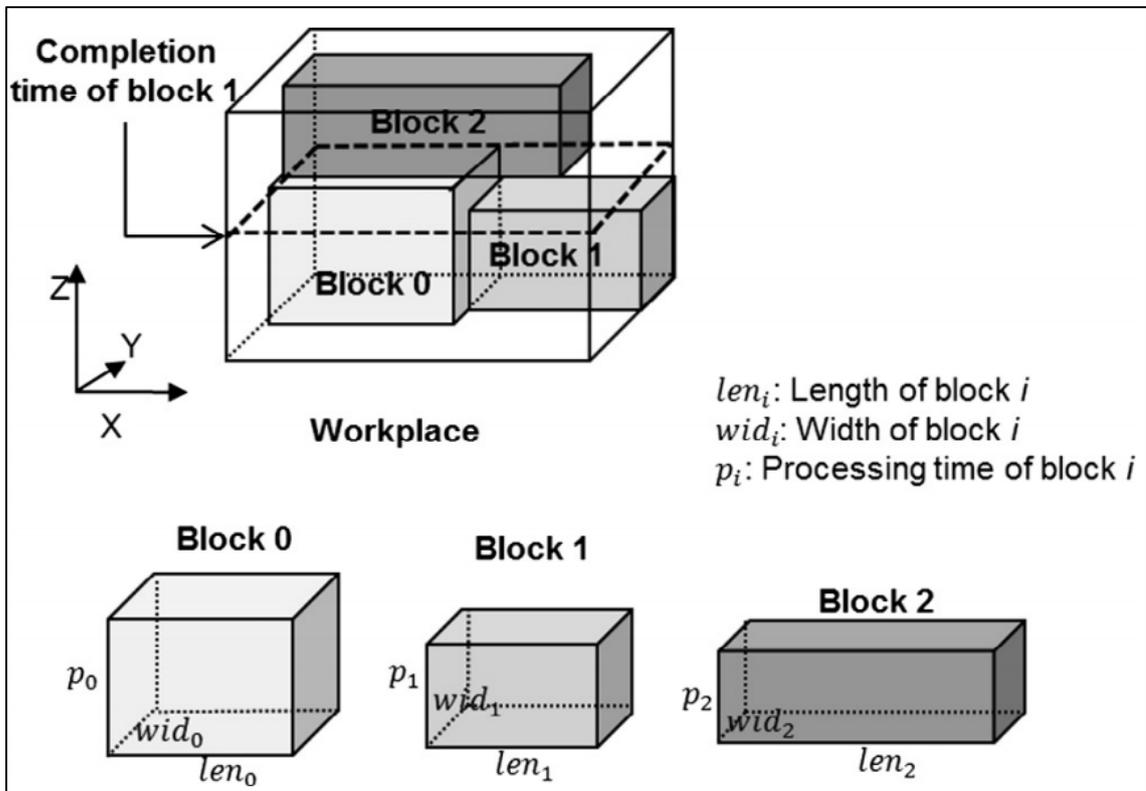


Figura 17. Representación tridimensional de la programación de bloques (Fuente: B. Kwon y G. M. Lee, 2015).

Como hemos visto, en estos artículos se trata siempre de encontrar primero una secuenciación para la asignación de los bloques que posteriormente se ubicarán uno a uno empleando diferentes reglas. En este trabajo se desarrolla un algoritmo que difiere respecto de estas otras heurísticas que priorizan la selección de bloques, de forma que se prioriza la selección de ubicaciones para posteriormente posicionar los bloques. El método en el que se buscan estas localizaciones también se distingue de los anteriores, ya que en este nos basaremos en áreas de localización, las cuales se derivan de la incorporación de nuevos bloques a la planificación existente.

5 METODOLOGÍA

5.1 Descripción del problema y del proceso

Como ya se ha expuesto en el inicio del trabajo, el objetivo que debe cumplir el algoritmo de planificación es la minimización del tiempo de construcción de buque, o lo que es lo mismo, la minimización del *makespan*. Para ello, el algoritmo trabaja con dos elementos: el parque de almacenamiento y los bloques con los que se formará el barco.

Comenzando con la definición del parque, este contará con una geometría rectangular, cuyas dimensiones serán **W** para el ancho y **L** para el largo. En concreto, en el caso que se va a tratar, las dimensiones del parque de bloques serán 90 x 35 metros. Para este, se define un sistema de coordenadas de forma que el origen de este sea el vértice inferior izquierdo, orientando el parque en la manera en que la arista mayor sea paralela al eje X y la arista menor sea paralela al eje Y. Por lo tanto, los vértices de dicho parque se situarán en los siguientes puntos: (0, 0), (L, 0), (0, W) y (L, W).

En cuanto a los bloques, se cuenta con un conjunto **B** de **N** bloques, el cual se recoge en la tabla del Anexo I:

$$B = \{b_i; i = 1, \dots, N\}$$

Para cada uno de los bloques se dispone de sus dimensiones espaciales, de forma que para un bloque **i** tenemos su longitud **l_i** y su ancho **w_i**; además se cuenta con la duración de trabajo en el parque **d_i** y el tiempo a partir del cual se dispone de estos bloques para comenzar con su trabajo en el parque **tr_i**.

El emplazamiento en el parque de un bloque **i**, vendrá dado por el punto (**x_i**, **y_i**), el cual representa las coordenadas de vértice del bloque más cercano al origen. Respecto a la orientación, las aristas del bloque deberán mantenerse paralelas a las del parque, por lo que únicamente se plantean dos orientaciones posibles para cada uno de los bloques, una con la mayor arista del bloque paralela al eje X y otra con la mayor arista del bloque paralela al eje Y. La colocación del bloque en una orientación u otra vendrá dada por la lógica del algoritmo.

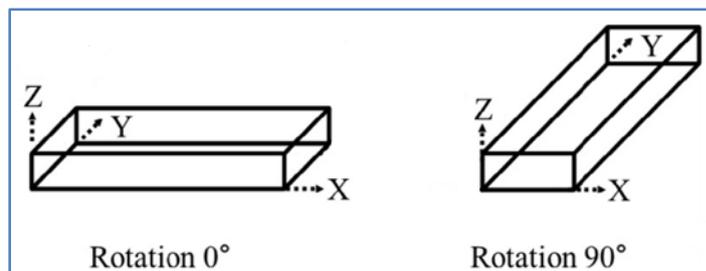


Figura 18. Orientaciones posibles de los bloques en este algoritmo (Fuente: B. Kwon y G. M. Lee, 2015).

En la siguiente figura se representa la disposición de un bloque en el parque y las variables empleadas para modelar las dimensiones geométricas del problema. La orientación del bloque puede ser cambiada durante la solución del problema, intercambiándose así los lados de dimensiones l_i y w_i . En la ubicación real de estos bloques se deberá dejar un cierto espacio de separación entre los mismos y entre los límites del parque, por lo que en la creación de los bloques en el modelo se les añadirán dos metros tanto en el ancho como en el largo, con el fin de que cada bloque tenga un metro de margen en cada uno de sus lados.

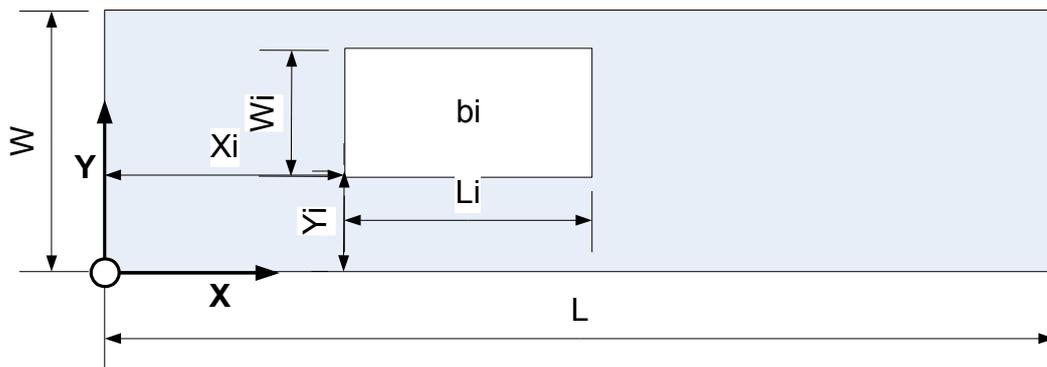


Figura 19. Variables geométricas del problema (Fuente: D. Crespo).

Cuando se comienza el procesamiento de un bloque, este es indicado en la variable t_i , por lo que el bloque se mantendrá en el parque entre los instantes t_i y $t_i + d_i$. Este instante t_i deberá estar comprendido entre el tiempo mínimo y máximo de inicio de trabajo en el parque que proporciona para cada bloque.

Con todo esto, podemos describir el problema de optimización, cuya función objetivo es la minimización del *makespan*, de la siguiente forma:

$$\min \left(\max_i (t_i + d_i) \right) \text{ sujeto a:}$$

$$b_i \cap b_j = \emptyset \quad \forall i \neq j$$

Donde las variables de decisión son los tiempos t_i de inicio de procesamiento de cada uno de los bloques, las coordenadas (x_i, y_i) que definen la posición de estos en el parque y la orientación que tendrá. La minimización del *makespan* está sujeta a una restricción que indica la prohibición de que se produzcan solapes espaciales entre distintos bloques un instante de tiempo.

Con el fin de desarrollar una heurística cuyas restricciones y suposiciones resulten en un mecanismo eficiente, es importante realizar un análisis de las características específicas del problema o subconjuntos de casos de este que, a pesar de reducir a posibilidad de aplicación del algoritmo a otros casos, nos permiten mejorar la eficiencia en los casos objetivo. En el caso de estudio, se observa que las dimensiones de los parques de almacenamiento habitualmente cuentan con una longitud significativamente mayor al ancho, de forma que la colocación de más de dos bloques en la dirección del ancho del parque es poco probable debido a las grandes dimensiones de los mismos, por lo que se pueden reducir las posibilidades de ubicación a la disposición de los bloques sobre las aristas de mayor longitud del parque, haciendo coincidir una arista de los bloques con la correspondiente del parque.

Para la identificación de las posibles ubicaciones presentes en el parque se empleará un método basado en áreas de localización, las cuales serán particiones tanto del espacio del parque de almacenamiento como del tiempo que se dispone de este. La geometría de estas áreas es rectangular, cuya posición vendrá definida por un punto de referencia llamado punto de localización. Este punto, en concordancia con la reducción del problema expuesta anteriormente, estará situado sobre una de las aristas longitudinales del problema. De esta forma, en función de la arista en la que se sitúen estos puntos distinguimos entre puntos de localización inferior y puntos de localización superior según estén situados en la arista inferior o superior del parque respecto al eje de coordenadas.

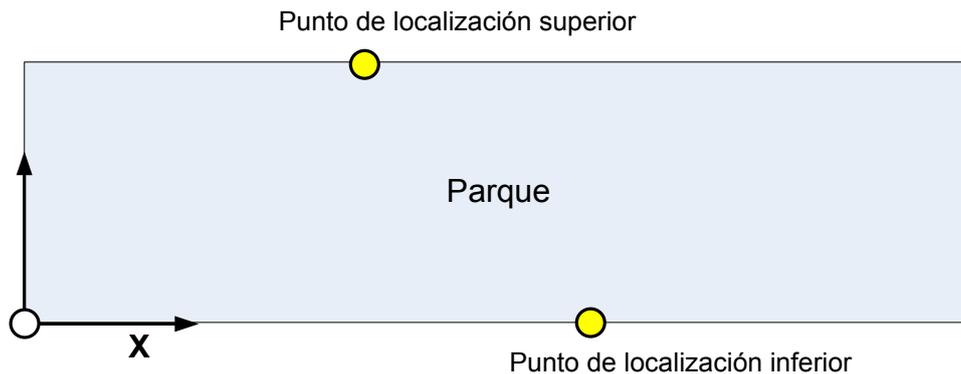


Figura 20. Clasificación de los puntos de localización (Fuente: D. Crespo).

El algoritmo desarrollado trabajará entonces con los siguientes seis parámetros que describen las áreas de localización: las coordenadas del punto de localización (lx_j , ly_j) con $j = 1, \dots, M$, siendo M el número de áreas de localización; la longitud mx_j , siendo esta la dimensión en la dirección X del sistema de coordenadas coincidiendo con la arista longitudinal del parque; el ancho my_j , siguiendo la dirección del eje Y y la del ancho del parque.

Estos parámetros que definen a cada área de localización serán actualizados a medida que se vaya ubicando nuevos bloques en el parque. De esta manera, se evitará que se produzcan colisiones entre bloques a la hora de realizar una nueva asignación.

Al realizarse la asignación de un bloque i a un área de localización j , el bloque se ubicará con un vértice coincidiendo con el punto de localización del área, de forma que los parámetros del bloque se asignarán tal que así:

$$x_i = lx_j$$
$$y_i = ly_j$$

Tras haber presentado las variables con las que trabajará el algoritmo, explicaremos el procedimiento secuencial en el que se basa, tratándose por tanto de una heurística de tipo constructivo. Este mecanismo no siempre alcanza soluciones óptimas, pero sí aporta una gran rapidez en la búsqueda de soluciones adecuadas.

La secuencia de pasos que sigue el algoritmo es la siguiente:

1. Todos los bloques del problema se insertan en la lista de bloques pendientes de asignación LO .
2. Se genera la primera área de localización con el punto de localización en el origen de coordenadas, cubriendo toda el área del parque y con tiempo cero. Se inserta en la lista de áreas de localización LA .
3. De entre las áreas presentes en LA , se selecciona una de ellas de acuerdo con la *regla de localización*.
4. Dada el área de localización seleccionada, se busca en LO un bloque cuya asignación sea factible respetando las restricciones del problema y de acuerdo con la *regla de selección de bloques*.
5. Si no es posible hallar ningún bloque asignable al área seleccionada, se elimina dicha área de la lista y se vuelve al paso 3, a menos que la lista de bloques esté vacía, en cuyo caso se ha completado la ejecución del algoritmo.
6. El bloque seleccionado se asigna al área de localización y por tanto pasa a formar parte de la solución del problema. Se elimina de LO . El área de localización se elimina de LA .
7. Si no quedan bloques en LO se ha completado la resolución del problema y se finaliza el algoritmo.
8. Se actualizan las áreas de localización existentes en LA comprobando posibles colisiones con el nuevo bloque asignado de acuerdo a la *regla de actualización de áreas*.
9. Se derivan nuevas áreas de localización a partir del bloque asignado de acuerdo con la *regla de derivación de áreas*. Si se produce la salida de algún bloque, se deriva un área en el hueco. Estas una vez creadas se añaden a LA y se verifican colisiones con los bloques ya asignados a la solución mediante de nuevo la *regla de actualización de áreas*.
10. Volver al paso 3.

Las reglas mencionadas en los pasos que sigue este algoritmo se explican a continuación.

5.1.1 Regla de localización

Las áreas de localización se priorizan según las siguientes pautas:

1. Menor valor de lx_j . Para priorizar la asignación de bloques próximos al origen de coordenadas y lograr una mayor compacidad de la solución.
2. Menor valor de ly_j . El objetivo de esto es mantener una dirección de ubicación preferente en la arista inferior del parque. Combinada con la regla anterior, se crea una alternancia entre asignaciones sucesivas de bloques en las dos aristas posibles del parque.

5.1.2 Regla de selección de bloques

Los bloques se priorizan siguiendo las reglas expuestas a continuación:

1. Mayor valor de su dimensión en el eje Y. En principio se comprueba si es posible la asignación con el lado longitudinal (de dimensión l_i) en la dirección del eje Y. Si no fuera así, se probaría con el otro lado (de dimensión w_i).
2. Menor duración del bloque en el parque.

El motivo de esto, en conjunto con las reglas de localización, es lograr una gran compacidad de almacenamiento, comenzando en uno de los vértices del parque con los bloques de mayores dimensiones, los cuales ocupan prácticamente toda el ancho del parque impidiendo la colocación de otro bloque de forma transversal, continuando con bloques cada vez menores en la misma arista del parque, que se van complementando con los bloques de menores dimensiones que caben en la arista opuesta a la que contiene los mayores bloques. De esta manera, a medida que se va ejecutando el algoritmo, los bloques se sitúan de forma complementaria, en aristas opuestas, se van aproximando más a un tamaño similar.

5.1.3 Regla de actualización de áreas de localización

La regla de actualización de áreas se pondrá en práctica cuando se invada el espacio definido para un área de localización debido a la ubicación de un nuevo bloque en el parque de almacenamiento. Si esto ocurre, se deberá realizar una actualización de los parámetros del área cumpliendo los siguientes pasos:

1. Primero se comprueba si la colisión se produce en dirección del eje Y debido a tratarse de un bloque ubicado en el lado opuesto al área de localización. Se reduce el margen de dicho eje my_j según sea necesario para garantizar que futuras asignaciones al área sean factibles y si este margen fuese inferior a la mínima dimensión de cualquier bloque ($\min_{b_i \in LO} w_i, l_i$), se elimina.

2. Por último, se comprueba si la colisión se produce en dirección del eje X por ser un bloque en el mismo lado que el área de localización. Se reduce el margen de dicho eje mx_j según sea necesario para garantizar que futuras asignaciones al área sean factibles y si este margen fuese inferior a la mínima dimensión de cualquier bloque ($\min_{b_i \in LO} w_i, l_i$), se elimina.

5.1.4 Regla de derivación de áreas de localización

Con el proceso expuesto anteriormente las únicas modificaciones que se pueden realizar sobre las áreas son su reducción o eliminación, por lo que será imprescindible la definición de una regla que nos permita crear nuevas áreas a medida que los bloques van consumiendo las áreas en las que han sido colocados.

Estas nuevas áreas se originan en las proximidades del nuevo bloque asignado, en los bordes libres de contacto del bloque, tanto en el mismo lado como en el lado opuesto del parque. Los pasos a seguir para la derivación de localizaciones son los siguientes:

1. Se añade un área de localización en el mismo lado que el bloque pero con el punto de localización con coordenada $lx_j = x_i + l_i$ (ver figura 6). La generación de este tipo de puntos de localización origina un proceso de construcción de las soluciones de izquierda a derecha.
2. Se añade otra localización en el lado opuesto al bloque con el mismo valor de coordenada X. De esta forma se habilita la alternancia de lados en la construcción de las soluciones.

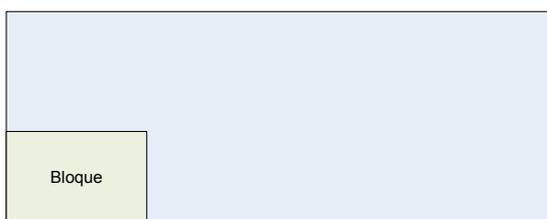


Figura 21. Posición de asignación del bloque (Fuente: D. Crespo).

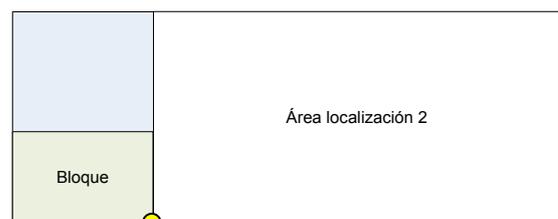


Figura 22. Localización derivada en el lado del bloque (Fuente: D. Crespo).

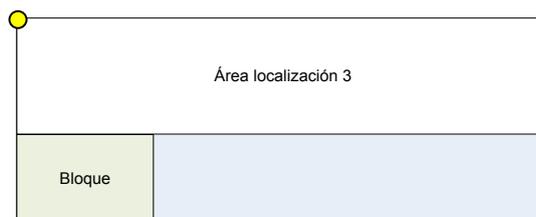


Figura 23. Localización derivada en el lado opuesto al bloque (Fuente: D. Crespo).

5.1.5 Regla de derivación de áreas por salida de bloque

Aplicando esta regla se creará un área en el espacio que deja un bloque al abandonar el parque, tras haber completado todos los procesos que se llevan a cabo para su construcción.

1. Se comprueba, para los bloques **i** situados en la misma arista que el que ha salido (bloque **s**), si $x_s \geq x_i + l_i$, si esto se cumple, se realiza la siguiente asignación $lx_j = \max_i(x_i + l_i)$, obteniendo así el punto de localización para la nueva área.
2. A continuación, para los bloques situados en la misma arista se comprueba si $x_s + l_s \leq x_i$, si esto se cumple, se realiza la siguiente asignación para obtener el extremo derecho del área $fx_j = \min_i(x_i)$, pudiendo calcular así el tamaño del área como: $mx_j = fx_j - lx_j$.

