

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

# **Sistema Multiagente aplicado al Problema de Asignación de Barcos a Muelles**

**ROMINA DENISE LÓPEZ LÓPEZ  
IVONNE ANTONIA INÉS RAMÍREZ FUSTER**

INFORME FINAL DEL PROYECTO  
PARA OPTAR AL TÍTULO PROFESIONAL DE  
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

**DICIEMBRE 2012**

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

## **Sistema Multiagente aplicado al Problema de Asignación de Barcos a Muelles**

**ROMINA DENISE LÓPEZ LÓPEZ  
IVONNE ANTONIA INÉS RAMÍREZ FUSTER**

Profesor Guía: **Claudio Cubillos Figueroa**

Profesor Co-referente: **Rodolfo Villarroel Acevedo**

Carrera: **Ingeniería de Ejecución en Informática**

**DICIEMBRE 2012**

Agradecemos a nuestros padres por habernos apoyado durante todos estos años de estudio, a nuestros amigos y compañeros por la ayuda incondicional, a nuestros profesores por los conocimientos entregados en este tiempo, especialmente a nuestro profesor guía por su ayuda en la realización de nuestro trabajo de título.

## Resumen

Este informe trata sobre el problema conocido como Berth Allocation Problem (BAP). BAP es un problema de optimización que resuelve el atraque de barcos, tratando de minimizar el tiempo y costo. Existen diferentes aproximaciones a este problema, ellas son: modelo discreto, continuo e híbrido, y modelo estático y dinámico. Este trabajo en particular trata sobre el BAP discreto y dinámico.

A su vez, hay diversas formas de resolver BAP, este documento propone una manera de resolución mediante sistemas multiagente, basándose en otros proyectos similares para su desarrollo. Dicho problema es resuelto mediante sistemas multiagente, basándose en otros proyectos similares para su desarrollo e implementación, obteniendo los resultados esperados según los objetivos planteados.

*Palabras-claves: BerthAllocationProblem, optimización, discreto, dinámico, sistemas multiagente.*

## Abstract

This paper treats about the Berth Allocation Problem (BAP). BAP is an optimization problem that solves the berthing of vessels, trying to minimize time and cost. There are different approaches to this problem, there are discrete model, continuous and hybrid, and static model and dynamics. This particular paper treats about the discrete and dynamic BAP.

In turn, there are different forms of solving BAP, this paper proposes a way of resolution through a multi- agent System, based on similar projects for their development. This problem is solved by multi-agent systems, based on similar projects for their development and implementation, obtaining the expected results according to the objectives.

*Key Words: Berth Allocation Problem, optimization, discrete, dynamic, multiagent systems.*

# Índice

## Capítulo 1: Aspectos Preliminares ..... 1

1.1	Introducción .....	1
1.2	Objetivos .....	2
1.2.1	Objetivo General .....	2
1.2.2	Objetivos Específicos.....	2
1.3	Planificación.....	3
1.4	Metodología .....	4
1.5	Estructura del Documento.....	4

## Capítulo 2: Tecnología Multiagente ..... 6

2.1	Definición de Agente .....	6
2.2	Agentes Inteligentes .....	6
2.3	Comparación de los Agentes.....	7
2.3.1	Agente v/s Objeto.....	7
2.3.2	Agente v/s Sistemas Expertos .....	8
2.4	Tipología de Agentes .....	8
2.5	Arquitecturas Para Construir Agentes.....	9
2.5.1	Arquitecturas Deliberativas.....	9
2.5.2	Arquitecturas Reactivas.....	9
2.5.3	Arquitecturas Híbridas .....	10
2.6	Arquitectura Multiagente .....	10
2.7	Arquitectura FIPA.....	12
2.8	Infraestructura de Agentes .....	14
2.8.1	Ontologías .....	14
2.8.2	Lenguajes de Comunicación entre Agentes (ACLs) .....	15
2.8.2.1	KQM (Knowledge Query Manipulation Language) .....	16
2.8.2.2	FIPA ACL .....	18
2.8.3	Protocolos de Interacción .....	18
2.9	Metodologías de Desarrollo .....	19
2.9.1	AAII/BDI .....	19
2.9.2	MAS-CommonKADS .....	20
2.9.3	GAIA.....	20
2.9.4	MaSE.....	20

2.9.5 PASSI.....	21
<b>Capítulo 3: Berth Allocation Problem .....</b>	<b>22</b>
3.1 Introducción al Problema .....	22
3.2 Formulación Matemática.....	25
3.3 Análisis de la Literatura .....	27
3.4 Datos de Pruebas .....	31
3.5 Algoritmos de Solución.....	31
3.5.1 MixedInteger Linear Programming .....	32
3.5.2 Branch and Bound.....	32
3.5.3 Genetic Algorithms .....	32
3.5.4 Simulated Annealing .....	32
3.5.5 Método de Relajación Lagrangeana .....	33
3.5.6 Heurística Constructiva .....	33
3.5.6.1 Algoritmo Greedy.....	33
3.5.7 TabuSearch (Búsqueda Tabú) .....	33
3.5.8 Algoritmo ADARTW.....	34
3.6 Arquitectura Multiagente para el BAP .....	34
3.6.1 Arquitectura MADARP.....	35
3.6.2 Arquitectura MA en BAP.....	38
3.6.3 Arquitectura DOSS (Distributed Omni Search Strategy).....	39
3.7 Trabajos relacionados.....	40
3.7.1 BAP Estático con Sistemas Multiagente .....	40
3.7.2 D-DARP con Sistemas Multiagente.....	41
<b>Capítulo 4: Solución Propuesta .....</b>	<b>43</b>
4.1 Problema Estático o Dinámico .....	43
4.2 Modelo Discreto, Continuo o Híbrido.....	43
4.3 Metodología de Desarrollo Elegida.....	44
4.4 Herramientas de Desarrollo.....	44
4.5 Diseño de la Arquitectura Propuesta.....	45
4.5.1 Agente Barco.....	47
4.5.2 Agente Muelle .....	47
4.5.3 Agente Simulador Barco .....	47
4.5.4 Agente Simulador Muelle .....	48

4.5.5 Agente Programador Atrake.....	48
4.5.6 Agente Planificador.....	49
4.5.7 Agente Petición Atrake .....	49
4.6 Diseño de la Heurística Propuesta.....	50
4.6.1 Especificación de Conceptos.....	50
4.6.1.1 Clientes de Entrada y Clientes de Salida.....	51
4.6.1.2 Estados del Vehículo .....	51
4.6.1.3 Bloques de Planificación.....	52
4.6.1.4 Ventanas de Tiempo.....	52
4.6.2 Algoritmo de Inserción.....	53
4.6.3 Algoritmo de Inserción para el BAP .....	56
<b>Capítulo 5: Diseño del Sistema .....</b>	<b>61</b>
5.1 Modelo de Requerimientos del Sistema.....	61
5.1.1 Diagrama de Identificación de Agentes .....	61
5.1.2 Diagrama de Identificación de Roles .....	64
5.1.3 Diagrama de Identificación de Tareas.....	67
5.2 Modelo de Sociedad de Agentes .....	70
5.2.1 Diagrama de Descripción de Ontología de Dominio .....	70
5.2.2 Diagrama de Descripción de Ontología de Comunicación .....	70
5.2.3 Diagrama de Descripción de Roles .....	70
5.3 Modelo de Sociedad de Agentes .....	74
5.3.1 Diagrama de Definición de la Estructura Multiagente .....	74
5.3.2 Diagrama de Definición de la Estructura del Agente Individual.....	74
<b>Capítulo 6: Diseño de Pruebas.....</b>	<b>77</b>
6.1 Planificación de la Pruebas .....	77
6.2 Diseño de las Pruebas.....	78
6.3 Datos de Entrada .....	78
<b>Capítulo 7: Implementación de Pruebas.....</b>	<b>81</b>
7.1 Pruebas Estáticas.....	81
7.2 Pruebas Dinámicas .....	84
7.2.1 Evento de desperfecto en el muelle.....	84
7.2.2 Eventos de Cancelación de Barcos.....	86
7.2.3 Evento de Cambios de Tiempo en los Barcos.....	87

7.2.4 Integración de Eventos .....	88
7.3 Pruebas de Coordinación de Tiempos .....	89
7.4 Pruebas al Sistema de Monitoreo .....	90
7.4.1 Interfaz Gráfica .....	90
7.4.2 Evaluación del Módulo de Monitoreo.....	91
7.5 Archivos de Salida .....	93
<b>Capítulo 8: Conclusión .....</b>	<b>96</b>
<b>Capítulo 9: Referencias Bibliográficas y Bibliografía.....</b>	<b>98</b>