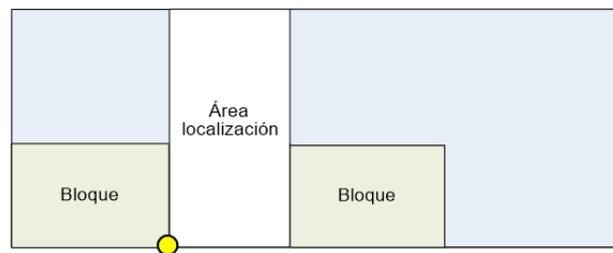


Figura 24. Localización derivada en el lado opuesto al bloque (Fuente: elaboración propia).

5.2 Modelo y simulación

Los modelos virtuales y la simulación de eventos discretos constituyen uno de los componentes principales de la transformación hacia la industrial 4.0. El uso de *software* de simulación como FlexSim nos permite ensayar y visualizar múltiples escenarios posibles, de lo que se puede extraer información para optimizar un proceso y realizar una planificación adecuada en numerosos aspectos de la industria.

En este trabajo, la simulación nos permitirá comprobar las soluciones obtenidas con la heurística que se ha desarrollado para la ubicación de bloques en el parque de almacenamiento. Por ello, se ha creado un modelo que representa dicho parque y el funcionamiento del mismo. La herramienta utilizada tanto para la creación del modelo como para su posterior simulación ha sido el software ya mencionado, FlexSim.

Anteriormente, se ha visto que una de las diferencias del método de planificación que se propone frente a los de otros autores es la secuenciación dinámica de los bloques, evitando así metodologías en las que primero se hace un ordenamiento de los bloques para posteriormente buscarles un emplazamiento uno a uno. Por este motivo la simulación es de vital importancia, ya que nos permite ver la capacidad del sistema en función de la cartera de pedidos estimada. De esta forma, podremos estudiar el impacto de las reglas de asignación a largo plazo en los niveles de

ocupación y con ello determinar problemas de saturación de la capacidad debidos al acumulamiento de órdenes de trabajo, junto con los retrasos que un algoritmo deficiente pueda inducir.

5.2.1 Software de modelado y simulación, FlexSim

El programa Flexsim es un software de simulación de eventos discretos que se emplea para modelar, visualizar, analizar y optimizar cualquier proceso industrial, desde procesos de fabricación hasta cadenas de suministro. Este, a diferencia de algunos de los *softwares* de la competencia, nos permite construir y ejecutar el modelo en un entorno tridimensional desde el inicio. Por las ventajas que ofrece, es uno de los programas más empleados en entornos industriales, contando con una gran comunidad de desarrolladores que mejoran las competencias del software. Además, se dispone de una gran multitud de herramientas adicionales a este, como modelos 3D y librerías, y existe un foro donde es posible manifestar dudas sobre el modelado y simulación con FlexSim que frecuentemente son resueltas con eficacia.

En el entorno de este programa, se construyen los modelos como sistemas de flujo de entidades (*Flowitems*) creados generalmente en una fuente (*Source*), a través de colas (*Queues*), procesadores (*Processor*) y sistemas de transporte (*Task Executors* o *Conveyors*), finalizando en un sumidero (*Sink*).

- **Fuente (*Source*).** Este es el primer elemento que suele aparecer en un modelo, siendo el que nos permite modelar la llegada de *items* a nuestro modelo según tablas, distribuciones, secuencias, etc. En este elemento podemos definir varios aspectos para los *items*, como por ejemplo de una botella si queremos representar un sistema de embotellado real.

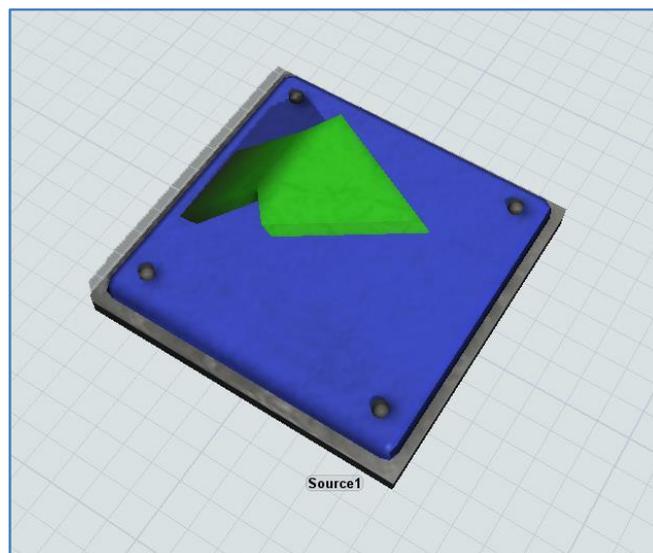


Figura 25. Modelo 3D por defecto del elemento Source en FlexSim (Fuente: Apuntes de simulación de sistemas logísticos).

- **Cola (Queue).** El elemento cola nos permite almacenar los *items* cuando el proceso está bloqueado aguas abajo. Por defecto tienen un funcionamiento tipo FIFO (*First In, First Out*), aunque este método de despacho puede ser configurado posteriormente. Para las colas podemos definir una capacidad, modificar la posición en la que se almacenan los *items*. También se puede definir el comportamiento de estas colas para que trabajen por lotes (*Batching*).

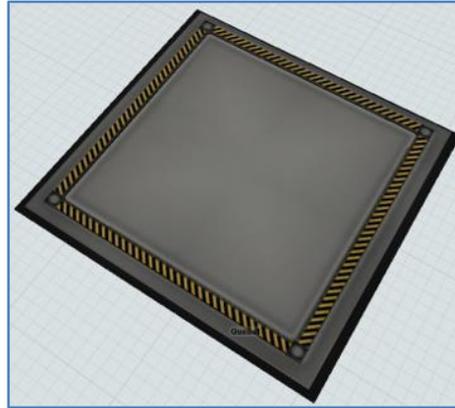


Figura 26. Modelo 3D por defecto del elemento Queue en FlexSim (Fuente: Apuntes de simulación de sistemas logísticos).

- **Procesador (Processor).** Este elemento se emplea para representar los distintos procesos que atraviesa el *item* real. Se trata de una simulación en cuanto al tiempo que pasa un *item* en dicho proceso, por lo que los procesadores retendrán el *item* hasta que pase el tiempo indicado en modelo. Este tiempo de retención podrá seguir diferentes pautas, igual que en el elemento fuente, ya sea en función de una tabla, una distribución, el tipo de *item*, etc. Para este, se puede definir también la capacidad de procesamiento de *items* simultáneamente. Además, es posible simular la intervención de un operario en el proceso, haciendo que la máquina solicite un trabajador cuando se va a procesar un *item*.

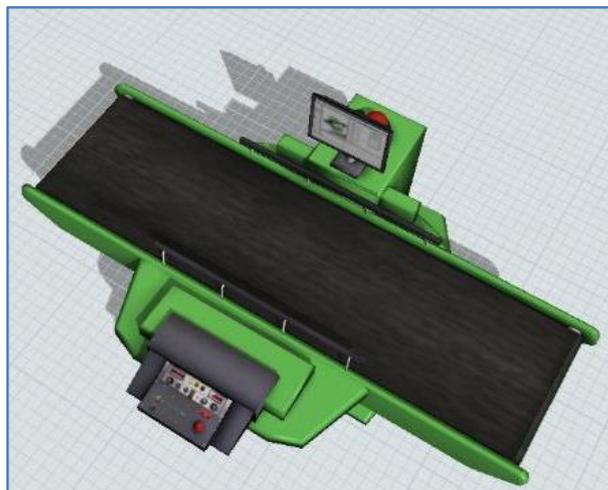


Figura 27. Modelo 3D por defecto del elemento Processor en FlexSim (Fuente: Apuntes de simulación de sistemas logísticos).

- **Sistemas de transporte (*Task Executors* o *Conveyors*).** Estos se utilizan para el simular el desplazamiento entre los elementos que hemos presentado. Aunque en algunos modelos se puede prescindir de ellos, lo más habitual es que se incluyan en todos los modelos. Estos pueden ser de diferentes móviles (*task executors*) o fijos (*conveyors*). En el primer grupo se encuentran los operarios, carretillas, AGVs, grúas, etc., y en el segundo los transportadores de cintas o rodillos. En ambos grupos se ofrecen diversos parámetros de configuración como las velocidades de desplazamiento, los tiempos de carga y descarga, la capacidad, lógicas para el desplazamiento, entre otros.



Figura 28. Ejemplo del modelo 3D por defecto del task executor Transporter en FlexSim (Fuente: elaboración propia).

- **Sumidero (*Sink*).** Su función es representar el final de la línea de fabricación. Al llegar a este los *items* son destruidos, por lo que para la obtención de algún dato de estos es necesario realizarla previamente o configurándolo en el momento de entrada de los *items* a este elemento.

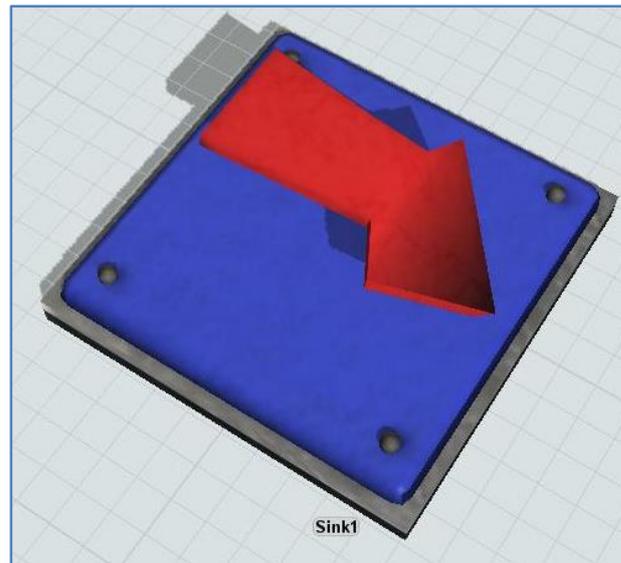


Figura 29. Modelo 3D por defecto del elemento Sink en FlexSim (Fuente: Apuntes de simulación de sistemas logísticos).

En el caso que vamos a tratar, los *flowitems* del modelo serían los elementos que representan a los bloques del barco. Sin embargo, en este modelo, por la complejidad del comportamiento de los elementos del sistema, no es suficiente como emplear los elementos básicos que ofrece FlexSim, sino que será necesario emplear elementos cuyo comportamiento ofrezca una mayor libertad de configuración. FlexSim ofrece un entorno sencillo que permite crear lógicas sin emplear un lenguaje estructurado, aunque sí existe esta posibilidad, que será aplicada en el modelo elaborado para este trabajo.

En el entorno del *process flow*, se nos permite “programar” por medio de diagramas de flujo, utilizando bloques que tienen funciones predeterminadas y que pueden interactuar con el modelo. Estos bloques se ejecutarán cuando sean atravesados por un elemento llamado *token*, el cual puede tener etiquetas con diferente información y referencias a los elementos del modelo. Algunos ejemplos de funciones de dichos bloques son: creación y destrucción de *tokens*, creación y destrucción de *items* en el modelo, desplazamiento y redimensionado de *items* en el modelo, inserción de etiquetas en los *tokens*, ejecución de *scripts*, entre otras.

Los *process flow* se pueden emplear tanto para definir comportamientos generales en todo el modelo como para definir el comportamiento de elementos customizados, para los que el empleo de los elementos básicos como *Processors* o *Queues* es insuficiente. Estos elementos customizados aparecen en FlexSim como *BasicFR*, al cual se le pueden dar diferentes aspectos para ajustarse mejor a lo que se quiere representar con él. El empleo de los *process flow*, tanto generales como en elementos customizados, además de la incorporación de algunos scripts en ellos nos permitirán modelar el comportamiento del parque de bloques implementando la heurística que se ha presentado.

Liberar Posiciones

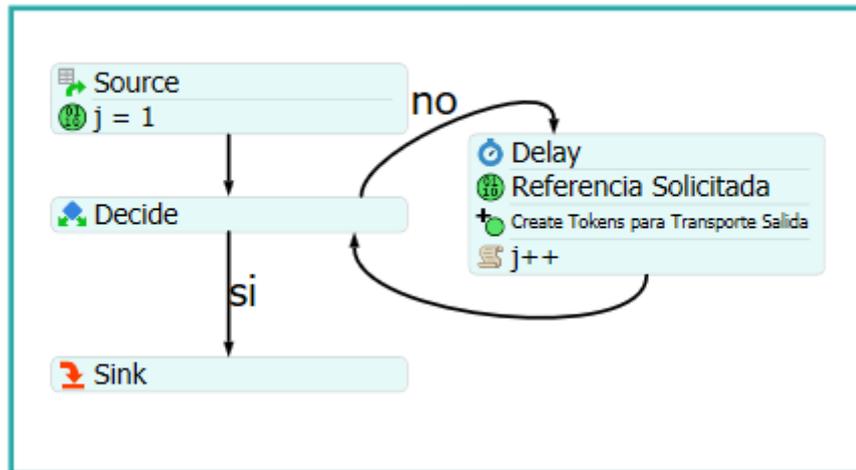


Figura 30. Ejemplo de Process Flow (Fuente: elaboración propia).

Otra de las funcionalidades de este programa es que permite configurar una ventana (*dashboard*) en la que se muestren los datos de salida de las simulaciones ejecutadas, dando varias opciones para la visualización. Con este podremos extraer fácilmente información como la ocupación de un trabajador o un proceso, el tiempo de ciclo, los *items* que atravesaron un cierto elemento, el número máximo de *items* que se acumularon en una cola, valores de variables utilizadas en el modelo, entre otros. Esto nos permitirá conocer los aspectos más vulnerables del sistema como dónde se crean los cuellos de botella, dónde está sobredimensionado el proceso, etc.

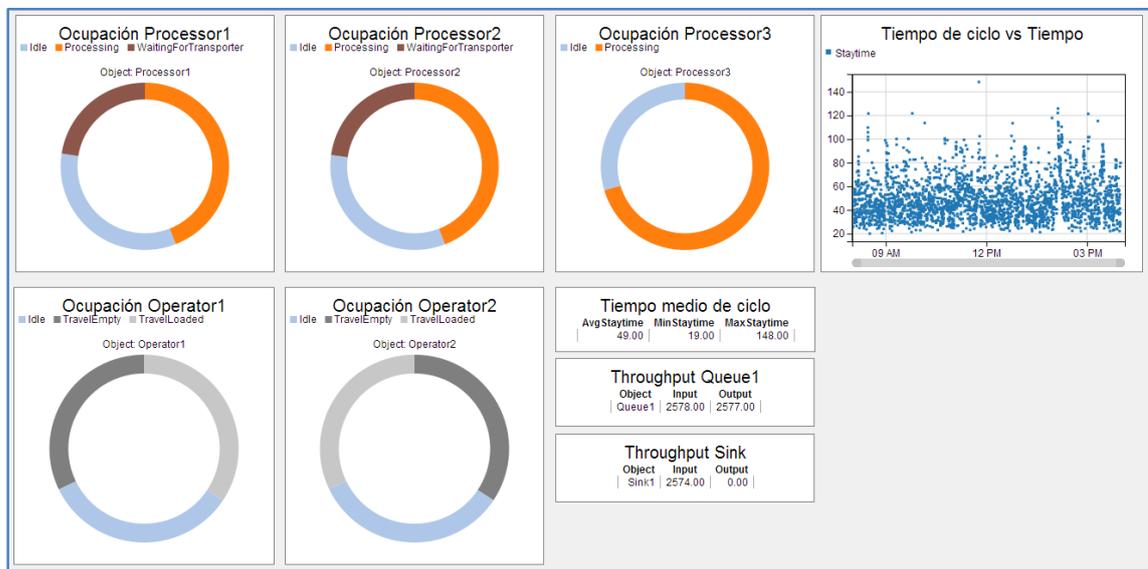


Figura 31. Ejemplo de representación de los datos de salida de la simulación (Fuente: Apuntes de simulación de sistemas logísticos).

Por último, cabe destacar la herramienta *Experimenter* con la que cuenta FlexSim. En esta podemos configurar varios escenarios con diferentes condiciones, como puede ser un número diferente de máquinas, una mayor o menor llegada de pedidos o cualquier condición de la que queramos observar las consecuencias en el sistema. Con ello, la labor de experimentación se realiza de una manera más sencilla y rápida, al poder ejecutar de una vez los diferentes escenarios previamente configurados. Además, por la variabilidad existente en la mayoría de procesos, se nos permite definir las veces que queremos que se ejecute cada caso, con el fin de obtener unos resultados promediados.

5.2.2 Modelo del parque de bloques

El modelo de simulación creado es el que aparece en la siguiente figura, en la que se representa el parque en un instante de tiempo en el que ya cuenta con bloques ubicados. El funcionamiento de este, con la correspondiente implementación del algoritmo será explicado a continuación.



Figura 32. Modelo del parque de almacenamiento de bloques (Fuente: elaboración propia).

5.2.2.1 Creación de bloques en el modelo

Primero, para obtener la información de los bloques en el modelo se cuenta con una tabla en el modelo llamada "Referencias" que recoge los siguientes datos de cada uno de los bloques: la referencia, las dimensiones, la duración del trabajo en el parque y el tiempo en el que están disponibles. Para la generación de los bloques en modelo, en lugar de emplear el bloque básico *source* se ha implementado una lógica general para crearlos directamente en una cola (*queue*) del modelo según una tabla que contiene la información de todos ellos.

Generar Stock Inicial

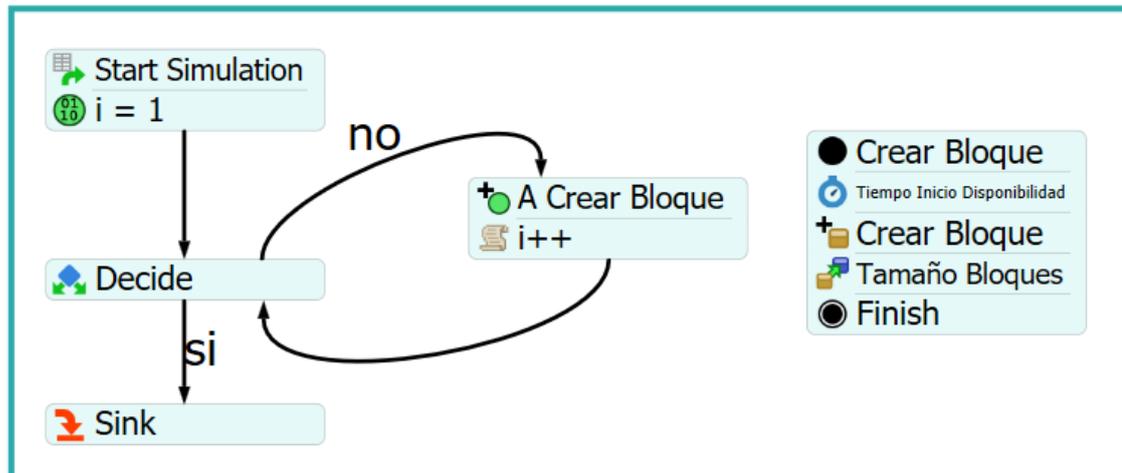


Figura 33. Diagrama de flujo para la generación de los bloques en el modelo (Fuente: elaboración propia).

En este se recorren todas las entradas de la tabla que contiene la información de los bloques. Para cada entrada, se genera un *token* que recorrerá las actividades de la derecha. La primera de ellas retiene el acceso de los *tokens* a las siguientes actividades. Este tiempo de retención dependerá de la información que se recoge en la tabla “Referencias” para el tiempo de disponibilidad en el parque que a su vez depende del navío al que pertenece el bloque.

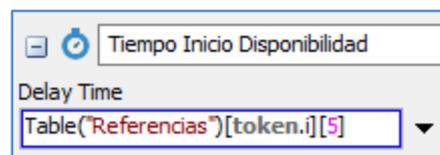


Figura 34. Configuración actividad "Tiempo Inicio Disponibilidad" (Fuente: elaboración propia)

Tras esto, el *token* atraviesa una actividad que crea un objeto bloque genérico en un elemento *queue* inicial en el modelo. En este objeto se insertan las siguientes etiquetas, cuya información proviene también de la tabla “Referencias”:

- **ReferenciaBloque:** contiene el código de referencia del bloque.
- **DuracionBloque:** contiene el tiempo que dura el trabajo sobre el bloque.
- **SizeBloqueX:** contiene lado de menor dimensión del bloque más el margen de separación.
- **SizeBloqueY:** contiene lado de mayor dimensión del bloque más el margen de separación.