# Índice de Figuras

Figura 2.1 Arquitectura FIPA [52].....	14
Figura 3.1 Representación espacio tiempo [17] .....	25
Figura 3.2 Modelo espacial del muelle [18]......	28
Figura 3.3 Arquitectura de agentes MADARP [22] .....	35
Figura 3.4 Infraestructura para agentes. [29].....	38
Figura 4.1 Arquitectura MA BAP.....	46
Figura 4.2 Arquitectura MABAP Propuesta.....	50
Figura 4.3 Estructura de un bloque de planificación. [29] .....	52
Figura 4.4 Construcción de una ventana de tiempo. [22].....	53
Figura 4.5 Evaluación para la inserción de un nuevo cliente. [22] .....	55
Figura 4.6 Representación de los elementos necesarios para la nueva inserción [22]......	56
Figura 4.7 Bloque de Planificación para el BAP .....	58
Figura 4.8 Ventanas de Tiempo asociadas al Barco.....	58
Figura 4.9 Ventana de Tiempo resultante para el BAP .....	59
Figura 4.10 Llegada de un Barco a un Muelle .....	59
Figura 5.1 Diagrama de identificación de agentes .....	63
Figura 5.2 Escenario de asignación de una petición exitosa .....	65
Figura 5.3 Escenario de cancelación de una notificación.....	66
Figura 5.4 Escenario de una re planificación por atraso.....	66
Figura 5.5 Especificación de tareas del agente berth.....	67
Figura 5.6 Especificación de tareas del agente barco .....	68
Figura 5.7 Especificación de tareas del agente simulador berth.....	68
Figura 5.8 Especificación de tareas del agente simulador barco .....	69
Figura 5.9 Descripción de ontología de dominio .....	71
Figura 5.10 Descripción de ontología de dominio .....	72
Figura 5.11 Descripción de roles .....	73
Figura 5.12 Diagrama de estructura multiagente.....	73
Figura 5.13 Diagrama de estructura individual agente planner.....	73
Figura 6.1 Archivo de datos de entrada para un barco .....	79
Figura 6.2 Archivo de datos de entrada para un muelle. ....	79
Figura 7.1 Gráfico de cantidad de barcos atendidos.....	83
Figura 7.2 Gráfico cantidad de muelles usados.....	83
Figura 7.3 Gráfico resultados de los costos totales. ....	84
Figura 7.4 Interfaz del módulo de monitoreo.....	89
Figura 7.5 Resultados de ventanas de tiempo.....	93
Figura 7.6 Imagen de costos asociados a los berths. ....	94
Figura 7.7 Resultados de la secuencia de barco por cada berth y los tiempos asociados.....	94
Figura 7.8 Archivos resumen de ejecuciones. ....	95

Figura 7.9 Archivos con el resultado de la solicitud hecha por un barco .....95

## Índice de Tablas

Tabla 1.1 Primera etapa de planificación.....	3
Tabla 1.2 Segunda etapa de planificación. ....	4
Tabla 2.1 Elementos de un mensaje reservado de KQML [1]. ....	17
Tabla 2.2 Elementos de un mensaje reservado FIPA ACL [1]. ....	17
Tabla 7.1 Ejemplo de asignación de barcos a muelles en sus eventos de arrival y departure. ...	81
Tabla 7.2 Resumen de datos de muelles con desperfectos. ....	85
Tabla 7.3 Ejemplo de secuencia de barcos luego de algún desperfecto. ....	85
Tabla 7.4 Resumen de barcos ccancelados. ....	86
Tabla 7.5 Ejemplo de secuencia de barcos luego de cancelaciones de barcos. ....	87
Tabla 7.6 Resumen de barcos con cambios de tiempo ....	88
Tabla 7.7 Ejemplo de secuencia de barcos luego de cambios de tiempo en los barcos. ....	88
Tabla 7.8 Resumen completo con todos los eventos posibles ....	19
Tabla 7.9 Ejemplo de eventos gatillados y respuesta según el monitoreo.....	91

# Capítulo 1: Aspectos Preliminares

## 1.1 Introducción

El tráfico marítimo dentro de todo el mundo y en especial en Chile, es muy importante, ya que a lo largo de él hay una gran cantidad de puertos (como lo son el puerto de San Antonio, de Valparaíso, de Antofagasta y muchos otros) dedicados al traslado de mercancías tanto dentro como fuera del país. Dentro de estos puertos, hay algunos que están orientados al flujo de mercancías, información, por lo que hay que otorgar espacio dentro del muelle a embarcaciones destinadas a dichos propósitos; por esto, es relevante que estas tareas sean realizadas de forma óptima, ya que la actividad de los puertos es uno de los principales nodos de la red de transporte marítimo.

Sin embargo, para que los barcos puedan atracar dentro de una sección específica del muelle, se deben tener varias consideraciones, como lo son el tamaño de los barcos, el tráfico, la cantidad de muelles existentes dentro del puerto, etc., por lo que surge una problemática al momento de asignarlos, éste se conoce como Berth Allocation Problem (BAP), o en su traducción al español, problema de asignación de barcos a muelle. Una mala asignación provocaría efectos negativos dentro del puerto, aumentando sus costos y mal gastando el tiempo.

La logística del BAP, está orientada al atraque o amarre de naves llegando a puerto, a un muelle, con esto también logrando que en la gestión del puerto las embarcaciones atraquen lo antes posible, la maquinaria encargada de realizar la carga y descarga de containers cumpla su labor en el menor tiempo y el costo del trasbordo de contenedores sea mínimo.

Por todo lo descrito anteriormente, es de vital importancia que se busque y utilice una logística que permita que todas estas tareas sean realizadas, como antes se mencionó, de manera eficiente, brindando una efectiva solución para el BAP, optimizando tiempos y costos asociados y entregándoles una asignación a los barcos, sin que se provoquen conflictos y puedan llevar a cabo su proceso de descarga y/o carga de la mercancía en cuestión.

Para dar una solución, existe el enfoque dinámico y estático, de los cuales, se considerará una visión dinámica del problema, brindando la posibilidad de la ejecución del sistema en tiempo real y con posibilidades de re planificación, hechos importantes que ayudan a las situaciones y eventos que ocurren en la realidad.

Para implementar esta logística, se utilizará la tecnología de sistemas multiagente. Un sistema multiagente, básicamente se compone de agentes que interactúan para lograr una meta común. Dentro de la principales características que tienen estos agentes, es que aparte

de ser agentes inteligentes, tienen la capacidad de interactuar permanentemente con otros agentes para lograr el propósito final, y para ello, ellos se comunican, cooperan y se coordinan, características que utilizan tanto para cumplir sus objetivos propios, como para cumplir aquellos que tienen que resolver en conjunto para poder lograr llegar a la solución del problema que tienen que resolver.

Si bien se ha abordado este tema en varias ocasiones, brindando diferentes soluciones, este proyecto se basará en algunos de ellos, dando énfasis al dinamismo de la solución, a la arquitectura para el diseño del sistema y a la parte de planificación, vista en la heurística como método de solución.

Tomando en cuenta la complejidad de este problema y la importancia de una buena solución, este proyecto ha sido separado en varias etapas, dentro de ellas se encuentran los estudios y explicaciones previos del tema, la determinación de objetivos, el análisis, el diseño de la logística del problema y modelos, implementación, ejecución y solución tangible del problema, basado en una arquitectura multiagente para el Berth Allocation Problem (BAP), y en síntesis a todo esto, las conclusiones pertinentes.

## **1.2 Objetivos**

### **1.2.1 Objetivo General**

Desarrollar una solución dinámica del problema de asignación de barcos a muelles o BerthAllocationProblem (BAP), basándose en una arquitectura de sistemas multiagente.

### **1.2.2 Objetivos Específicos**

Para desglosar el principal objetivo del proyecto, se plantearon los siguientes:

- Investigar y analizar el problema de BAP, para conocer las características que van relacionadas al problema.
- Investigar sobre la arquitectura de los sistemas multiagente, obteniendo sus definiciones y características fundamentales para su utilización en el BAP.
- Desarrollar un sistema multiagente que solucione el BAP basado en la arquitectura MADARP.
- Realizar pruebas necesarias al programa ejecutable para dar la mejor solución al problema.

### 1.3 Planificación

La planificación que se mostrará a continuación es solo un estimado de cómo se pretende trabajar este proyecto y de sus fechas de término. Ésta se dividió en dos etapas, la primera corresponde al desarrollo de investigación, análisis y diseño (ver tabla 1.1); mientras la segunda corresponde a la implementación del sistema y sus pruebas (ver tabla 1.2).