Object	Box
Quantity	1
Create In	<input checked="" type="radio"/>
Create At	<input type="radio"/>
StockInicial	X
Assign To	<input checked="" type="radio"/>
Insert at Front of	<input type="radio"/>
token.item	
Assign Labels to Created Objects	
Name	ReferenciaBloque
Value	Table("Referencias")[token.i][1]
Name	DuracionBloque
Value	Table("Referencias")[token.i][4]
Name	SizeBloqueX
Value	Comprobar Lado Menor
Name	SizeBloqueY
Value	Comprobar Lado Mayor

Figura 35. Configuración de la actividad “Crear Bloque” (Fuente: elaboración propia).

Los bloques se generarán siempre con una disposición vertical para facilitar procesos que se llevarán a cabo posteriormente en el *process flow*. Esto, se ha realizado mediante dos *scripts*, que leen en la tabla “Referencias” cual es el mayor lado del bloque y en función de esto devuelven el menor o el mayor, dependiendo si se va a asignar la etiqueta “SizeBloqueX” o “SizeBloqueY”. Además, se suma el margen de separación que es necesario en la ubicación real de los bloques, cuyo valor está contenido en una variable global de FlexSim que se ha creado previamente, llamada “MargenBloques”. Los correspondientes códigos, basados en c++, son los siguientes:

- Código “SizeBloqueX”:

```
/**Comprobar Lado Menor*/  
Object current = param(1);  
treenode activity = param(2);  
Token token = param(3);  
Object item = param(4);  
Variant assignTo = item;  
int createdrank = param(5);  
string labelName = param(6);  
treenode processFlow = ownerobject(activity);  
  
if (Table("Referencias")[token.i][2] > Table("Referencias")[token.i][3])  
{
```

```
        return Table("Referencias")[token.i][3] + MargenBloques;
    }
    else
    {
        return Table("Referencias")[token.i][2] + MargenBloques;
    }
}
```

- Código “SizeBloqueY”:

```
/**Comprobar Lado Mayor*/
Object current = param(1);
treenode activity = param(2);
Token token = param(3);
Object item = param(4);
Variant assignTo = item;
int createdrank = param(5);
string labelName = param(6);
treenode processFlow = ownerobject(activity);

if (Table("Referencias")[token.i][2] > Table("Referencias")[token.i][3])
{
    return Table("Referencias")[token.i][2] + MargenBloques;
}
else
{
    return Table("Referencias")[token.i][3] + MargenBloques;
}
```

Los bloques creados tienen todas unas dimensiones genéricas en el modelo, ya que con lo anterior únicamente se asignan etiquetas con los valores de las dimensiones, pero no de les ha dado ese tamaño. Para esto, los *tokens* que atraviesan este diagrama, donde cada uno contiene una referencia a los bloques que se han creado en el modelo, pasan a la actividad “Tamaño Bloques”, en la que los bloques serán redimensionados para ajustarse a los valores de tamaño que aparecen en las etiquetas que les hemos asignado.

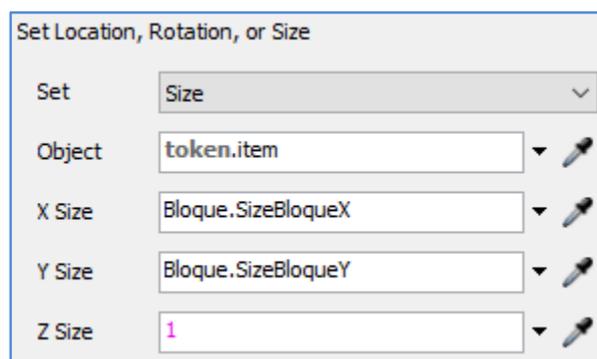


Figura 36. Configuración de la actividad “Crear Bloque” (Fuente: elaboración propia).

Una vez se han creado todos los bloques, estos pasan a otra cola en la que esperan a ser posicionados. La referencia a cada uno de los elementos bloque se guarda en una lista de FlexSim llamada “BloquesAPosicionar” con un *trigger* que se activa al entrar el bloque en esta segunda cola, para poder filtrarlos y ubicarlos posteriormente siguiendo la regla de selección de bloques de la heurística diseñada.

Estas dos colas son elementos auxiliares para el modelo, no representan espacios reales del astillero.

5.2.2.2 Implementación de la heurística

Como ya se ha mencionado anteriormente, para representar el almacén se ha empleado un elemento customizable *BasicFR*, el cual se dimensionará de acuerdo con la longitud y ancho del parque real, 90 x 35 m. La lógica que debe seguir se ha modelado utilizando un *process flow* propio para el elemento. En este, además de emplear las funciones predeterminadas que proporciona FlexSim en este entorno, es necesario programar algunas funciones empleando código. El lenguaje de programación que se emplea FlexSim se basa en c++.

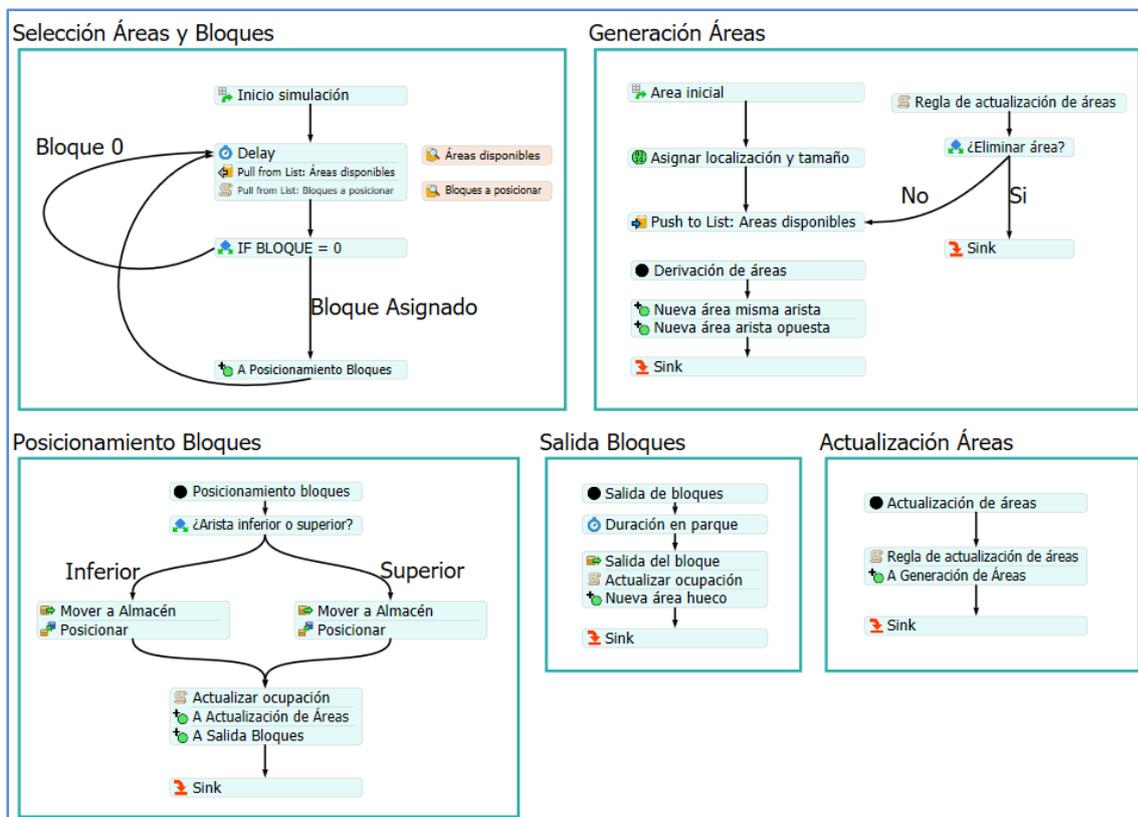


Figura 37. Process flow que modela el comportamiento del almacén según la heurística diseñada (Fuente: Elaboración propia).

Las áreas de localización no tienen un elemento de representación en el modelo en sí, sino que se representan en este *process flow* mediante *tokens*, los cuales cuentan en sus etiquetas con las características del área que representan. En el inicio de la simulación, en la actividad “Área inicial” de la sección “Generación Áreas” se crea un *token* que representará el primer área, que cuyas dimensiones abarcan todo el parque. Las etiquetas que definen el área se asignan en la siguiente actividad, las cuales son:

- **IDArea:** contiene un identificador único, asignándose mediante una variable global “ContadorAreas” que incrementa en uno cada vez que se genera un área nueva.

- **PosicionAreaX**: contiene la coordenada X del punto de localización del área.
- **PosicionAreaY**: contiene la coordenada Y del punto de localización del área.
- **SizeAreaX**: contiene la longitud del área.
- **SizeAreaY**: contiene la anchura del área.
- **Arista**: contiene la arista en la que se sitúa el área, inferior o superior.

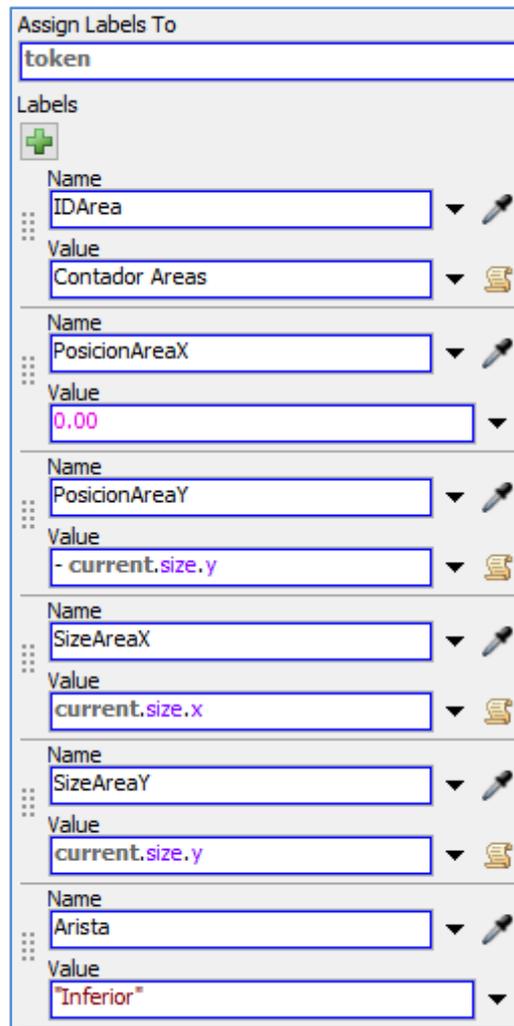


Figura 38. Configuración de la actividad “Asignar localización y tamaño” (Fuente: elaboración propia).

Cabe destacar que las características geométricas de esta área se asignan de forma dinámica en función de las dimensiones del parque (representado por el código `current`). De esta forma al variar el tamaño del parque en el modelo no es necesario realizar ningún cambio en el *process flow*, ya que el área siempre se crea según las dimensiones actuales del parque.

Tras esto, el *token* accede a una actividad que guarda al mismo *token* en una lista llamada "AreasDisponibles", dónde quedará a la espera de que se le asigne un bloque.

Por otro lado se crea un *token* en la sección llamada "Selección Áreas y Bloques" el cual recorrerá en bucle el diagrama en el que se encuentran implementadas la regla de localización y la regla de selección de bloques. Este atraviesa primero una actividad que retira un área de la lista "AreasDisponibles" según la regla definida en la heurística y la asigna en el *token* que recorre este bucle una etiqueta con la referencia al *token* que representa al área en cuestión llamada "Área". La manera en la que se aplica la regla es mediante un filtro (*query*) con la siguiente condición de retirada de *tokens*:

ORDER BY TiempoArea ASC, PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1

The image shows a configuration window for a 'Pull from List' activity. The 'List Reference' is set to 'Areas disponibles'. The 'Request Number' and 'Require Number' are both set to '1.00'. The 'Assign To' radio button is selected, and the field contains 'token.Area'. The 'Query / Object / Array' field contains 'ORDER BY PosicionAreaX ASC, PosicionAreaY'. The 'Partition ID' is set to 'None'. The 'Puller' is set to 'token'. At the bottom, there are several unchecked checkboxes: 'All or Nothing', 'Leave Entries On List', 'Use Max Wait Timer', 'Use Max Idle Timer', and 'Keep Back Order On Early Release'.

Figura 39. Configuración de la actividad "Pull from List: Areas disponibles" (Fuente: elaboración propia).

Tras esto, el *token* pasa a una actividad en la se realizará la selección del bloque. Por las características que se deben cumplir en esta regla, las opciones que nos ofrece el bloque predefinido para retirar elementos de listas no es suficiente, por ello esto se ha ejecutado mediante un *script* con el siguiente código:

```
/**Regla de selección de bloques*/  
Object current = param(1);  
treenode activity = param(2);  
Token token = param(3);  
treenode processFlow = ownerobject(activity);  
string ReferenciaCorrespondiente = "0";  
Object BloqueSeleccionado;
```

```
int Rotar = 0;

Table Bloques = Table.query("SELECT * FROM BloquesAPosicionar \
ORDER BY SizeBloqueY DESC, SizeBloqueX DESC, DuracionBloque ASC ");

for (int i = 1; i <= Bloques.numRows; i++)
{
    if (Bloques[i]["SizeBloqueY"] <= token.Area.SizeAreaY &&
Bloques[i]["SizeBloqueX"] <= token.Area.SizeAreaX)
    {
        ReferenciaCorrespondiente = Bloques[i]["ReferenciaBloque"];

        break;
    }
    else if (Bloques[i]["SizeBloqueX"] <= token.Area.SizeAreaY &&
Bloques[i]["SizeBloqueY"] <= token.Area.SizeAreaX)
    {
        ReferenciaCorrespondiente = Bloques[i]["ReferenciaBloque"];
        Rotar = 1;
        break;
    }
}
if (ReferenciaCorrespondiente == "0")
{
    token.labels.assert("Bloque").value = 0;
}
else
{
    string query = "WHERE ReferenciaBloque = " + "'" +
ReferenciaCorrespondiente + "'";
    BloqueSeleccionado = List("BloquesAPosicionar").pull(query, 1, 1);

    if (Rotar == 1)
    {

        setsize(BloqueSeleccionado, BloqueSeleccionado.SizeBloqueY, BloqueSeleccio
nado.SizeBloqueX, BloqueSeleccionado.size.z);

    }

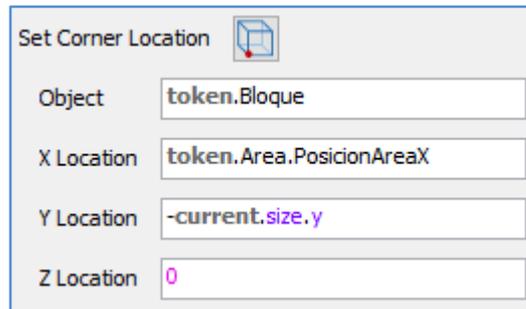
    BloqueSeleccionado.labels.assert("PosicionBloqueX").value =
token.Area.PosicionAreaX;
    BloqueSeleccionado.labels.assert("Arista").value = token.Area.Arista;
    BloqueSeleccionado.labels.assert("AreaAsignacion").value =
token.Area.IDArea;

    token.labels.assert("Bloque").value = BloqueSeleccionado;
}
}
```

Con esto se comprueba si hay algún bloque cuya geometría sea adecuada para el área seleccionada, ya sea en orientación vertical u horizontal, y si existe dicho bloque lo retira de la lista de bloques esperando a ser colocados y asigna la referencia a dicho bloque en el modelo en una etiqueta del *token* llamada "Bloque". Si no existiese un bloque que se adecue al área, esta sería desechada por el propio bucle mediante una actividad de decisión, cuya condición depende de la etiqueta "Bloque" del *token* (el código anterior asigna el valor 0 si no se produce la asignación de un bloque), ya que se volvería al paso en el que se selecciona un área de nuevo y el área sin asignación de bloque ya no estaría incluida en la lista de áreas disponibles.

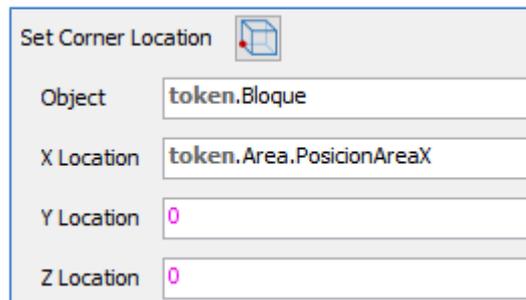
Tras producirse un emparejamiento a área y bloque válidos, se creará un *token* con las mismas etiquetas de referencia al bloque y al área seleccionados que el *token*

anterior, en la sección llamada "Posicionamiento Bloques". En esta sección, se realizará la colocación del bloque seleccionado en el parque en las coordenadas correspondientes al área selección. Para ello, primero se atraviesa una actividad de decisión en función de si el área es inferior o superior (comprobando la etiqueta "Arista" del *token* que representa al área). Tras esto, atraviesa una actividad que mueve el bloque al almacén y luego se posiciona con la siguiente actividad, llamada "Posicionar", cuyas coordenadas y punto de referencia dependerán de la posición del área seleccionada.



Field	Value
Object	token.Bloque
X Location	token.Area.PosicionAreaX
Y Location	-current.size.y
Z Location	0

Figura 40. Configuración de la actividad "Posicionar" para la arista inferior (Fuente: elaboración propia).



Field	Value
Object	token.Bloque
X Location	token.Area.PosicionAreaX
Y Location	0
Z Location	0

Figura 41. Configuración de la actividad "Posicionar" para la arista superior (Fuente: elaboración propia).

Completada la ubicación, el *token* atraviesa primero un *script* en el que se calcula la ocupación del parque, para posteriormente elaborar una gráfica que registre su evolución durante el proceso de ubicación de los bloques. En este, se actualiza el valor de una variable global "AreaOcupada" sumando el área del bloque que se acaba de ubicar. Con esto se calcula el valor de la ocupación y se asigna en una *tracked variable* llamada "OcupacionParque", que es la que nos permitirá graficar dicho valor. El código es el siguiente:

```
/**Actualizar ocupación*/  
Object current = param(1);  
treenode activity = param(2);  
Token token = param(3);  
treenode processFlow = ownerobject(activity);  
  
Object Bloque = token.Bloque;  
AreaOcupada = AreaOcupada + (Bloque.size.x * Bloque.size.y);  
double OcupacionParque = 100 * AreaOcupada / (current.size.x *  
current.size.y);
```

```
TrackedVariable("OcupacionParque").value = OcupacionParque;
```

En el final de esta sección se crea dos *tokens* con las mismas etiquetas de referencia al bloque y al área, uno en la sección “Actualización Áreas”, con la que se seguirá la explicación; y otro en la sección “Salida Bloques”, que se explicará al final.