**Tabla 1.1 Primera etapa de planificación**

Id.	Nombre de tarea	Comienzo	Fin	Duración	mar 2012				abr 2012				may 2012				jun 2012				
					4-3	11-3	18-3	25-3	1-4	8-4	15-4	22-4	29-4	6-5	13-5	20-5	27-5	3-6	10-6	17-6	
1	Investigación y obtención de información	05-03-2012	16-03-2012	10d	■																
2	Recopilación de información	12-03-2012	23-03-2012	10d	■																
3	Análisis de problema	12-03-2012	30-03-2012	15d	■																
4	Diseño de la solución	20-03-2012	06-04-2012	14d					■												
5	Confección informe	20-03-2012	13-04-2012	19d					■												
6	Corrección errores	10-04-2012	16-04-2012	5d									■								
7	Verificación de fuentes	16-04-2012	17-04-2012	2d													■				
8	Revisión final	17-04-2012	18-04-2012	2d													■				
9	20-04-2012	20-04-2012	20-04-2012	1d													■				
10	Presentación	23-04-2012	23-04-2012	1d													■				
11	Revisión informe	24-04-2012	18-05-2012	19d									■								
12	Corrección errores señalados	21-05-2012	20-06-2012	23d													■				
13	Entrega	22-06-2012	22-06-2012	1d													■				

**Tabla 1.2 Segunda etapa de planificación.**

Id.	Nombre de tarea	Comienzo	Fin	Duración	ago 2012				sep 2012				oct 2012				nov 2012			
					5-8	12-8	19-8	26-8	2-9	9-9	16-9	23-9	30-9	7-10	14-10	21-10	28-10	4-11	11-11	18-11
1	Elaborar cambios necesarios a la heurística	01-08-2012	06-08-2012	4d	■															
2	Integrar cambios realizadas a la heurística existente	06-08-2012	10-08-2012	5d	■															
3	Presentación prototipo funcional	14-08-2012	14-08-2012	0d	◆															
4	Implementar agentes monitoreo	15-08-2012	03-09-2012	14d		■	■	■												
5	Implementar agentes planificación	13-09-2012	02-10-2012	14d					■	■	■									
6	Implementar interfaces	12-10-2012	31-10-2012	14d							■	■	■							
7	Revisar y cambiar prototipo	01-11-2012	05-11-2012	3d														■		
8	Agregar agentes de monitoreo	06-11-2012	19-11-2012	10d														■	■	
9	Definir datos entradas	19-11-2012	21-11-2012	3d															■	
10	Realizar pruebas	22-11-2012	27-11-2012	4d															■	
11	Registrar resultados	28-11-2012	29-11-2012	2d															■	
12	Comparar resultados obtenidos con anteriores	28-11-2012	29-11-2012	2d															■	
13	Presentar sistema	30-11-2012	30-11-2012	1d															■	

## 1.4 Metodología

Se ha decidido trabajar con la metodología RUP, ya que es el proceso de desarrollo más general de los existentes actualmente. No es realista utilizar modelos lineales de desarrollo, ya que un modelo iterativo permite una comprensión creciente del sistema en desarrollo. Éste trata las “mejores prácticas” de desarrollo de software como obtener los requerimientos, organizarlos, documentar y otros. También éste aborda las tareas más riesgosas primero, con esto dejando las más simples para después, así reduce riesgos y se obtiene un subsistema ejecutable tempranamente.

## 1.5 Estructura del Documento

Los contenidos que se presentan a continuación en este documento muestran el análisis y el resultado de una investigación sobre el BAP, todo a partir de la entrega de conocimientos teóricos que respalden las decisiones a tomar en el desarrollo del proyecto en cuestión. Estos contenidos se encuentran estructurados de la siguiente manera:

El primer capítulo es el ya leído, el cual trata de introducir hacia el tema del BAP y de la tecnología multiagente, además de la forma en que se desea ejecutar el proyecto.

En el segundo capítulo, se da una introducción acerca de los contenidos relacionados con la tecnología multiagente, como lo son la definición de agentes, características, sistemas multiagente y tecnologías asociadas a su desarrollo e implementación. Todo lo anterior permite que de manera general esté especificado el marco conceptual de la tecnología de sistemas multiagente, brindando con ello la base del desarrollo y logro del proyecto.

El tercer capítulo, trata sobre el contenido general del problema de optimización BAP, la cual es un área de investigación muy extensa ya que de ella es el tema primordial del proyecto. En éste se profundizan las características principales, las posibles propuestas y soluciones del problema y la importancia que tiene éste dentro de la realidad de los puertos.

El cuarto capítulo, presenta las decisiones que se debieron tomar para llevar a cabo el posterior diseño del sistema, dentro de estas decisiones se encuentra solucionar el BAP en su versión dinámica y discreta, la arquitectura propuesta y la heurística, incluyendo en ella el algoritmo de inserción a utilizar, junto con ello, la metodología implementada y las herramientas necesarias para la ejecución del proyecto. Además de ello se revisa un resumen acerca del trabajo que se utilizó como base y se ha continuado.