En la sección “Actualización Áreas” se aplica la regla de actualización de áreas que hemos descrito. Para ello, el *token* atraviesa una actividad *script* con el siguiente código:

```
/**Regla de actualización de áreas*/
Object current = param(1);
treenode activity = param(2);
Token token = param(3);
treenode processFlow = ownerobject(activity);

Object Bloque = token.Bloque;
Table Areas = Table.query("SELECT * FROM AreasDisponibles");
var EntradasAreas = List("AreasDisponibles").entries();
Token AreaEvaluada;
Table Bloques = Table.query("SELECT * FROM BloquesAPosicionar");
int Eliminar = 0;
double InicioBloqueX = Bloque.location.x;
double FinalBloqueX = Bloque.location.x + Bloque.size.x;
double InicioAreaX;
double FinalAreaX;
double NuevoInicioAreaX;
double NuevoFinalAreaX;
double NuevoSizeAreaY;

// Recorremos todas las áreas disponibles
for (int i = 1; i <= Areas.numRows; i++)
{
    InicioAreaX = Areas[i]["PosicionAreaX"];
    FinalAreaX = Areas[i]["PosicionAreaX"] + Areas[i]["SizeAreaX"];
    NuevoInicioAreaX = InicioAreaX;
    NuevoFinalAreaX = FinalAreaX;
    NuevoSizeAreaY = Areas[i]["SizeAreaY"];
    Eliminar = 0;

    // Comprobamos si el bloque que acabamos de colocar está en la arista
    opuesta a la del área que vamos a comprobar
    if (Bloque.Arista != Areas[i]["Arista"])
    {
        // Si se cumple alguna de las siguientes condiciones no hay
        colisión
        if ((FinalAreaX <= InicioBloqueX) || (InicioAreaX >=
        FinalBloqueX))
        {
            /* No hay colisión */
        }
        // Comprobamos si hay colisión con entre el bloque y un área en
        la arista opuesta (Ancho Parque - Ancho Bloque < Ancho Area)
        else if (Bloque.size.y > current.size.y - Areas[i]["SizeAreaY"])
        {
            // Calculamos el nuevo tamaño de ese área para que no se
            produzca una colisión
            NuevoSizeAreaY = current.size.y - Bloque.size.y;
        }
    }
    // Si el bloque y el área están en la misma arista
    else
    {
        // Si se cumple alguna de las siguientes condiciones no hay
        colisión
    }
}
```

```
        if ((FinalAreaX <= InicioBloqueX) || (InicioAreaX >=
FinalBloqueX))
        {
            /* No hay colisión */
        }
        // Si el bloque ocupa todo el área se elimina
        else if ((InicioAreaX >= InicioBloqueX) && (FinalAreaX <=
FinalBloqueX))
        {
            Eliminar = 1;
        }
        // Se reduce el margen derecho del área que estamos comprobando
        // si habiendo colisión el punto de localización del bloque está más a la derecha
        // que el del área
        else if ((InicioAreaX < InicioBloqueX) && (FinalAreaX >
InicioBloqueX))
        {
            NuevoFinalAreaX = InicioBloqueX;
        }
        // Se reduce el margen izquierdo del Área que estamos comprobando
        // si habiendo colisión el punto de localización del bloque está más a la
        // izquierdo que el del área
        else if ((FinalAreaX > FinalBloqueX) && (InicioAreaX <
FinalBloqueX))
        {
            NuevoInicioAreaX = FinalBloqueX;
        }
    }
    // Comprobamos si algún bloque esperando a colocación cabe en este área
    // si no hay motivo para eliminarla
    if (Eliminar == 0)
    {
        for (int j = 1; j <= Bloques.numRows; j++)
        {
            Eliminar = 1;
            // Si alguno de los bloques cabe en este margen no
            // eliminamos el área
            if ((Bloques[j]["SizeBloqueX"] ||
Bloques[j]["SizeBloqueY"]) < (NuevoFinalAreaX - NuevoInicioAreaX))
            {
                Eliminar = 0;
            }
        }
    }

    // Se ejecutan los cambios pertinentes en el área que estamos
    // comprobando
    AreaEvaluada = EntradasAreas[i].value;
    AreaEvaluada.PosicionAreaX = NuevoInicioAreaX;
    AreaEvaluada.SizeAreaX = NuevoFinalAreaX - NuevoInicioAreaX;
    AreaEvaluada.SizeAreaY = NuevoSizeAreaY;

    if (Eliminar == 1)
    {
        AreaEvaluada = EntradasAreas[i].value;
        AreaEvaluada.SizeAreaX = 0;
        AreaEvaluada.SizeAreaY = 0;
    }
}
```

En este, se comprueba que el bloque que se acaba de colocar no colisione con las áreas contenidas en la lista de áreas disponibles. Si el bloque entrase en conflicto con un área, se comprueba cómo es la colisión, y en función de ello se procede a reducir el

tamaño del área en la dimensión que corresponda o a la eliminación del área si el bloque la invalida totalmente.

Como ya se ha presentado al explicar el algoritmo, es necesario también crear nuevas áreas, por ello, al final de esta sección anterior el *token* se dirige a la actividad “Derivación de áreas” dentro de la sección “Generación de áreas”. En esta continuación del diagrama de flujo se cuenta con dos actividades que crearán los *tokens* que representan las nuevas áreas según la regla de derivación de áreas. En estos, se cuenta con las mismas etiquetas que para el área inicial, pero con los valores correspondientes a las nuevas áreas creadas.

Estos *tokens* que representan a las áreas derivadas se crean en la actividad “Regla de actualización de áreas” de la misma sección. Como esta vez, al aplicar la regla de actualización de áreas hay que recorrer todos los bloques posicionados, se emplea un nuevo *script* con el siguiente código:

```
/**Regla de actualización de áreas*/
Object current = param(1);
treenode activity = param(2);
Token token = param(3);
treenode processFlow = ownerobject(activity);

Table Bloques = Table.query("SELECT * FROM BloquesAPosicionar");
Object Bloque;
int Eliminar = 0;
double InicioBloqueX;
double FinalBloqueX;
double InicioAreaX = token.PosicionAreaX;
double FinalAreaX = token.PosicionAreaX + token.SizeAreaX;
double NuevoSizeAreaY = token.SizeAreaY;
double NuevoInicioAreaX = InicioAreaX;
double NuevoFinalAreaX = FinalAreaX;

// Recorremos todos los bloques ya colocados para comprobar si colisionan con
// el área generada
for (int i = 1; i <= current.subnodes.length; i++)
{
    // Asignamos el bloque a una variable tipo Object para poder leer su
    // posición y tamaño
    Bloque = current.subnodes[i];

    InicioBloqueX = Bloque.location.x;
    FinalBloqueX = Bloque.location.x + Bloque.size.x;

    // Comprobamos si el área que acabamos de crear está en la arista
    // opuesta a la del bloque con el que comprobamos la colisión
    if (token.Arista != Bloque.Arista)
    {
        // Si se cumple alguna de las siguientes condiciones no hay
        // colisión
        if ((FinalBloqueX <= InicioAreaX) || (InicioBloqueX >=
        FinalAreaX))
        {
            /* No hay colisión */
        }
        // Comprobamos si hay colisión con entre área creada y un bloque
        // en la arista opuesta (Ancho Parque - Ancho Bloque < Ancho Area)
        else if (Bloque.size.y > current.size.y - token.SizeAreaY)
        {
            NuevoSizeAreaY = min(NuevoSizeAreaY, current.size.y -
            Bloque.size.y);
        }
    }
    // Si el bloque y el área están en la misma arista
```

```
        else
        {
            // Si se cumple alguna de las siguientes condiciones no hay
colisión
            if ((FinalBloqueX <= InicioAreaX) || (InicioBloqueX >=
FinalAreaX))
            {
                /* No hay colisión */
            }
            // Si el bloque ocupa todo el área se elimina
            else if ((InicioBloqueX <= InicioAreaX) && (FinalBloqueX >=
FinalAreaX))
            {
                //Eliminar área
                Eliminar = 1;
            }
            // Se reduce el margen derecho del área que estamos comprobando
si habiendo colisión el punto de localización del bloque está más a la derecha
que el del área
            else if ((InicioBloqueX < FinalAreaX) && (InicioBloqueX >
InicioAreaX))
            {
                //Se reduce el margen derecho del área
                NuevoFinalAreaX = min(NuevoFinalAreaX, InicioBloqueX);
            }
            // Se reduce el margen izquierdo del área que estamos comprobando
si habiendo colisión el punto de localización del bloque está más a la
izquierda que el del área
            else if ((FinalBloqueX > InicioAreaX) && (FinalBloqueX <
FinalAreaX))
            {
                //Se reduce el margen izquierdo del área
                NuevoInicioAreaX = max(NuevoInicioAreaX, FinalBloqueX);
            }
        }
    }
    // Comprobamos si algún bloque cabe en este área si no hay motivos para
eliminarla
    if (Eliminar == 0)
    {
        for (int j = 1; j <= Bloques.numRows; j++)
        {
            Eliminar = 1;
            // Si alguno de los bloques cabe en este margen no eliminamos el
área
            if ((Bloques[j]["SizeBloqueX"] || Bloques[j]["SizeBloqueY"]) <
(NuevoFinalAreaX - NuevoInicioAreaX))
            {
                Eliminar = 0;
            }
        }
    }

    // Se ejecutan los cambios pertinentes en el área que estamos
comprobando
    token.PosicionAreaX = NuevoInicioAreaX;
    token.SizeAreaX = NuevoFinalAreaX - NuevoInicioAreaX;
    token.SizeAreaY = NuevoSizeAreaY;
    token.labels.assert("Eliminar").value = Eliminar;
```

En este *script* se sigue una metodología similar a la que ya se ha explicado en el de actualización de áreas anterior, solo que ahora el bucle recorre los bloques ubicados en el parque. Para eliminar áreas no válidas, en este caso se inserta una etiqueta en el *token* llamada "Eliminar", de forma que posteriormente el *token* sea descartado en un actividad de decisión si el valor de esta etiqueta es 1. En el caso

anterior, las áreas no válidas se eliminaban reduciendo su tamaño a 0, ya que eliminarlas en ese instante de la lista impediría que el código recorra correctamente la lista de áreas disponibles. Tras esto, si el área creada no debe ser eliminada, el *token* que la representa accede a la actividad en la que se añade dicho *token* a la lista de "AreasDisponibles", en la que ya se había incluido el primer área en el inicio, cerrando así el ciclo del *process flow*.

Por último, queda reproducir la salida de los bloques del parque. Esta se efectuará cuando se completa el tiempo de procesamiento del bloque en el parque. Para ello, en el instante que ocurre la colocación en la sección "Posicionamiento Bloques" se había creado un *token* en la sección "Salida Bloques". Este *token* entra en dicho instante en una actividad en la que permanecerá el mismo tiempo que dura el trabajo del bloque en el parque, en función de la etiqueta "DuracionBloque" de dicho bloque. Pasado el tiempo correspondiente, el *token* accederá a una actividad que retira el bloque del parque llevándolo al sumidero (*sink*) del que se dispone en el modelo. Tras esto, se actualiza la variable global "AreaOcupada" y la *tracked variable* "OcupacionParque" para la posterior representación de esta en una gráfica. Para esto, se emplea un código similar al de la actividad "Actualizar ocupación" anterior, pero en este caso al actualizar "AreaOcupada" se resta el área del bloque que sale del parque, en lugar de efectuar una suma.

Como se necesita crear en área que se adecúe al hueco que deja el bloque saliente, se crea un *token* para esta a continuación en la actividad "Nueva área hueco". A este token se le asignan las mismas etiquetas que para las demás áreas, pero con los valores adecuados al espacio que deja un bloque al salir. Para ajustarse correctamente a las dimensiones de dicho espacio, para el valor de las etiquetas "PosicionAreaX" y "SizeAreaX" se han empleado los siguientes dos *scripts*:

- Código "PosicionAreaX":

```
/**Comprobar inicio hueco*/
Object current = param(1);
treenode activity = param(2);
Token token = param(3);
Token createdToken = param(4);
Variant assignTo = createdToken;
int creationRank = param(5);
string labelName = param(6);
treenode processFlow = ownerobject(activity);

Object BloqueSaliente = token.Bloque;
Object Bloque;

double InicioBloqueSaliente = BloqueSaliente.location.x;
double InicioHueco = 0;

for (int i = 1; i <= current.subnodes.length; i++)
{
    // Asignamos el bloque a una variable tipo Object para poder trabajar
    con él
    Bloque = current.subnodes[i];

    if ((Bloque.Arista == BloqueSaliente.Arista) && (Bloque.ReferenciaBloque
    != BloqueSaliente.ReferenciaBloque))
    {
        if ((Bloque.location.x + Bloque.size.x) <= InicioBloqueSaliente)
        {
            InicioHueco = max(Bloque.location.x +
            Bloque.size.x, InicioHueco);
        }
    }
}
```

```
    }  
}  
  
return InicioHueco;
```

- Código “SizeAreaX”:

```
/**Comprobar tamaño hueco*/  
Object current = param(1);  
treenode activity = param(2);  
Token token = param(3);  
Token createdToken = param(4);  
Variant assignTo = createdToken;  
int creationRank = param(5);  
string labelName = param(6);  
treenode processFlow = ownerobject(activity);  
  
Object BloqueSaliente = token.Bloque;  
Object Bloque;  
  
double InicioBloqueSaliente = BloqueSaliente.location.x;  
double FinalBloqueSaliente = BloqueSaliente.location.x +  
BloqueSaliente.size.x;  
double InicioHueco = 0;  
double FinalHueco = current.size.x;  
  
for (int i = 1; i <= current.subnodes.length; i++)  
{  
    // Asignamos el bloque a una variable tipo Object para poder trabajar  
    con él  
    Bloque = current.subnodes[i];  
  
    if ((Bloque.Arista == BloqueSaliente.Arista) && (Bloque.ReferenciaBloque  
!= BloqueSaliente.ReferenciaBloque))  
    {  
        if ((Bloque.location.x + Bloque.size.x) <= InicioBloqueSaliente)  
        {  
            InicioHueco = max(Bloque.location.x +  
Bloque.size.x, InicioHueco);  
        }  
        if (Bloque.location.x >= FinalBloqueSaliente)  
        {  
            FinalHueco = min(Bloque.location.x, FinalHueco);  
        }  
    }  
}  
  
return (FinalHueco - InicioHueco)
```

Con esto, se comprueba las coordenadas en el eje X de inicio y final del hueco. Para ello se recorren todos los bloques que se sitúan en el mismo lado del parque que el bloque que sale, siguiendo la regla correspondiente que se ha presentado en la explicación del algoritmo.

Posteriormente, el *token* que representa esta área atraviesa la actividad que contiene la regla de actualización de áreas en la sección “Generación Áreas”, igual que en la creación de áreas tras la colocación de un bloque y tras esto se incluye en la lista de áreas disponibles de la misma forma.

5.2.3 Ejecución del modelo

Completado el desarrollo del modelo, se ejecuta la simulación de este para comprobar su correcto funcionamiento. En la ejecución se visualiza el proceso de ubicación de bloques, el cual se completa adecuadamente. A continuación se muestra el estado del parque en diferentes instantes de tiempo:

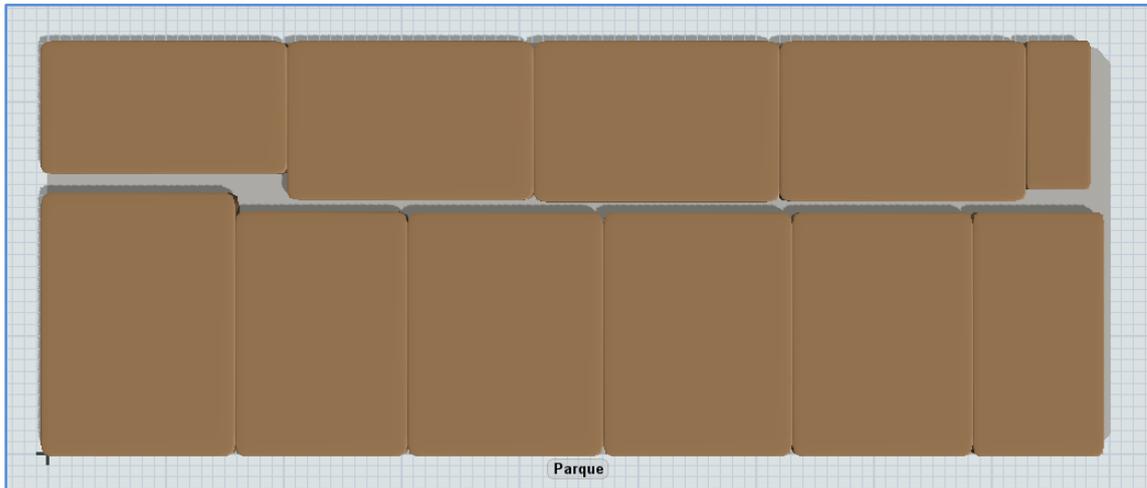


Figura 42. Estado del parque en $t = 200$ h (Fuente: elaboración propia).

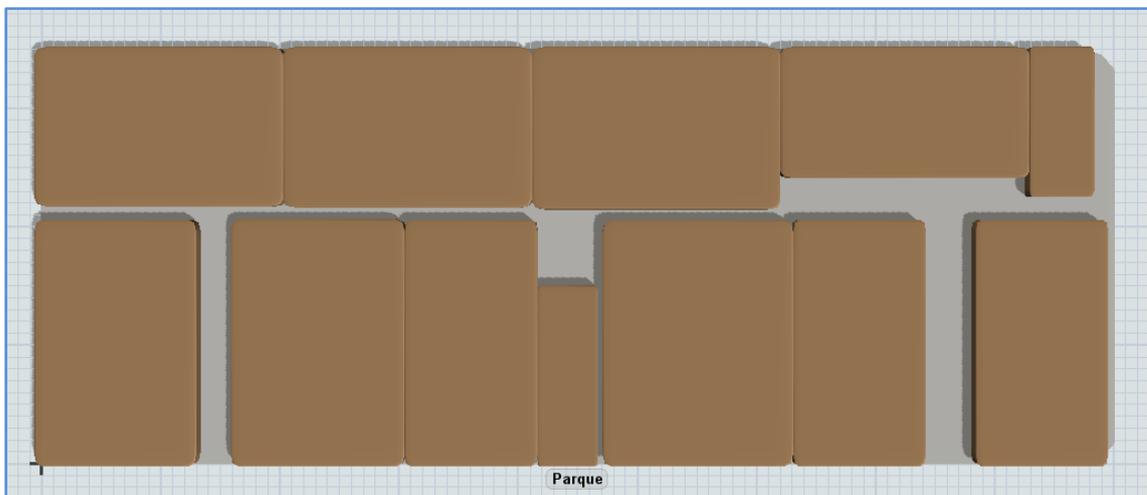


Figura 43. Estado del parque en $t = 1100$ h (Fuente: elaboración propia).

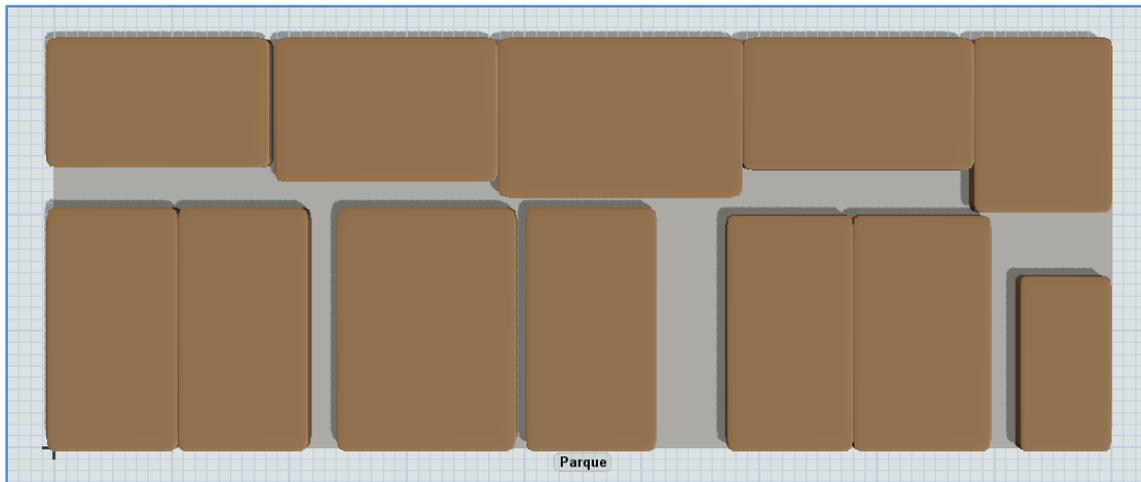


Figura 44. Estado del parque en $t = 7000$ h (Fuente: elaboración propia).

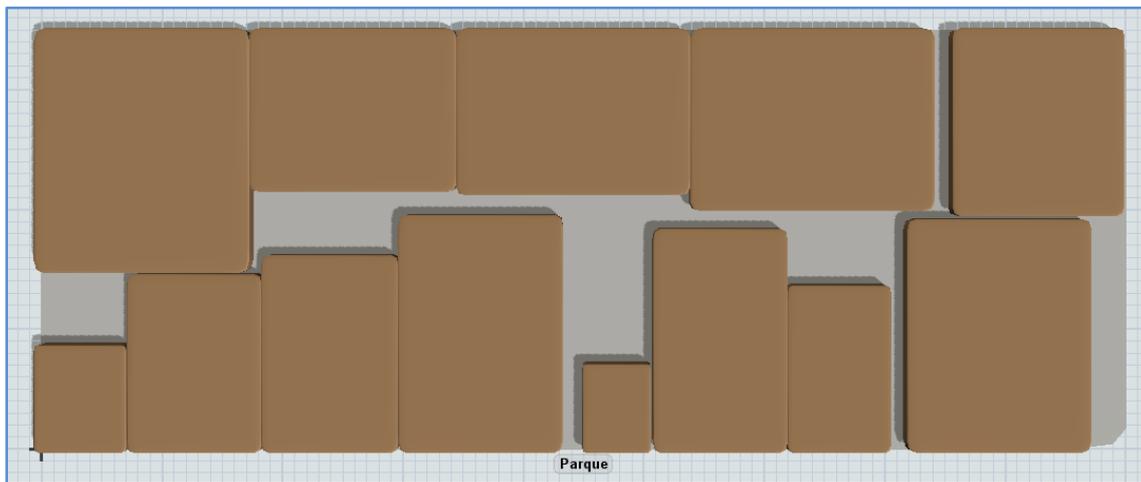


Figura 45. Estado del parque en $t = 15000$ h (Fuente: elaboración propia).

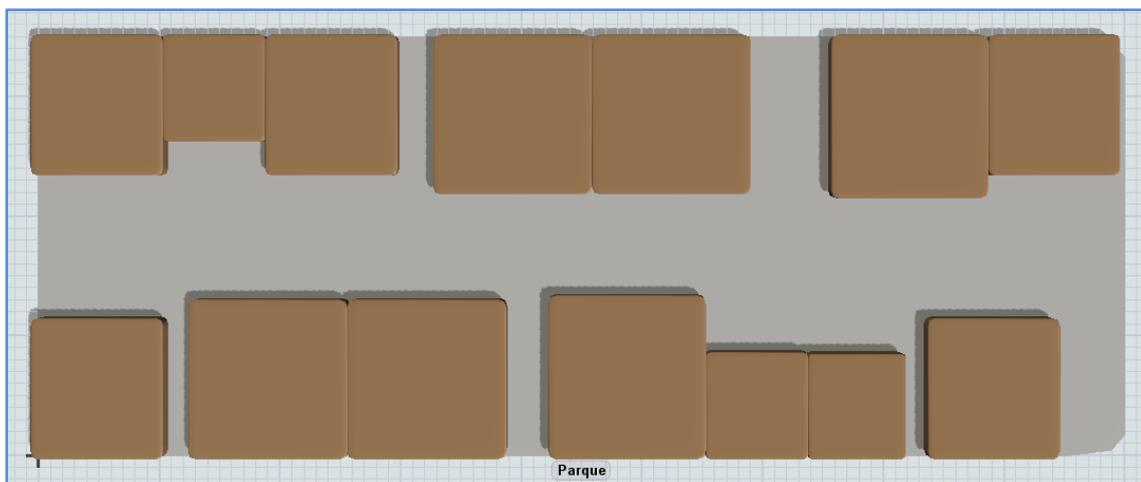


Figura 46. Estado del parque en $t = 24800$ h (Fuente: elaboración propia).

6 EXPERIMENTACIÓN

Tras haber creado el modelo y comprobado su correcto funcionamiento, se diseñan una serie de escenarios que nos permitirán obtener el caso que mejor resultados nos ofrece en cuanto a la ocupación del parque de bloques y al *makespan*.

Se realizan dos experimentos, uno para realizar una optimización del algoritmo y otro para observar la sensibilidad del *makespan* respecto a las dimensiones del parque. Para cada uno se crea un conjunto de escenarios, de manera que para el conjunto de escenarios del primer experimento se realizan variaciones de la lógica del algoritmo y para el conjunto del segundo experimento se realizan variaciones de las dimensiones del parque.

6.1 Optimización de la heurística

Los escenarios creados para este experimento son los siguientes:

- **Escenario 1.** Se siguen las reglas del algoritmo que se ha descrito previamente.
- **Escenario 2.** Se modifica la regla de localización, de manera que ahora se prioriza la arista en la que se coloca el bloque frente a la cercanía al origen.
- **Escenario 3.** Se modifica la regla de localización, de manera que ahora se prefieren áreas alejadas al origen.
- **Escenario 4.** Se modifica la regla de selección de bloques, de manera que ahora se priorizan los bloques de menor tamaño.
- **Escenario 5.** Se modifica la regla de selección de bloques, de manera que ahora, ante una igualdad en tamaño, se selecciona el bloque de mayor duración.
- **Escenario 6.** Se modifica la regla de selección de bloques, de manera que ahora se prioriza la menor duración de los bloques frente a su tamaño.
- **Escenario 7.** Se modifica la regla de selección de bloques, de manera que ahora se prioriza la mayor duración de los bloques frente a su tamaño.

Tabla 7. Escenarios diseñados para optimizar la heurística (Fuente: elaboración propia).

Escenario	Dimensión Parque (m)	Priorización Áreas	Priorización Bloques
1	90 x 35	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC
2	90 x 35	ORDER BY PosicionAreaY ASC, PosicionAreaX ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC
3	90 x 35	ORDER BY PosicionAreaX DESC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC
4	90 x 35	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y ASC, SizeBloqueX ASC, DuracionBloque ASC
5	90 x 35	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque DESC
6	90 x 35	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY DuracionBloque ASC, SizeBloque Y DESC, SizeBloqueX DESC
7	90 x 35	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY DuracionBloque DESC, SizeBloque Y DESC, SizeBloqueX DESC

6.1.1 Resultados y análisis del primer experimento

Con estos siete escenarios se pretende observar la influencia de los cambios en las reglas del algoritmo sobre el *makespan* y la ocupación máxima y media de la superficie del parque. Como se ha visto, se han realizado variaciones en las reglas de prioridad que se habían definido, obteniendo los resultados que se recogen en la siguiente tabla.

Tabla 8. Resultados obtenidos en el primer experimento (Fuente: elaboración propia).

Escenario	Makespan (h)	Jornadas (16h)	Ocupación Máxima (%)	Ocupación media (%)
1	25690,11	1606	94,97	83,49
2	26167,26	1636	95,36	81,97
3	26889,15	1681	92,84	79,77
4	31533,73	1971	89,11	68,02
5	26729,77	1671	94,97	80,25
6	27771,03	1736	94,18	77,24
7	26492,34	1656	90,78	80,97
8	24248,52	1516	96,83	83,80
9	24008,09	1501	94,09	78,18

10	28331,59	1771	95,12	75,71
11	27050,10	1691	94,40	79,30

A continuación, se muestran las gráficas de la evolución de la ocupación del parque a lo largo de todo el proceso para cada uno de los escenarios.

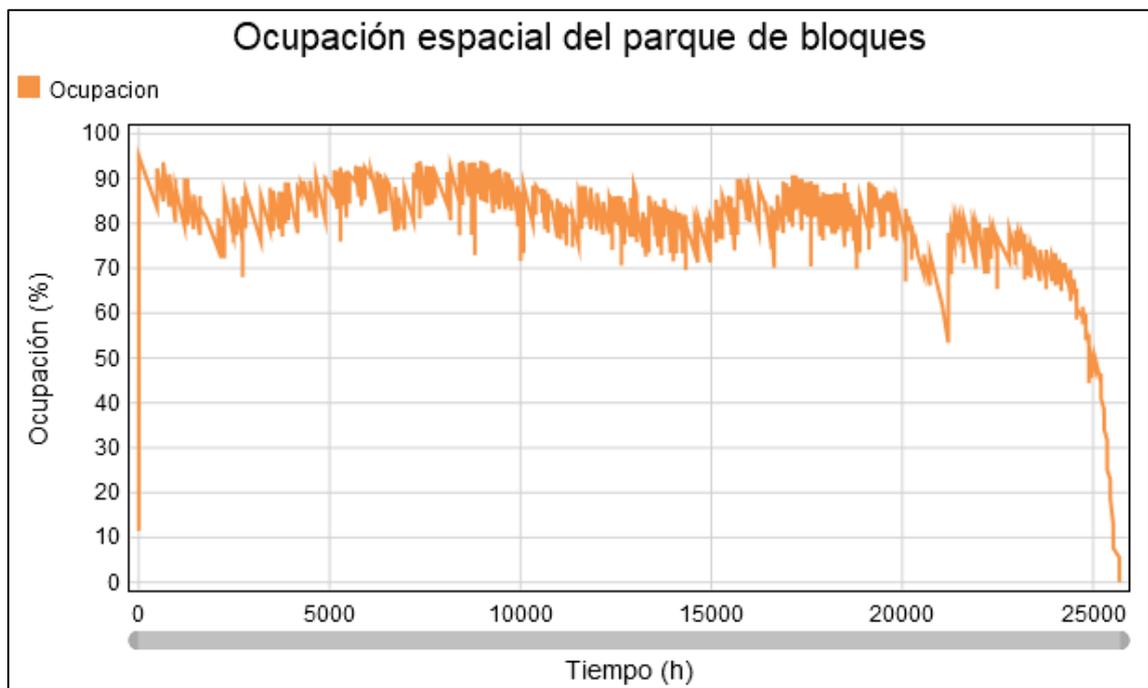


Figura 47. Evolución de la ocupación del parque del escenario 1 (Fuente: elaboración propia).

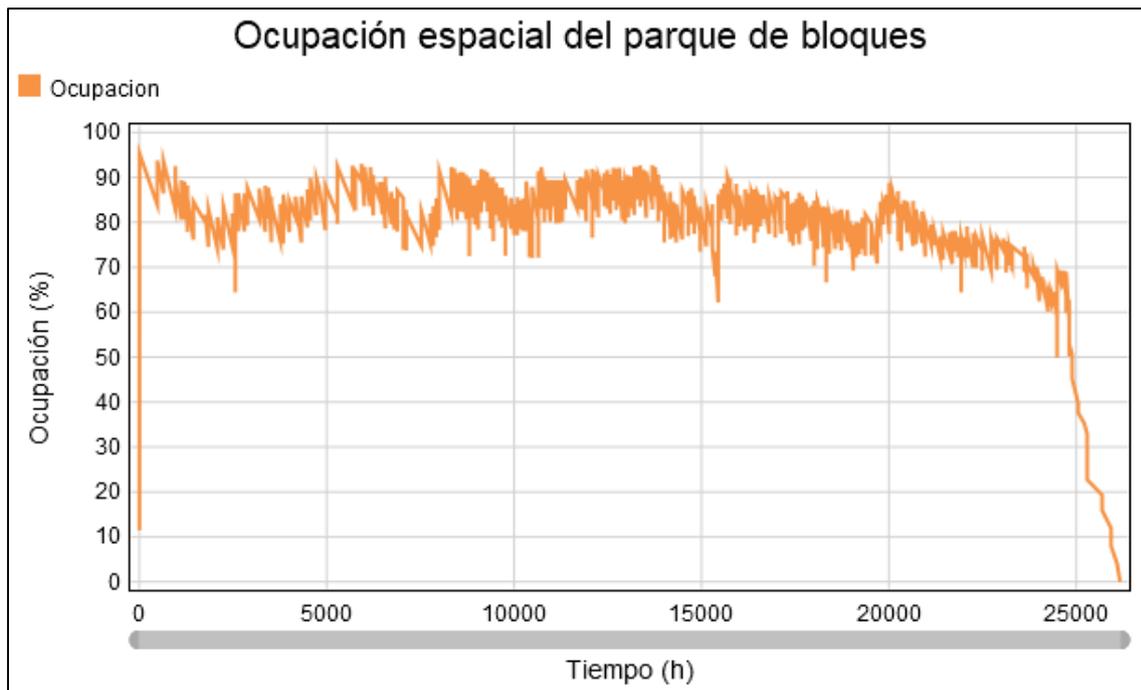


Figura 48. Evolución de la ocupación del parque del escenario 2 (Fuente: elaboración propia).

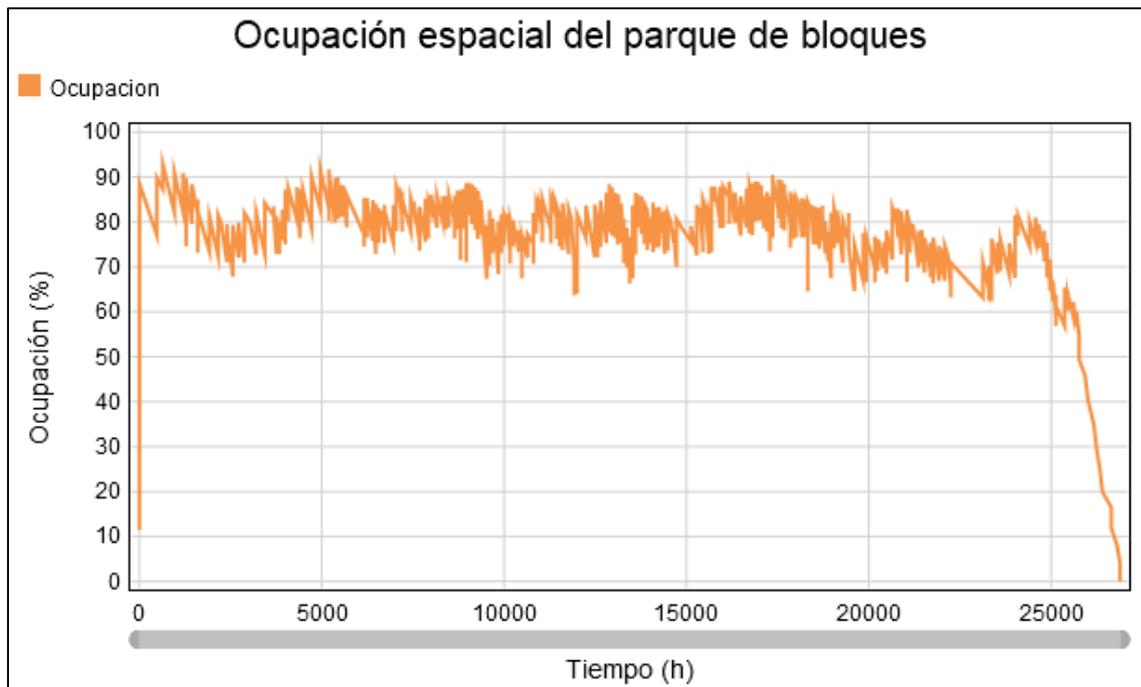


Figura 49. Evolución de la ocupación del parque del escenario 3 (Fuente: elaboración propia).

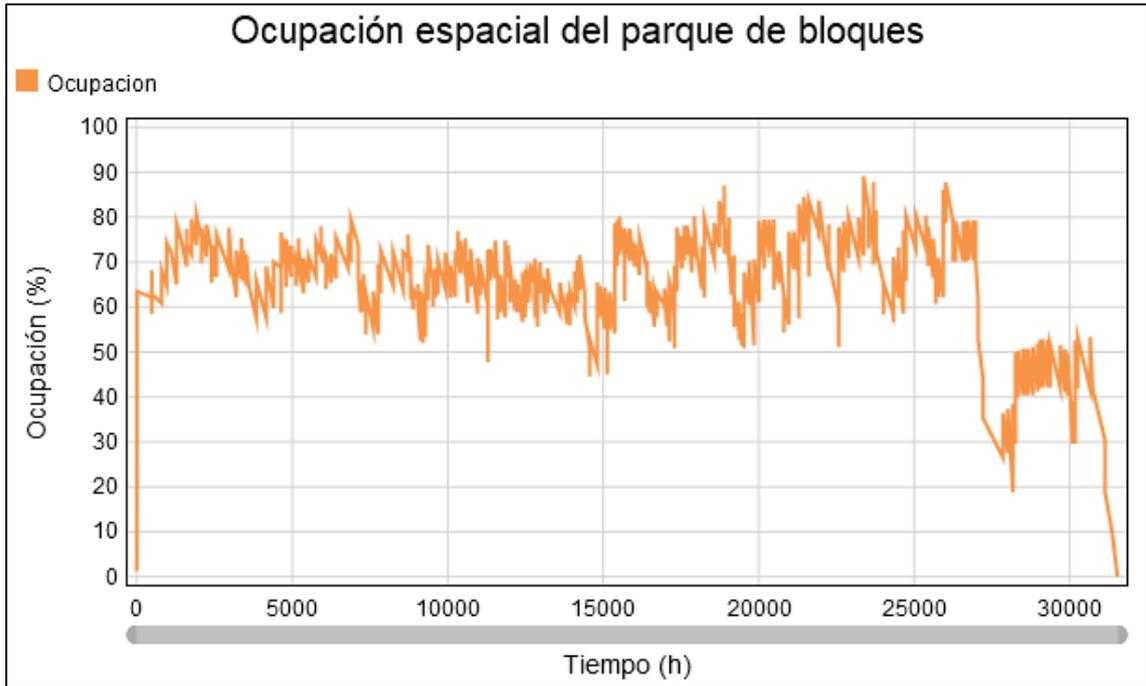


Figura 50. Evolución de la ocupación del parque del escenario 4 (Fuente: elaboración propia).

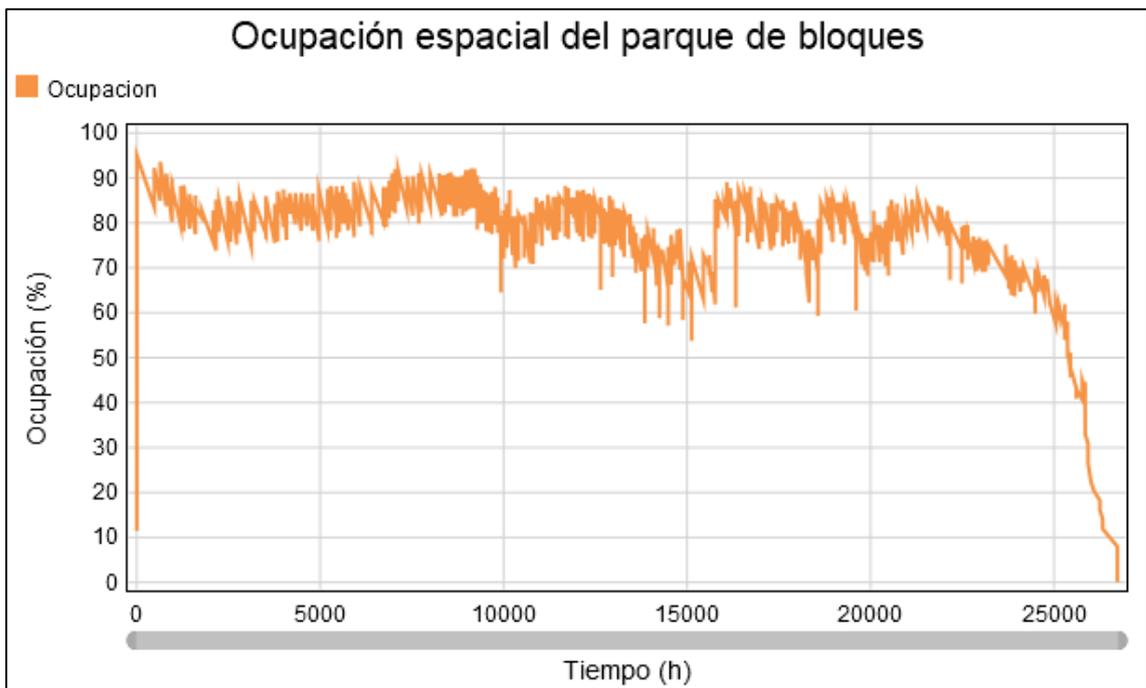


Figura 51. Evolución de la ocupación del parque del escenario 5 (Fuente: elaboración propia).

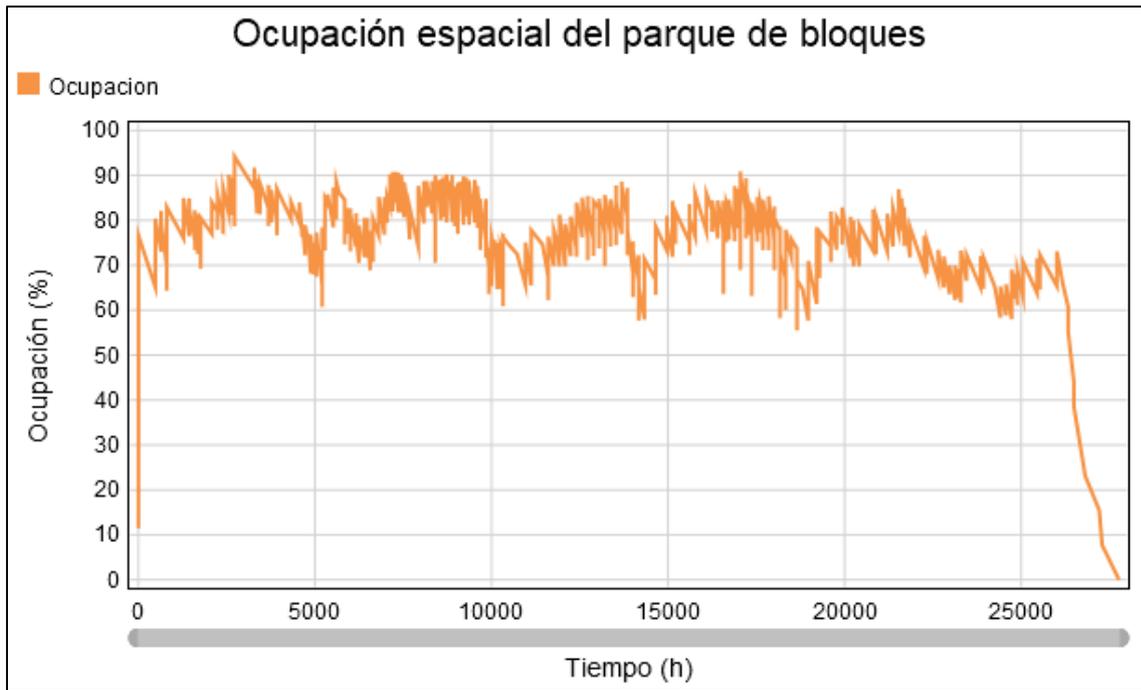


Figura 52. Evolución de la ocupación del parque del escenario 6 (Fuente: elaboración propia).