El quinto capítulo, presenta formalmente el diseño de la implementación del sistema, incluyendo algunos de los diagramas necesarios para la implementación del sistema, entre los que se encuentran los modelos de dominio, los modelos de sociedad de agentes y los de implementación de agentes.

En el capítulo seis, se muestra un modelo de las pruebas que se le efectuarán al sistema para comprobar su funcionalidad y junto con ello la planificación que se hizo para hacerlas efectivas.

El penúltimo capítulo, da a conocer las pruebas experimentales realizadas, diseñadas en el capítulo anterior y la conclusión de dichos resultados.

Finalmente, en el último capítulo se da a conocer la conclusión sobre el trabajo realizado y el futuro de él.

## Capítulo 2: Tecnología Multiagente

### 2.1 Definición de Agente

El término de agente [1] proviene del latín “agere” que significa hacer; sin embargo, este término es muy confuso, ya que no existe una definición exacta de él, al contrario, existen múltiples definiciones diferentes y siguen apareciendo aún más. Según el diccionario de la RAE [2] significa: “persona o cosa que produce un efecto” o “persona que obra con poder de otra”. A pesar de ello existen criterios claros que permiten distinguir un agente de lo que no es un agente.

Para poder entender esta confusión de significados, hay que determinar mejor de qué es lo que se está hablando, que en este caso son los “agentes software” que están formados por programas de computador ejecutados en un procesador o en una red de procesadores.

Una de las definiciones más citadas es la de Wooldridge [3] que dice: “un agente es un sistema informático situado en un entorno y que es capaz de realizar acciones de forma autónoma para conseguir sus objetivos de diseño”. Realizar acciones de forma autónoma significa que los agentes actúan sin intervención humana ni de otros agentes y tienen el control sobre su estado interno y su comportamiento. Sin embargo, esta definición también es ambigua, y nos indica que cualquier sistema computacional es un agente, pero no son agentes inteligentes.

### 2.2 Agentes Inteligentes

Uno de los significados más comunes es el que definió Bárbara Hayes-Roth [1], que dice: “Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; action to affect conditions in the environment; and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions”; esto es: “ Los agentes inteligentes realizan continuamente 3 funciones: perciben las condiciones dinámicas del ambiente, actúan para modificar las condiciones en el ambiente; y razonan para interpretar percepciones, resolver problemas, sacar conclusiones y determinar acciones”.

Entonces, un agente inteligente es aquel capaz de **situarse** en un entorno (percibir el ambiente en donde se encuentra) y además realizar acciones **autónomas** (actuar sin intervención de otros) y **flexible** para lograr sus objetivos, donde flexible aborda las siguientes características:

- **Reactividad:** Los agentes perciben el entorno en el que están operando, responden y reaccionar de manera oportuna y apropiada a los cambios. Para que un agente actúe adecuadamente debe poder conocer en todo momento el “mundo” que le rodea.

- **Pro actividad (iniciativa):** Tiene que tomar la iniciativa para actuar guiado por los objetivos que tiene que satisfacer. En cada momento el agente decide que acción llevar a cabo. No solo actúa en función de los estímulos que percibe sino que realiza acciones como resultado de sus decisiones.
- **Habilidad Social:** Tienen la capacidad de interactuar con otros agentes, incluso con seres humanos, utilizando alguna clase de lenguaje de comunicación de agentes para alcanzar un objetivo o meta. Los agentes colaboran entre sí para la ejecución de tareas; esto se ve mayormente reflejado en los sistemas multiagente.

Además de estas características que nos permiten distinguir a los agentes de los agentes inteligentes, también pueden tener otros atributos, algunos de los más comunes son [4]:

- **Movilidad:** La capacidad y habilidad de trasladarse de una manera autónoma por las redes.
- **Veracidad:** Se supone que un agente no comunicará con conocimiento información falsa, o sea intencionalmente.
- **Benevolencia:** Los agentes no tienen metas que están en conflicto, con objetivos contradictorios y que por lo tanto cada agente intentará realizar siempre la tarea que se le solicita.
- **Racionalidad:** El agente actuará según los conocimientos del entorno para alcanzar sus metas, y no actuará para evitar el alcance de sus metas.
- **Adaptabilidad:** El agente es capaz de aprender de sus comportamientos anteriores y de mejorar con experiencia.