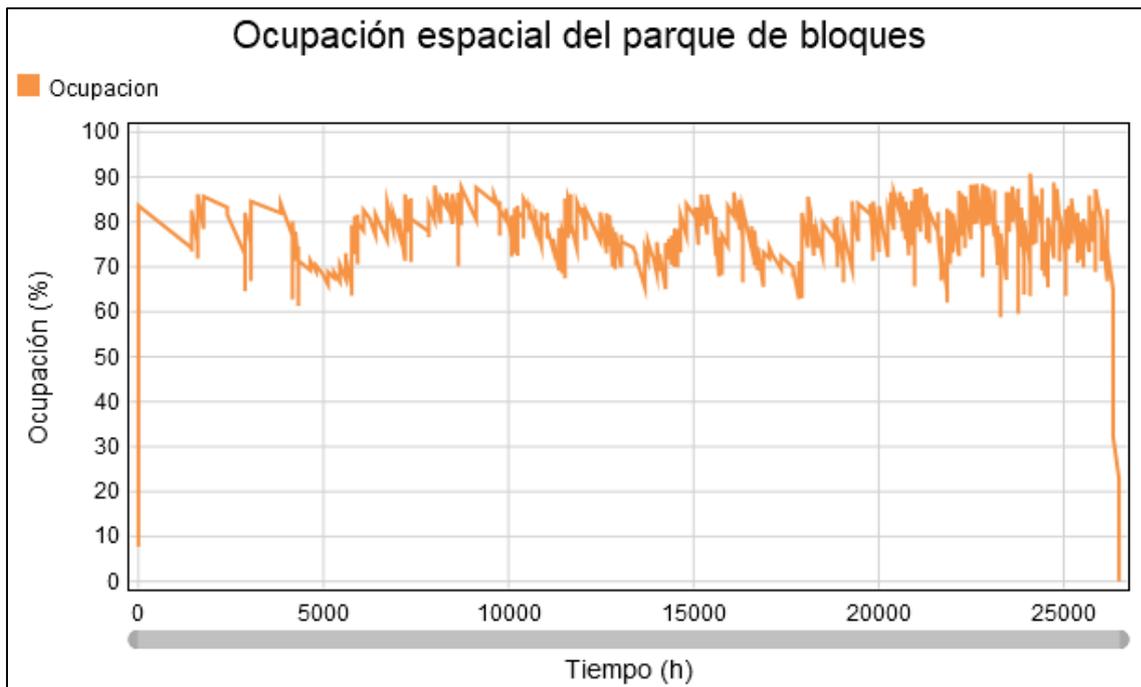


Figura 53. Evolución de la ocupación del parque del escenario 7 (Fuente: elaboración propia).

Respecto al *makespan*, de entre todos los escenarios creados variando el algoritmo, el que mejor resultado ofrece es el inicial. Esto se debe principalmente a que ofrece una mayor compacidad que los demás como se puede observar en la mayor ocupación media que proporciona, por los motivos que ya se han descrito al explicar las reglas del mismo. El segundo escenario, en el cual se favorece que primero se ubiquen los bloques en una única arista hasta que esta esté completa frente a alternar como en el primero, se observa que a pesar de tener una ocupación máxima mayor que el anterior, tanto el *makespan* como la ocupación media son inferiores, por lo tanto siendo un peor resultado.

El dato de la ocupación máxima no es representativo de que tan bueno es un escenario, ya que simplemente es un único *layout* de los bloques de entre todos los que se producen en cada uno de los escenarios, coincidiendo que este instante ocurre al llenar el parque con los primeros bloques que entran para la mayoría de escenarios, salvo en los que no se prioriza el mayor tamaño de los bloques. Esto se debe al ubicar los bloques desde un estado completamente vacío, en las áreas de ubicación que se van creando se puede ubicar siempre el bloque óptimo, ya que estas áreas iniciales son las de mayor tamaño. Posteriormente, como los bloques esperarán a que vayan quedando huecos, es más difícil que ocurra un estado de mayor ocupación.

Priorizar bloques de menor tamaño conduce a unos malos resultados, como es lógico, ya que una vez llenado el parque con los menores bloques, las áreas que van surgiendo con la salida de los mismos también son pequeñas, atrasando mucho la ubicación de los bloques de mayores dimensiones. En el escenario 7, al priorizar la duración frente al tamaño, también se empeoran los resultados, aunque no tanto como si damos preferencia a los menores bloques.

6.2 Sensibilidad respecto a las dimensiones del parque

Los escenarios creados para este experimento son los siguientes:

- **Escenario 8.** Se aumentan la longitud del área para observar que beneficio se obtendría al contar con un parque de mayor longitud.
- **Escenario 9.** Se aumentan el ancho del área para observar que beneficio se obtendría al contar con un parque de mayor anchura.
- **Escenarios 10 a 15.** Manteniendo la superficie del parque, se realiza una variación del ancho entre 30 y 40 m ajustando la longitud.

Tabla 9. Escenarios diseñados para analizar la sensibilidad respecto a las dimensiones del parque (Fuente: elaboración propia).

Escenario	Dimensión Parque (m)	Priorización Áreas	Priorización Bloques
8	95 x 35	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC
9	90 x 40	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC

10	105 x 30	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC
11	98,4 x 32	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC
12	92,6 x 34	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC
13	87,5 x 36	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC
14	82,9 x 38	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC
15	78,7 x 40	ORDER BY PosicionAreaX ASC, PosicionAreaY ASC LIMIT 1	ORDER BY SizeBloque Y DESC, SizeBloqueX DESC, DuracionBloque ASC

6.2.1 Resultados y análisis del segundo experimento

En este conjunto de escenarios, hemos realizado primero un aumento de cinco metros en cuanto a la longitud y a la anchura del parque, independientemente, que es lo que se recoge en los dos primeros escenarios. Para los siguientes, se ha realizado una variación del ancho del parque entre 30 y 40 metros, ajustando también su longitud para que la superficie del mismo se mantenga constante respecto a la original. Con esto, se pretende observar la influencia de la geometría y dimensiones del parque sobre los resultados, los cuales se recogen en la siguiente tabla.

Tabla 10. Resultados obtenidos en el primer experimento (Fuente: elaboración propia).

Escenario	Makespan (h)	Jornadas (16h)	Ocupación Máxima (%)	Ocupación media (%)
8	24248,52	1516	96,83	83,80
9	24008,09	1501	94,09	78,18
10	33293,15	2081	92,81	64,43
11	27370,20	1711	96,00	78,40
12	25610,98	1601	97,24	83,80
13	25689,10	1606	93,03	79,10
14	27050,72	1691	94,36	79,29
15	28010,07	1751	91,56	76,63

A continuación, se muestran las gráficas de la evolución de la ocupación del parque a lo largo de todo el proceso para cada uno de los escenarios.

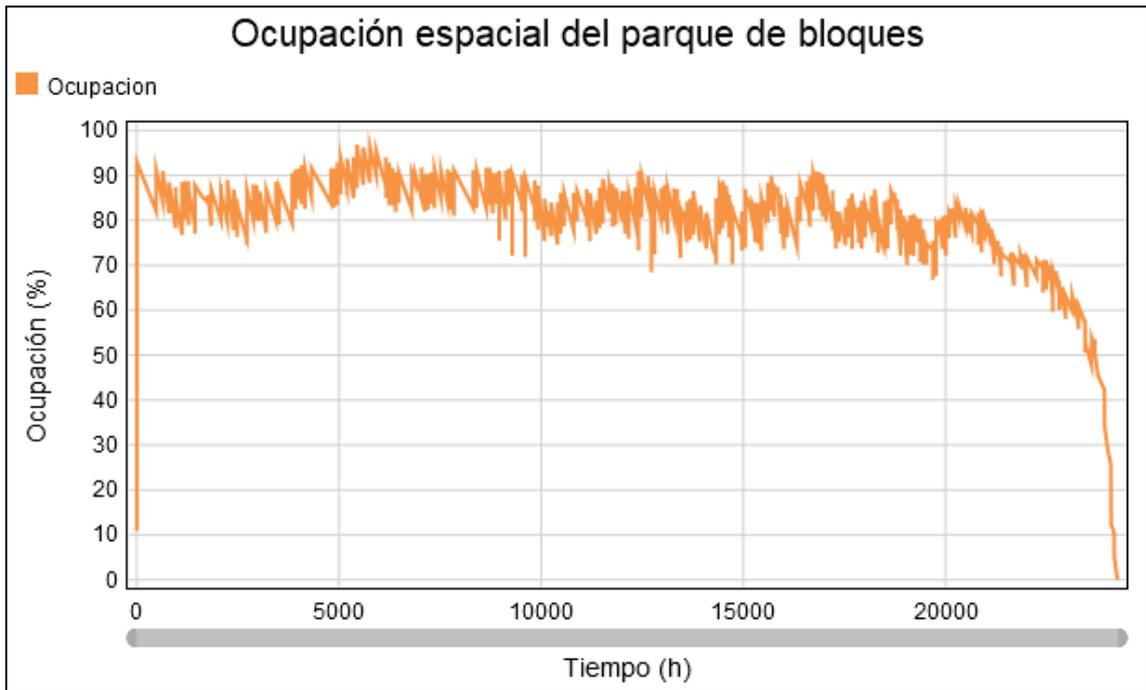


Figura 54. Evolución de la ocupación del parque del escenario 8 (Fuente: elaboración propia).

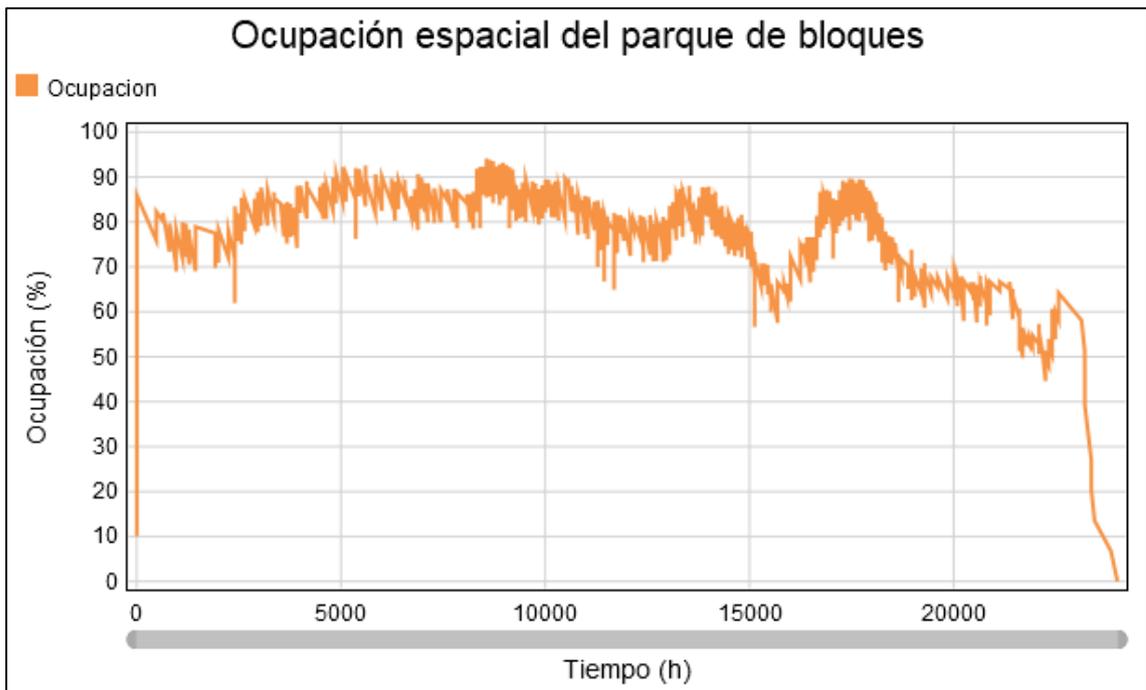


Figura 55. Evolución de la ocupación del parque del escenario 9 (Fuente: elaboración propia).

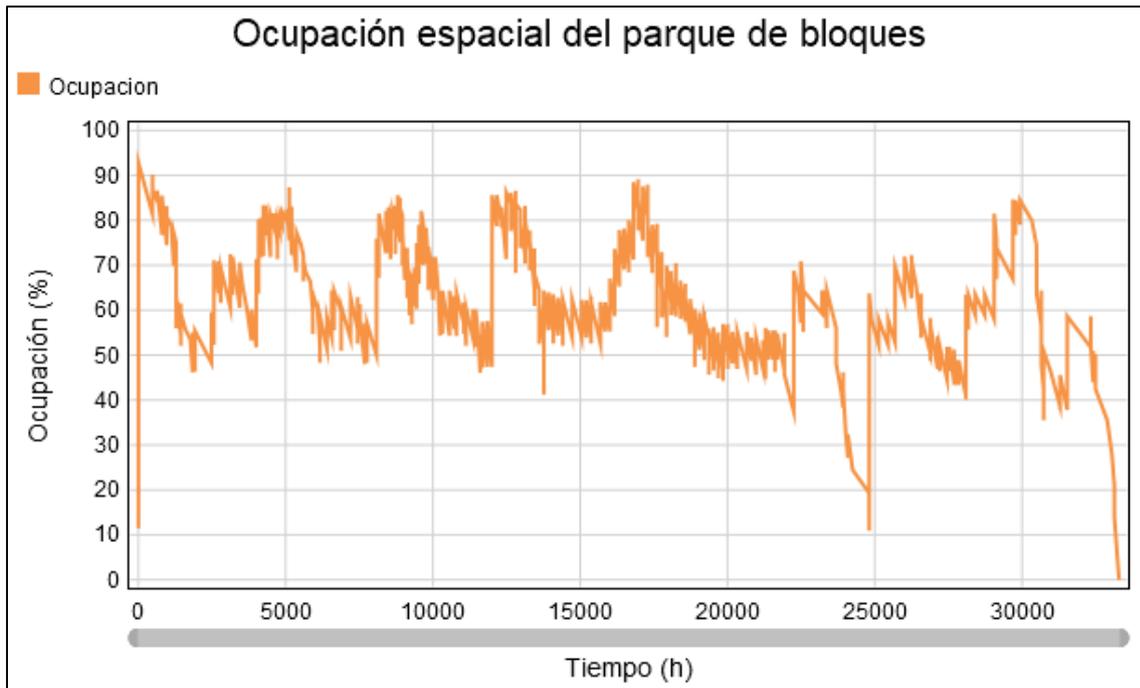


Figura 56. Evolución de la ocupación del parque del escenario 10 (Fuente: elaboración propia).

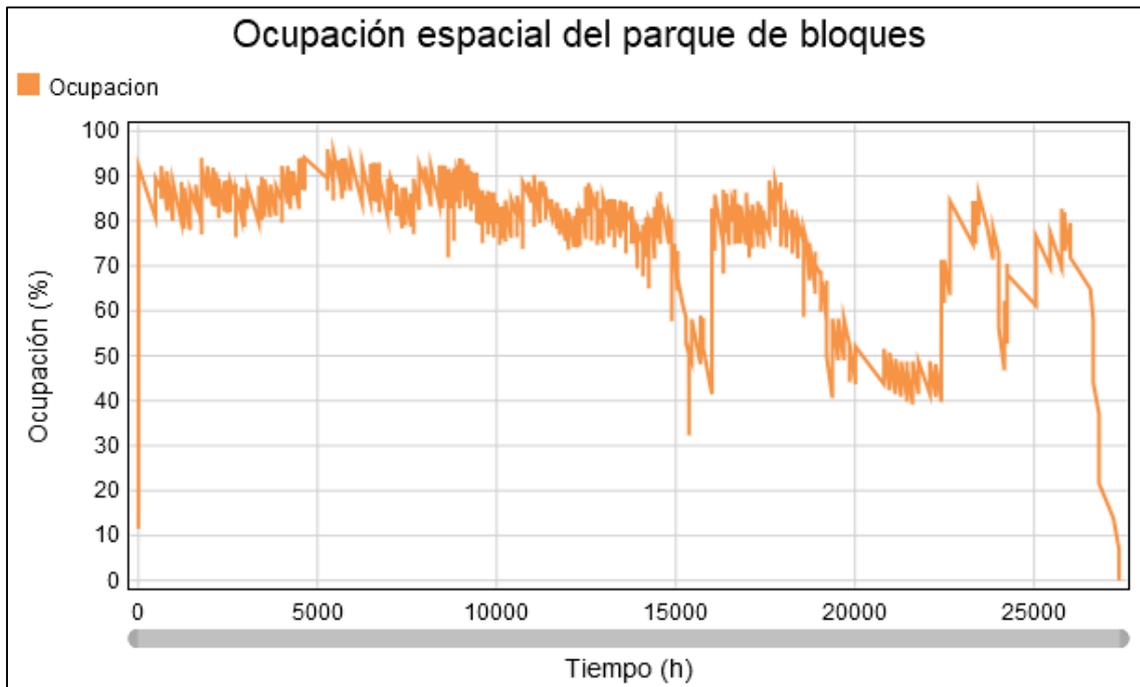


Figura 57. Evolución de la ocupación del parque del escenario 11 (Fuente: elaboración propia).

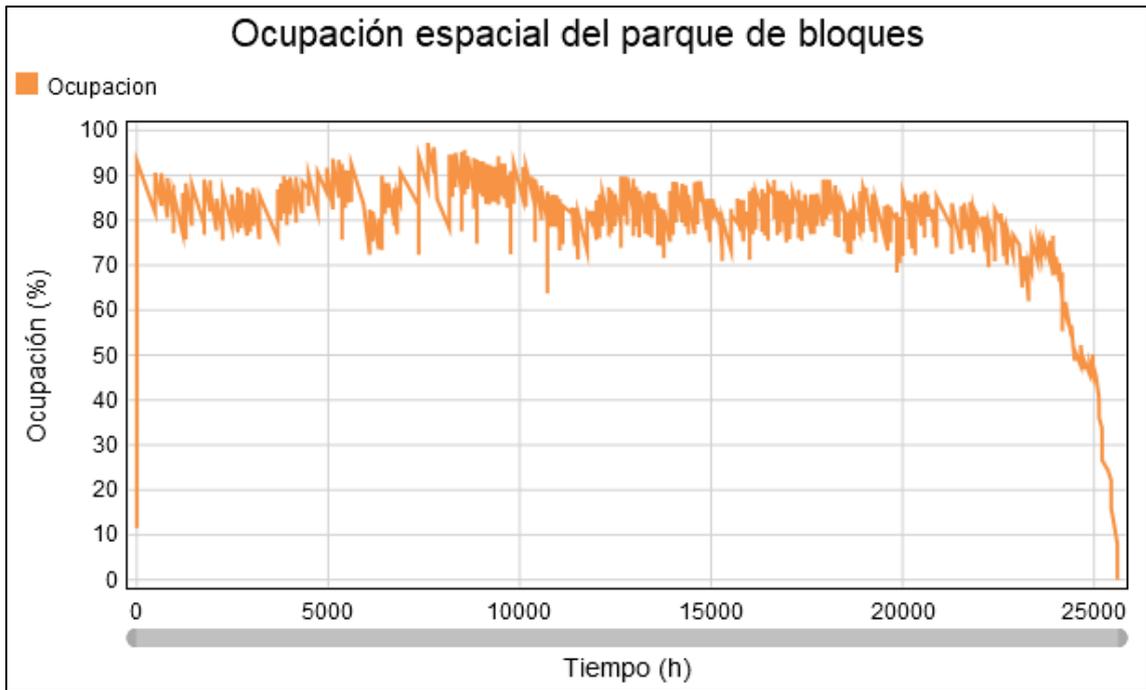


Figura 58. Evolución de la ocupación del parque del escenario 12 (Fuente: elaboración propia).