## 2.3 Comparación de los Agentes

A parte de la diferenciación entre un agente y un agente inteligente, existen confusiones con otros términos dentro de la informática, éstos son objetos y sistemas expertos. Es por ello que se hacen las distinciones correspondientes:

### 2.3.1 Agente v/s Objeto

Si bien es correcto que existen ciertas semejanzas entre agentes y objetos, como que se comunican mediante mensajes y se caracterizan por su estado y comportamiento, hay tres diferencias fundamentales para comprobar que no son lo mismo [5]:

- **Autónomo:** Un objeto recibe invocaciones de métodos, no tienen ningún control sobre sus acciones; un agente recibe peticiones y controla sus acciones para cumplir el objetivo y es él quien determina que puede o no realizar.

- **Flexible:** La noción de tener comportamiento reactivo, proactivo y habilidad social de los agentes que no existen en los objetos.
- **Activo:** Cada agente tiene su propio hilo de control que pueda seguir, en cambio los objetos sólo están activos cuando sus métodos son llamados por otros objetos, el resto del tiempo no.

### 2.3.2 Agente v/s Sistemas Expertos

Los agentes se sitúan en un ambiente, en cambio los sistemas expertos se basa en conocimientos a través de preguntas, no está consciente del entorno y no interacciona directamente con él. Los agentes actúan, los sistemas expertos no.

## 2.4 Tipología de Agentes

Los criterios de clasificación de agentes se pueden establecer desde distintos puntos de vista, teniendo en cuenta las propiedades que posean, sus características individuales, su utilidad, etc. En adelante se presentan algunas clasificaciones[51]:

- **Agentes Reactivos:** Son aquellos que tienen la capacidad de reaccionar a cambios de su ambiente o a mensajes provenientes de otros agentes, pero no son capaces de razonar acerca de sus intenciones (manejo de sus metas u objetivos). Los agentes reactivos realizan tareas sencillas y en donde la reacción consiste en la ejecución de procedimientos según el estado interno del agente.
- **Agentes Intencionales:** Estos agentes son capaces de razonar acerca de sus intenciones y conocimientos, seleccionando metas y siendo capaces de resolverlas, crear planes de acción (programarlos), y ejecutar dichos planes detectando posibles conflictos y también revisarlos.
- **Agentes Sociales:** Son capaces de mantener los modelos de otros agentes mediante la actualización de conocimientos, metas y planes, capaces de razonar sobre el conocimiento de estos modelos, tomar sus decisiones y crear sus planes con respecto a los modelos de los otros agentes.
- **Agentes Colaborativos o Cooperativos:** Son aquellos agentes autónomos pero que cooperan con otros agentes para realizar sus tareas y también pueden aprender de ellos. Estos agentes tienen que negociar para alcanzar compromisos y metas comunes. Ellos son capaces de resolver problemas que son demasiado grandes para sistemas centralizados, dar solución a problemas distribuidos, dar solución a

problemas en los que existen varias fuentes de información y dar solución a problemas en donde la experiencia se encuentra distribuida.

- **Agentes de Interfaz:** Un ejemplo claro de este tipo de agentes es el de un asistente personal que colabora con su usuario en el mismo ambiente de trabajo. Los agentes de interfaz asisten y dan soporte al usuario para aprender el uso de una aplicación (autónomos) y aprenden para mejorar su ayuda al usuario a través de la observación, imitación, recepción de instrucciones y al pedir consejos a otros (colaboración).
- **Agentes Móviles:** Los agentes móviles son programas de software capaces de viajar por redes de computadoras (por ejemplo por Internet), de interactuar con hosts, pedir información a nombre de su usuario y regresar a su lugar de origen una vez que ha realizado las tareas especificadas por su usuario.

## 2.5 Arquitecturas Para Construir Agentes

La estructura interna de un agente está determinada por alguna arquitectura. Una arquitectura permite descomponer un sistema computacional en componentes más pequeños y ver como es la relación entre ellos; en el caso de las arquitecturas de agentes, determina los mecanismos que utiliza un agente para reaccionar, actuar, comunicarse, etc.

A continuación se presenta una revisión de las arquitecturas y modelos más significativos en los que se puede basar el diseño de cada agente, que a modo general se encuentran las arquitecturas [1] deliberativas, reactivas e híbridas.

### 2.5.1 Arquitecturas Deliberativas

Son aquellas arquitecturas que expresan el comportamiento y el ambiente en términos de conocimiento y utilizan modelos de representación simbólica del conocimiento. Están basadas en la teoría clásica de la planificación, donde existe un estado inicial de partida, un conjunto de planes y un objetivo a cumplir. En esta arquitectura el sistema de planificación que contenga un determinado agente, es el encargado de determinar los pasos que se van a ejecutar para lograr el objetivo buscado. Entonces en este caso se dice que un agente deliberativo es aquel que contiene un modelo simbólico del mundo que se representa, en donde las decisiones se toman utilizando mecanismos de razonamiento lógico basados en la concordancia de patrones y la manipulación simbólica (planes). Algunas arquitecturas de este tipo son la arquitectura BDI y la arquitectura abstracta.