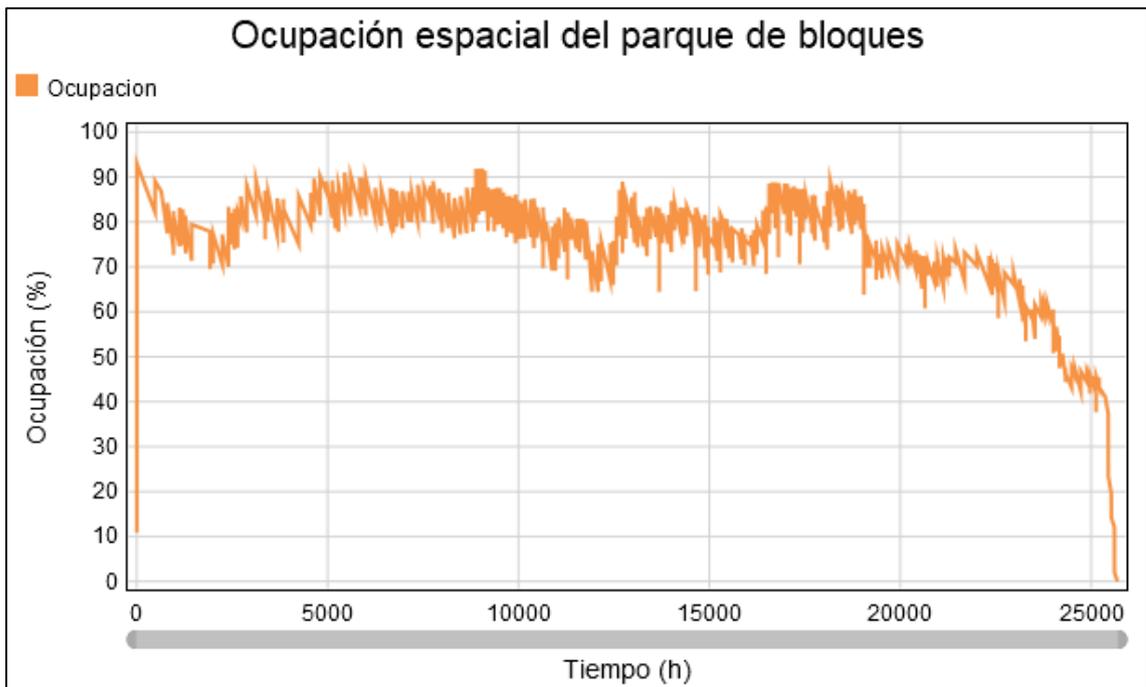


Figura 59. Evolución de la ocupación del parque del escenario 13 (Fuente: elaboración propia).

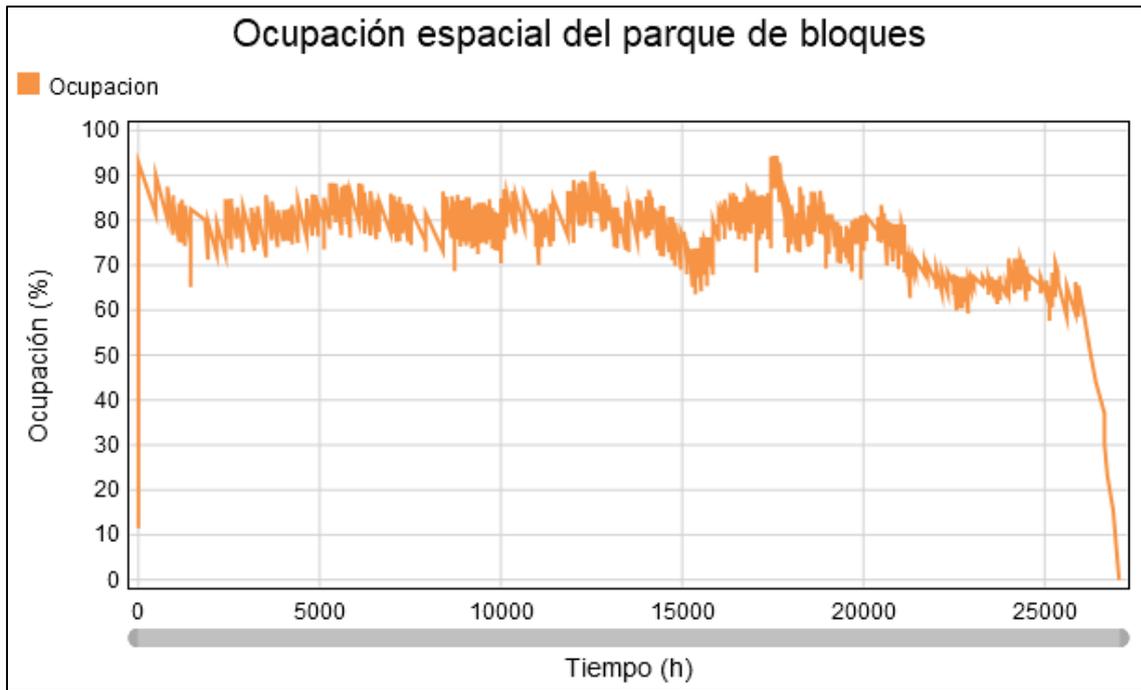


Figura 60. Evolución de la ocupación del parque del escenario 14 (Fuente: elaboración propia).

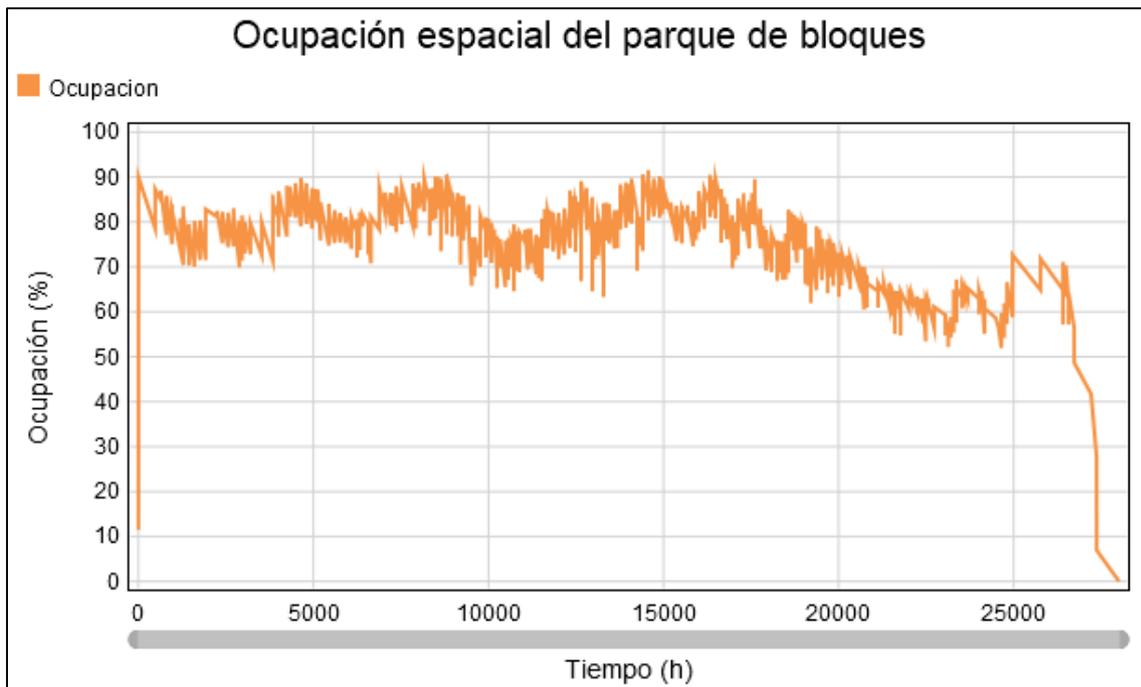


Figura 61. Evolución de la ocupación del parque del escenario 15 (Fuente: elaboración propia).

En los primeros dos escenarios se observa una mejora muy importante del *makespan* con tan solo aumentar cinco metros la longitud o la anchura del parque. El mejor caso de los dos es el de aumentar la anchura cinco metros, por un lado debido a que esto ofrece una mayor superficie y por otro lado porque una gran parte del desaprovechamiento de parque viene dado por el espacio entre los bloques del lado inferior y los del lado superior, sobre todo en el caso de que los dos bloques enfrentados estén con orientaciones alternas. En contar con más anchura se traduce en que más bloques cabrán con una disposición vertical en ambos lados del parque (las menores aristas de los bloques en paralelo con las mayores del parque).

Por otro lado, para los siguientes escenarios en los que se mantiene la superficie constante, vemos que los casos extremos no ofrecen buenos resultados, sobretodo en el caso de contar con un parque de gran longitud pero más estrecho. En ese caso, son pocos los bloques que se pueden colocar en la arista superior, únicamente los de menor tamaño, que tras haberlos procesados todos la ocupación del parque baja drásticamente, como se observa en la gráfica correspondiente al escenario 10. La gran fluctuación de la ocupación se debe a la llegada en distintos instantes de tiempo de los bloques de menor tamaño de los diferentes navíos. Los mejores casos se sitúan en los más cercanos al parque original, siendo el caso de contar con un parque de 92,6 x 34 metros el que mejor *makespan* presenta.

7 CONCLUSIÓN

En este trabajo se ha realizado el desarrollo de un algoritmo de planificación espacial para la ubicación de bloques en un parque de almacenamiento para la construcción de buques. Este mecanismo, a diferencia de otros que han sido propuestos anteriormente en los que se priorizaba la búsqueda de la secuencia de asignación de los bloques, se prioriza ahora la búsqueda de las ubicaciones para posteriormente encontrar un bloque que pueda ser asignado a dicha ubicación de forma óptima.

Cuando se diseña y construye un nuevo tipo de barco, existe una mayor probabilidad de que aparezcan situaciones inesperadas en los flujos de producción. Estos imprevistos causan pérdidas significativas de tiempo, mano de obra, recursos, etc., por tanto, pueden suponer grandes pérdidas económicas, tratándose de proyectos con grandes inversiones. Por tanto, se incrementa la necesidad de disponer de un método de planificación que ofrezca una gran flexibilidad, de manera que se puedan realizar cambios en la programación fácil y rápidamente. Se ha manifestado que la heurística propuesta en este trabajo se ajusta a estos requisitos debido a la priorización de las ubicaciones por encima de la secuenciación de bloques.

La mayor velocidad de planificación que aporta emplear una metodología como esta se apoya también en el manejo de modelos de simulación. Dichos modelos nos permiten obtener un mejor manejo del algoritmo, además de permitir una fácil visualización de los resultados. Esto no solo supone una gran ventaja por la posibilidad de calcular un gran número de escenarios para la planificación en un corto periodo de tiempo, sino que además, su interfaz nos permite que los resultados sean fácilmente interpretados por personal de cualquier nivel.

Como se ha visto en los resultados del primer experimento, el emplear una metodología u otra conlleva a unos resultados diferentes tanto de ocupación como de *makespan*, siendo diferencias de tiempo significativas para este último. Por tanto, se manifiesta la importancia de encontrar un método que proporcione un método más cercano al óptimo, ya que cualquier exceso de tiempo suponen pérdidas para la empresa. En general, con las variaciones que se han ejecutado en la experimentación, se obtienen soluciones adecuadas salvo si se priorizan bloques de menor tamaño, destacando las reglas de prioridad originales frente a las demás en cuanto al plazo y ocupación media.

El emplear una metodología como esta, basada en el la implementación de un algoritmo en un modelo de simulación, no solo optimiza el tiempo de los procesos en sí que se están representando, sino que también reduce la duración del proceso de planificación al poder resolver el problema con la ejecución de un modelo. Esto supone un gran avance respecto a la metodología que se estaba empleando actualmente en Navantia, la cual consistía en simplemente ir encajando manualmente los bloques en el parque en una representación mediante PowerPoint, por lo que difícilmente es un proceso que conducía a un resultado cercano al óptimo, además de consumir mucho más tiempo en la planificación.

Por otro lado, del segundo experimento se pueden extraer conclusiones en cuanto que geometría resulta óptima para el parque de bloques. Los resultados de los primeros dos escenarios, en los que se realiza un aumento de parque, ofrecen una disminución importante del *makespan*. Esta mejora para ambos casos puede llevar a

la realización un análisis de inversión para aumentar la capacidad del parque, comparando si es rentable dicho aumento si el beneficio que se obtiene dada la reducción del plazo es suficiente.

En cuanto a los escenarios del segundo experimento, se observa que la relación entre longitud y anchura del parque que mejores resultados ofrecen está cerca de la actual. Esto nos puede dar una idea para construcciones futuras del astillero, aunque que una geometría sea óptima frente a otra también depende de la geometría de los bloques que compondrán el buque.

8 TRABAJO FUTURO

Con el algoritmo desarrollado en este trabajo se ha obtenido una buena solución para la planificación de los trabajos de los bloques en el parque, teniendo en cuenta las restricciones espaciales. Esto ha sido comprobado mediante un modelo de simulación, el cual nos ha permitido extraer y visualizar la ejecución de las lógicas del proceso y extraer información de ello. Sin embargo, la tarea de la construcción de barcos no se limita únicamente a la construcción de los bloques, como hemos visto previamente, porque el modelo que se ha creado no abarca la totalidad del proceso de construcción de los buques.

Como trabajo a futuro, cabe la posibilidad de implementar la heurística desarrollada en un modelo completo del astillero, que recoja todas las etapas y procesos del proyecto. Con este se puede realizar una planificación general del proyecto en cuanto a los plazos de las actividades y a los recursos de los que dispone el astillero que deben de compartirse en los distintos procesos, ya sea en cuanto a elementos de transporte interno, operarios, etc.

Además, aunque en cuanto a la programación espacial de los bloques con la aplicación del algoritmo y su verificación no sea necesario un modelo de mayor detalle, de cara a la planificación del proyecto sí puede ser interesante una representación menos simplificada, representando a los agentes de las tareas como operarios y herramientas, los tiempos de cada uno de los procesos que se llevan a cabo en la elaboración de los bloques, entre otros detalles.



Figura 62. Ejemplo de modelo del astillero. (Fuente: Unidad Mixta de Investigación, UMI).



Figura 63. Ejemplo de modelo del astillero. (Fuente: Unidad Mixta de Investigación, UMI).

Estos dos aspectos, el contar con un modelo que, por un lado, abarque más procesos del astillero y por otro que a su vez esté más detallado en cuanto a las actividades en las que se descompone cada uno de esos procesos y los recursos que emplea nos aproximan al concepto del *digital twin*, el cual consiste en una réplica digital de la planta de producción, contando con una comunicación constante entre el sistema real y dicho modelo.

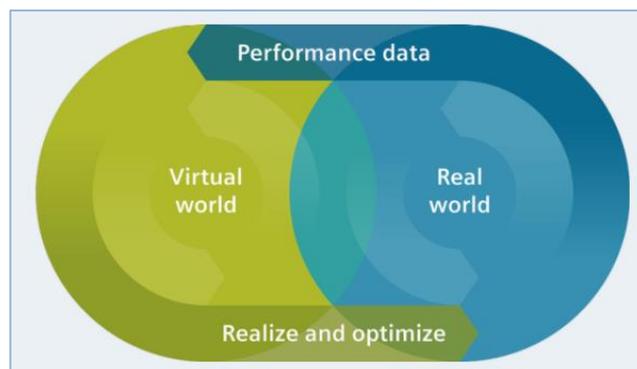


Figura 64. Flujo de información entre el modelo virtual y el sistema real (Fuente: Siemens).

El desarrollo de un *digital twin* y la correspondiente sensorización del astillero ofrecen una serie de ventajas, como permitir una monitorización detallada del astillero, pudiendo analizar en qué estado se encuentra en cualquier momento, poder realizar pruebas sin poner en riesgo el sistema real, obtener datos útiles del funcionamiento del astillero, entre otros.

Además, el contar con un gemelo digital puede dar pie a la implantación de realidad aumentada, siendo esto otra de las bases de la industria 4.0. La utilidad que puede ofrecer esta tecnología en el astillero es múltiple, pudiendo emplearse para la formación de los trabajadores en el entorno del astillero mediante experiencias inmersivas evitando riesgos reales, para la asistencia técnica remota o para cualquier tipo de proceso que se pueda dar guiado con realidad aumentada.

Otras de las líneas a seguir como trabajo a futuro es el desarrollo de nuevos algoritmos a aplicar en la programación de tareas espacial del astillero. Como se ha expuesto anteriormente, existen varias investigaciones en las que se han desarrollado diversos métodos para la resolver la programación de tareas espacial en el sector de la construcción naval. En estos, se desarrollan rutinas basadas en diferentes conceptos como los algoritmos genéticos, métodos heurísticos, etc., además de que en cada uno de ellos se priorizan aspectos diferentes de los bloques, e incluso se puede diferir en priorizar la selección de la ubicación antes que la de la secuencia de bloques, como se ha decidido en este proyecto. Por ello, diseñar métodos basados en otras premisas puede ser de gran ayuda a la hora de obtener una programación más cercana al óptimo, reduciendo así el *makespan* de este proceso y mejorando la ocupación del espacio del que se dispone.

9 BIBLIOGRAFÍA

9.1 Construcción naval e industria 4.0

Pérez García, J.E. (2018). La industria de la construcción naval, la Unión Europea y el siglo XXI. *Información Comercial Española*, N° 901, pp. 115-130.

BRS Group. (2018). *Shipping & Shipbuilding Markets*. Anual Review.

BRS Group. (2019). *Shipping & Shipbuilding Markets*. Anual Review.

UNCTAD. (2019). *Informe sobre el transporte marítimo*.

Clarkson Research (2017). *World Shipyard Monitor*.

Stopford, M. (2008). *Maritime Economics*, 3ª ed. Nueva York, Estados Unidos: Routledge.

Wijnolst, N. y Wegeland, T. (2008). *Shipping Innovation*. Rotterdam, Países Bajos: Delft University Press.

Bruce, G.J. y Garrard, I. (1999). *The Business of Shipbuilding*. Londres, Reino Unido: Lloyds of London Press

Gómez Alén, J. (2019). La construcción naval en España 1950-2019: una introducción general y notas sobre los sistemas constructivos en NAVANTIA/Ferrol. *CEEED*, N° 12, pp. 279-314.

Lorenz, M., Rüßmann, M., Waldner, M. y Gerbert, P. (2015). *Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries*.

Recamán Rivas, A. (2018). Navantia's Shipyard 4.0 model overview. *Ship Science & Technology*, vol. 11, N° 22, pp. 77-85.

Cerezo Preysler, J.L. (2004). El sector de la construcción naval en España. Situación y perspectivas. *Economía Industrial*, N° 355-356, pp. 185-196.

9.2 Programación de tareas y algoritmos heurísticos y metaheurísticos

Pinedo, M.L. (1995). *Scheduling. Theory, Algorithms and Systems*. Englewood Cliffs, Estados Unidos: Prentice-Hall.

Matfeld, D.C. (1996). *Evolutionary Search and The Job Shop*. Heidelberg, Alemania: Physica-Verlag.

Rayward-Smith, V.J., Osman, I.H., Reeves, C.R. y Smith, G.D. (1996). *Modern Heuristic Search Methods*. Chichester, Reino Unido: John Wiley & Sons, Ltd.

Martí, R. (2003). *Procedimientos metaheurísticos en optimización combinatoria*. Valencia, España: Univesitat de València.

Suárez, O. (2011). Una aproximación a las heurísticas y metaheurísticas. *Fondo Editorial Universidad Antonio Narino*, vol. 1, N° 1, pp. 44-51.

García del Valle, A. Apuntes de Simulación de procesos industriales y optimización.

García del Valle, A. Apuntes de Organización de empresas. Capítulo 8: Programación de trabajos.

Franco Gálvez, M.A. *Técnicas heurísticas y metaheurísticas para el problema de la máxima diversidad (Maximum Diversity Problem (MDP))* (tesis de pregrado). Universidad de Murcia, Murcia, España.

Morillo, D., Moreno, L., Díaz, J. (2014). Metodologías analíticas y heurísticas para la solución del problema de programación de tareas con recursos restringidos (RCPSP): una revisión. Parte 1. *Ingeniería y Ciencia*, vol 10, N° 19, pp. 247-271.

Morillo, D., Moreno, L., Díaz, J. (2014). Metodologías analíticas y heurísticas para la solución del problema de programación de tareas con recursos restringidos (RCPSP): una revisión. Parte 2. *Ingeniería y Ciencia*, vol 10, N° 20, pp. 203-227.

Baker, K.R. y Trietsch, D. (2009). *Principles of Sequencing and Scheduling*. Hoboken, Estados Unidos: John Wiley & Sons, Inc.

9.3 Modelado y simulación

Coyle, R.G. (1996). *System Dynamics Modelling: A Practical Approach*. Londres, Reino Unido: Chapman & Hall.

Checkland, P. (1981). *Systems Thinking, Systems Practice*. Chichester, Reino Unido: John Wiley & Sons, Ltd.

Shannon, R.E. (1975). *Systems Simulation: The Art and Science*. Englewood Cliffs, Estados Unidos: Prentice Hall.

Banks, J., Carson, J.S. y Nelson, B.L. (1996). *Discrete-Event System Simulation*, 2ª ed. UpperSaddle River, Estados Unidos: Prentice-Hall.

Pidd, M. (1998). *Computer Simulation in Management Science*, 4ª ed. Chichester, Reino Unido: John Wiley & Sons, Ltd.

Robinson, S. y Higton, N. (1995). Computer simulation for quality and reliability engineering. *Quality and Reliability Engineering International*, vol. 11, pp. 371–377.

Robinson, S. (2004). *Simulation: The Practice of Model Development and Use*. Chichester, Reino Unido: John Wiley & Sons, Ltd.

García del Valle, A. Apuntes de Simulación de procesos industriales y optimización.

9.4 Estado del arte

Lee, K.J., Lee, J.K. y Choi, S.Y. (1996). A Spatial Scheduling System and its Application to Shipbuilding: DAS-CURVE. *Expert Systems With Applications*, vol. 10, N° 3-4, pp. 311-324.

Lee, J.K., Lee, K.J., Park, H.K., Hong, J.S. y Lee, J.S. (1997). Developing scheduling systems for Daewoo Shipbuilding: DAS project. *European Journal of Operational Research*, vol. 97, pp 380-395.

Park, J.S. (2004). *Optimal nesting for structural elements of ship blocks using differential evolution* (tesis de maestría). Universidad Nacional de Seúl, Seúl, Corea del Sur.

Cho, K.K., Chung, K.h., Park, C., Park, J.C. y Kim, H.S. (2002). A spatial scheduling application at the block paint shop in shipbuilding: the HYPOS project. *Production Planning & Control*, vol. 13, N° 4, pp. 342-354.

Shin, J.G., Kwon, O.H. y Ryu, C. (2008). Heuristic and metaheuristic spatial planning of assembly blocks with process schedules in an assembly shop using differential evolution. *Production Planning & Control: The Management of Operations*, vol. 19, N° 6, pp. 605-615.

Chazale, B. (1983). The bottom-left bin-packing heuristic: An efficient implementation. *IEEE Transactions on Computers*, vol. 32, N° 8, pp. 697-707.

Baker, B.S., Coffman, E.G. y Rivest, R.L. (1980). Orthogonal packing in two dimensions. *SIAM Journal of Computing*, vol. 9, N° 4, pp. 846-855.

Jakobs, S. (1996). On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, vol. 88, N° 1, pp. 165-181.

Liu, D.Q. y Teng, H.F. (1999). An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *European Journal of Operations Research*, vol. 112 N° 2, pp. 413-420.

Kwon, B. y Lee, G.M. (2015). Spatial scheduling for large assembly blocks in shipbuilding. *Computers & Industrial Engineering*, vol. 89, pp. 203-212.

9.5 FlexSim

Crespo Pereira, D. Apuntes de Simulación de sistemas logísticos.

En Ferrol, a septiembre de 2020.



Fdo.: Iago Montes Flores



UNIVERSIDADE DA CORUÑA



Escola Politécnica Superior

**TRABAJO FIN DE MÁSTER
CURSO 2019/20**

*SIMULACIÓN Y PROGRAMACIÓN ESPACIAL EN
UN TALLER DE PREFABRICADO DE BLOQUES*

Máster Universitario en Ingeniería Industrial

Anexo I

DATOS

Índice

1 Datos del conjunto de bloques	5
---------------------------------------	---

Índice de tablas

Tabla 9. Conjunto de bloques (Fuente: D. Crespo)	5
--	---

1 DATOS DEL CONJUNTO DE BLOQUES

Tabla 1. Conjunto de bloques (Fuente: D. Crespo)

IDBuque	Referencia	Eslora (m)	Manga (m)	Duración prefabricación (h)	Tiempo inicio disponibilidad (h)
1	13111	9,2	14,28	1600	0
1	13112	9,2	17,365	1120	0
1	13121	13,1	14,1	1760	0
1	13122	13,1	17,27	1440	0
1	13211	9,2	18	640	0
1	13212	9,2	18,6	640	0
1	13221	13,1	18	800	0
1	13222	13,1	18,6	800	0
1	14111	9	12,705	1600	0
1	14112	9	16,675	1440	0
1	14113	11,5	10,855	1600	0
1	14114	11,5	14,95	1440	0
1	14121	9,6	8,785	1440	0
1	14122	9,6	12,805	1280	0
1	14211	9	17,83	1280	0
1	14212	9	18,6	1280	0
1	14221	11,5	17,295	960	0
1	14222	11,5	18,585	960	0
1	14231	9,6	15,585	960	0
1	14232	9,6	17,95	960	0
1	12111	12,1	13,5	1600	0
1	12112	12,1	17,33	1440	0
1	12121	13	14,28	1600	0
1	12122	13	17,365	1440	0
1	12211	13,8	18	1280	0
1	12212	13,8	18,6	1280	0
1	12221	11,3	18,01	960	0
1	12222	11,3	18,6	960	0
1	11111	11,8	17,16	1440	0
1	11231	11,4	18,025	1120	0
1	11232	11,4	18,62	1200	0
1	11221	9	16,455	1600	0
1	11222	9	18,01	1120	0
1	11223	9	18,6	1200	0
1	11211	8,57	17,98	1600	0
1	11212	8,925	18,6	1280	0

Datos

lago Montes Flores

1	18311	9	18,6	1280	0
1	18312	9	17,615	1280	0
1	18321	11,55	18,45	1200	0
1	18322	11,55	17,615	1200	0
1	18323	11,25	16,6	1200	0
1	18331	11,95	18,45	1280	0
1	18332	11,55	18,6	1280	0
1	18333	11,32	16,6	1440	0
1	19411	15,62	18,195	960	0
1	19412	16,64	14,14	880	0
1	19413	12,8	12,8	800	0
1	19414	11	11,16	800	0
1	19415	6,88	5,55	800	0
1	19511	6,485	11,8	480	0
1	19512	6,375	6,82	480	0
1	19513	5,32	3,6	480	0
1	17311	11,4	18,6	1120	0
1	17312	11,4	17,615	1120	0
1	17321	14,4	18,6	960	0
1	17322	14,4	17,615	960	0
1	17411	11,7	8,41	480	0
1	17412	5,1	6	480	0
1	16311	12,4	18,62	1440	0
1	16312	11,2	18,164	1440	0
1	16313	6,7	6,01	480	0
1	16321	11,5	18,6	960	0
1	16322	11,5	17,615	960	0
1	16411	9	6	480	0
1	16412	6,3	6	480	0
1	15111	10,1	6,255	1280	0
1	15112	10,1	9,86	1280	0
1	15211	10,1	12,91	1120	0
1	15212	10,1	16,31	1200	0
1	15213	10,1	18,04	1280	0
1	15121	7,8	3,275	800	0
1	15122	7,8	6,25	800	0
1	15221	7,8	8,97	960	0
1	15222	7,8	12,69	960	0
1	15223	7,8	16,54	1120	0
1	15231	10,5	3,395	1760	0
1	15232	12,79	5,615	1600	0
1	15233	16,675	10,235	1600	0
1	15311	20,26	14,26	480	0
1	15131	13,1	3,055	640	0
2	23111	9,2	14,28	1600	4000

Datos

Iago Montes Flores

2	23112	9,2	17,365	1120	4000
2	23121	13,1	14,1	1760	4000
2	23122	13,1	17,27	1440	4000
2	23211	9,2	18	640	4000
2	23212	9,2	18,6	640	4000
2	23221	13,1	18	800	4000
2	23222	13,1	18,6	800	4000
2	24111	9	12,705	1600	4000
2	24112	9	16,675	1440	4000
2	24113	11,5	10,855	1600	4000
2	24114	11,5	14,95	1440	4000
2	24121	9,6	8,785	1440	4000
2	24122	9,6	12,805	1280	4000
2	24211	9	17,83	1280	4000
2	24212	9	18,6	1280	4000
2	24221	11,5	17,295	960	4000
2	24222	11,5	18,585	960	4000
2	24231	9,6	15,585	960	4000
2	24232	9,6	17,95	960	4000
2	22111	12,1	13,5	1600	4000
2	22112	12,1	17,33	1440	4000
2	22121	13	14,28	1600	4000
2	22122	13	17,365	1440	4000
2	22211	13,8	18	1280	4000
2	22212	13,8	18,6	1280	4000
2	22221	11,3	18,01	960	4000
2	22222	11,3	18,6	960	4000
2	21111	11,8	17,16	1440	4000
2	21231	11,4	18,025	1120	4000
2	21232	11,4	18,62	1200	4000
2	21221	9	16,455	1600	4000
2	21222	9	18,01	1120	4000
2	21223	9	18,6	1200	4000
2	21211	8,57	17,98	1600	4000
2	21212	8,925	18,6	1280	4000
2	28311	9	18,6	1280	4000
2	28312	9	17,615	1280	4000
2	28321	11,55	18,45	1200	4000
2	28322	11,55	17,615	1200	4000
2	28323	11,25	16,6	1200	4000
2	28331	11,95	18,45	1280	4000
2	28332	11,55	18,6	1280	4000
2	28333	11,32	16,6	1440	4000
2	29411	15,62	18,195	960	4000
2	29412	16,64	14,14	880	4000

Datos

lago Montes Flores

2	29413	12,8	12,8	800	4000
2	29414	11	11,16	800	4000
2	29415	6,88	5,55	800	4000
2	29511	6,485	11,8	480	4000
2	29512	6,375	6,82	480	4000
2	29513	5,32	3,6	480	4000
2	27311	11,4	18,6	1120	4000
2	27312	11,4	17,615	1120	4000
2	27321	14,4	18,6	960	4000
2	27322	14,4	17,615	960	4000
2	27411	11,7	8,41	480	4000
2	27412	5,1	6	480	4000
2	26311	12,4	18,62	1440	4000
2	26312	11,2	18,164	1440	4000
2	26313	6,7	6,01	480	4000
2	26321	11,5	18,6	960	4000
2	26322	11,5	17,615	960	4000
2	26411	9	6	480	4000
2	26412	6,3	6	480	4000
2	25111	10,1	6,255	1280	4000
2	25112	10,1	9,86	1280	4000
2	25211	10,1	12,91	1120	4000
2	25212	10,1	16,31	1200	4000
2	25213	10,1	18,04	1280	4000
2	25121	7,8	3,275	800	4000
2	25122	7,8	6,25	800	4000
2	25221	7,8	8,97	960	4000
2	25222	7,8	12,69	960	4000
2	25223	7,8	16,54	1120	4000
2	25231	10,5	3,395	1760	4000
2	25232	12,79	5,615	1600	4000
2	25233	16,675	10,235	1600	4000
2	25311	20,26	14,26	480	4000
2	25131	13,1	3,055	640	4000
3	33111	9,2	14,28	1600	8000
3	33112	9,2	17,365	320	8000
3	33121	13,1	14,1	1760	8000
3	33122	13,1	17,27	320	8000
3	33211	9,2	18	320	8000
3	33212	9,2	18,6	320	8000
3	33221	13,1	18	320	8000
3	33222	13,1	18,6	320	8000
3	34111	9	12,705	1600	8000
3	34112	9	16,675	320	8000
3	34113	11,5	10,855	1600	8000

Datos

Iago Montes Flores

3	34114	11,5	14,95	320	8000
3	34121	9,6	8,785	1440	8000
3	34122	9,6	12,805	320	8000
3	34211	9	17,83	320	8000
3	34212	9	18,6	320	8000
3	34221	11,5	17,295	320	8000
3	34222	11,5	18,585	320	8000
3	34231	9,6	15,585	320	8000
3	34232	9,6	17,95	320	8000
3	32111	12,1	13,5	1600	8000
3	32112	12,1	17,33	320	8000
3	32121	13	14,28	1600	8000
3	32122	13	17,365	320	8000
3	32211	13,8	18	320	8000
3	32212	13,8	18,6	320	8000
3	32221	11,3	18,01	320	8000
3	32222	11,3	18,6	320	8000
3	31111	11,8	17,16	1440	8000
3	31231	11,4	18,025	320	8000
3	31232	11,4	18,62	320	8000
3	31221	9	16,455	1600	8000
3	31222	9	18,01	320	8000
3	31223	9	18,6	320	8000
3	31211	8,57	17,98	1600	8000
3	31212	8,925	18,6	320	8000
3	38311	9	18,6	320	8000
3	38312	9	17,615	320	8000
3	38321	11,55	18,45	320	8000
3	38322	11,55	17,615	320	8000
3	38323	11,25	16,6	320	8000
3	38331	11,95	18,45	320	8000
3	38332	11,55	18,6	320	8000
3	38333	11,32	16,6	320	8000
3	39411	15,62	18,195	960	8000
3	39412	16,64	14,14	880	8000
3	39413	12,8	12,8	800	8000
3	39414	11	11,16	800	8000
3	39415	6,88	5,55	800	8000
3	39511	6,485	11,8	480	8000
3	39512	6,375	6,82	480	8000
3	39513	5,32	3,6	480	8000
3	37311	11,4	18,6	320	8000
3	37312	11,4	17,615	320	8000
3	37321	14,4	18,6	320	8000
3	37322	14,4	17,615	320	8000

Datos

lago Montes Flores

3	37411	11,7	8,41	480	8000
3	37412	5,1	6	480	8000
3	36311	12,4	18,62	320	8000
3	36312	11,2	18,164	320	8000
3	36313	6,7	6,01	480	8000
3	36321	11,5	18,6	320	8000
3	36322	11,5	17,615	320	8000
3	36411	9	6	480	8000
3	36412	6,3	6	480	8000
3	35111	10,1	6,255	1280	8000
3	35112	10,1	9,86	320	8000
3	35211	10,1	12,91	320	8000
3	35212	10,1	16,31	320	8000
3	35213	10,1	18,04	320	8000
3	35121	7,8	3,275	800	8000
3	35122	7,8	6,25	320	8000
3	35221	7,8	8,97	320	8000
3	35222	7,8	12,69	320	8000
3	35223	7,8	16,54	320	8000
3	35231	10,5	3,395	1760	8000
3	35232	12,79	5,615	1600	8000
3	35233	16,675	10,235	1600	8000
3	35311	20,26	14,26	480	8000
3	35131	13,1	3,055	640	8000
4	43111	9,2	14,28	1600	12000
4	43112	9,2	17,365	320	12000
4	43121	13,1	14,1	1760	12000
4	43122	13,1	17,27	320	12000
4	43211	9,2	18	320	12000
4	43212	9,2	18,6	320	12000
4	43221	13,1	18	320	12000
4	43222	13,1	18,6	320	12000
4	44111	9	12,705	1600	12000
4	44112	9	16,675	320	12000
4	44113	11,5	10,855	1600	12000
4	44114	11,5	14,95	320	12000
4	44121	9,6	8,785	1440	12000
4	44122	9,6	12,805	320	12000
4	44211	9	17,83	320	12000
4	44212	9	18,6	320	12000
4	44221	11,5	17,295	320	12000
4	44222	11,5	18,585	320	12000
4	44231	9,6	15,585	320	12000
4	44232	9,6	17,95	320	12000
4	42111	12,1	13,5	1600	12000

Datos

Iago Montes Flores

4	42112	12,1	17,33	320	12000
4	42121	13	14,28	1600	12000
4	42122	13	17,365	320	12000
4	42211	13,8	18	320	12000
4	42212	13,8	18,6	320	12000
4	42221	11,3	18,01	320	12000
4	42222	11,3	18,6	320	12000
4	41111	11,8	17,16	1440	12000
4	41231	11,4	18,025	320	12000
4	41232	11,4	18,62	320	12000
4	41221	9	16,455	1600	12000
4	41222	9	18,01	320	12000
4	41223	9	18,6	320	12000
4	41211	8,57	17,98	1600	12000
4	41212	8,925	18,6	320	12000
4	48311	9	18,6	320	12000
4	48312	9	17,615	320	12000
4	48321	11,55	18,45	320	12000
4	48322	11,55	17,615	320	12000
4	48323	11,25	16,6	320	12000
4	48331	11,95	18,45	320	12000
4	48332	11,55	18,6	320	12000
4	48333	11,32	16,6	320	12000
4	49411	15,62	18,195	960	12000
4	49412	16,64	14,14	880	12000
4	49413	12,8	12,8	800	12000
4	49414	11	11,16	800	12000
4	49415	6,88	5,55	800	12000
4	49511	6,485	11,8	480	12000
4	49512	6,375	6,82	480	12000
4	49513	5,32	3,6	480	12000
4	47311	11,4	18,6	320	12000
4	47312	11,4	17,615	320	12000
4	47321	14,4	18,6	320	12000
4	47322	14,4	17,615	320	12000
4	47411	11,7	8,41	480	12000
4	47412	5,1	6	480	12000
4	46311	12,4	18,62	320	12000
4	46312	11,2	18,164	320	12000
4	46313	6,7	6,01	480	12000
4	46321	11,5	18,6	320	12000
4	46322	11,5	17,615	320	12000
4	46411	9	6	480	12000
4	46412	6,3	6	480	12000
4	45111	10,1	6,255	1280	12000

Datos

lago Montes Flores

4	45112	10,1	9,86	320	12000
4	45211	10,1	12,91	320	12000
4	45212	10,1	16,31	320	12000
4	45213	10,1	18,04	320	12000
4	45121	7,8	3,275	800	12000
4	45122	7,8	6,25	320	12000
4	45221	7,8	8,97	320	12000
4	45222	7,8	12,69	320	12000
4	45223	7,8	16,54	320	12000
4	45231	10,5	3,395	1760	12000
4	45232	12,79	5,615	1600	12000
4	45233	16,675	10,235	1600	12000
4	45311	20,26	14,26	480	12000
4	45131	13,1	3,055	640	12000
5	53111	9,2	14,28	1600	16000
5	53112	9,2	17,365	320	16000
5	53121	13,1	14,1	1760	16000
5	53122	13,1	17,27	320	16000
5	53211	9,2	18	320	16000
5	53212	9,2	18,6	320	16000
5	53221	13,1	18	320	16000
5	53222	13,1	18,6	320	16000
5	54111	9	12,705	1600	16000
5	54112	9	16,675	320	16000
5	54113	11,5	10,855	1600	16000
5	54114	11,5	14,95	320	16000
5	54121	9,6	8,785	1440	16000
5	54122	9,6	12,805	320	16000
5	54211	9	17,83	320	16000
5	54212	9	18,6	320	16000
5	54221	11,5	17,295	320	16000
5	54222	11,5	18,585	320	16000
5	54231	9,6	15,585	320	16000
5	54232	9,6	17,95	320	16000
5	52111	12,1	13,5	1600	16000
5	52112	12,1	17,33	320	16000
5	52121	13	14,28	1600	16000
5	52122	13	17,365	320	16000
5	52211	13,8	18	320	16000
5	52212	13,8	18,6	320	16000
5	52221	11,3	18,01	320	16000
5	52222	11,3	18,6	320	16000
5	51111	11,8	17,16	1440	16000
5	51231	11,4	18,025	320	16000
5	51232	11,4	18,62	320	16000

Datos

Iago Montes Flores

5	51221	9	16,455	1600	16000
5	51222	9	18,01	320	16000
5	51223	9	18,6	320	16000
5	51211	8,57	17,98	1600	16000
5	51212	8,925	18,6	320	16000
5	58311	9	18,6	320	16000
5	58312	9	17,615	320	16000
5	58321	11,55	18,45	320	16000
5	58322	11,55	17,615	320	16000
5	58323	11,25	16,6	320	16000
5	58331	11,95	18,45	320	16000
5	58332	11,55	18,6	320	16000
5	58333	11,32	16,6	320	16000
5	59411	15,62	18,195	960	16000
5	59412	16,64	14,14	880	16000
5	59413	12,8	12,8	800	16000
5	59414	11	11,16	800	16000
5	59415	6,88	5,55	800	16000
5	59511	6,485	11,8	480	16000
5	59512	6,375	6,82	480	16000
5	59513	5,32	3,6	480	16000
5	57311	11,4	18,6	320	16000
5	57312	11,4	17,615	320	16000
5	57321	14,4	18,6	320	16000
5	57322	14,4	17,615	320	16000
5	57411	11,7	8,41	480	16000
5	57412	5,1	6	480	16000
5	56311	12,4	18,62	320	16000
5	56312	11,2	18,164	320	16000
5	56313	6,7	6,01	480	16000
5	56321	11,5	18,6	320	16000
5	56322	11,5	17,615	320	16000
5	56411	9	6	480	16000
5	56412	6,3	6	480	16000
5	55111	10,1	6,255	1280	16000
5	55112	10,1	9,86	320	16000
5	55211	10,1	12,91	320	16000
5	55212	10,1	16,31	320	16000
5	55213	10,1	18,04	320	16000
5	55121	7,8	3,275	800	16000
5	55122	7,8	6,25	320	16000
5	55221	7,8	8,97	320	16000
5	55222	7,8	12,69	320	16000
5	55223	7,8	16,54	320	16000
5	55231	10,5	3,395	1760	16000

Datos

lago Montes Flores

5	55232	12,79	5,615	1600	16000
5	55233	16,675	10,235	1600	16000
5	55311	20,26	14,26	480	16000
5	55131	13,1	3,055	640	16000

En Ferrol, a septiembre de 2020.



Fdo.: Iago Montes Flores