### 2.5.2 Arquitecturas Reactivas

Estas arquitecturas no incluyen un modelo del mundo simbólico ni hacen ningún tipo de razonamiento simbólico, sino que estudian sobre modelos más efectivos de representación del conocimiento. Esta arquitectura desarrolla y combina comportamientos

individuales, situados en un entorno y que reaccionan ante ese entorno, en donde la toma de decisiones se basa en el comportamiento que se tenga al momento de cumplir alguna tarea. Puede suceder que los comportamientos ocurran al mismo tiempo, por lo que para elegir uno de ellos, éstos se disponen en capas de manera jerárquica, en donde las capas inferiores tienen más prioridad que las superiores.

La arquitectura de subsunción propuesta por Rodney Brooks [8] es un ejemplo típico de este tipo de arquitectura, la cual se basa en las hipótesis de que la inteligencia es una propiedad emergente en ciertos sistemas complejos y por tanto se puede generar comportamientos inteligentes sin necesidad de construir un modelo simbólico, sino más bien mediante una jerarquía de tareas que define un comportamiento, organizadas en capas.

En robótica es donde se aplica mayoritariamente esta arquitectura, viendo a los robots como agentes reales que actúan en un entorno cambiante e impredecible y reaccionen y respondan con agilidad a los cambios.

### **2.5.3 Arquitecturas Híbridas**

Este sistema pretende combinar aspectos de la arquitectura deliberativa y la reactiva; deliberativa para que utilice un modelo simbólico y que genere planes y reactiva para que se pueda reaccionar en un periodo oportuno de tiempo ante los eventos que tenga el entorno y que no tenga un mecanismo de razonamiento complejo.

Se estructura en dos grandes capas: vertical y horizontal. La vertical permite que solamente una capa tenga acceso a los sensores (entradas) y actuadores (salidas). Por lo contrario, en la horizontal todas las capas tienen acceso, en donde cada una actúa por sí misma como un agente.

Se encuentran tres niveles: reactivo, conocimiento y social. El nivel más bajo es el reactivo, que reacciona y se toman decisiones en base a los estímulos del entorno en tiempo real, y que suele implementarse con una arquitectura de subsunción. El nivel intermedio es el de conocimiento, éste posee información del medio y se centra en el conocimiento que posee el agente, a través de una representación simbólica. Y el nivel más alto es el social, acá se manejan los aspectos sociales del entorno e información de otros agentes.

## **2.6 Arquitectura Multiagente**

Una definición formal es descrita por Weiss [9]: “Un sistema multiagente es una red de problem-solvers que trabajan conjuntamente para encontrar respuestas a problemas que ven más allá de las capacidades o conocimientos individuales de cada entidad”.

Un sistema multiagente [1] es un sistema compuesto de múltiples agentes, capaces de cumplir objetivos que son difíciles de alcanzar por un sistema individual, en donde la inteligencia es distribuida entre ellos para así permitir la gestión inteligente de un sistema

complejo. Cada agente trabaja en común con los otros agentes para resolver el problema, ya que por sí solos son incapaces de resolverlo, además son capaces de tomar la iniciativa, compartir conocimiento, cooperar y negociar y de comprometerse con metas comunes.

Estos sistemas proveen una infraestructura específica de comunicación y protocolos de interacción, son típicamente abiertos y no tienen un diseño centralizado, ya que el control del sistema es netamente descentralizado, y los datos son dispersos.

La coordinación, cooperación y negociación entre los agentes son propiedades claves dentro de un MAS (Sistema Multiagente), ya que mediante estas políticas se puede resolver el problema y conseguir los objetivos predefinidos; todo ello implica que el funcionamiento del sistema está correcto. Por ello se necesita que el sistema multiagente sea primeramente cooperante, donde las características de éste son las siguientes:

- Los agentes deben ser autónomos en cuanto a sus habilidades de adquisición de datos, comunicación, planificación y actuación.
- Se debe perseguir un objetivo común, el cual se puede dividir en varias tareas independientes que se pueden ejecutar en paralelo, ya que cada una de ellas es asignada a distintos componentes.
- Cada agente posee un conocimiento limitado del entorno, del objetivo global y de las intenciones que tengan los demás agentes al ejecutar sus propias tareas.
- Cada agente es especializado en lo que debe hacer, en función de lo que conoce, la capacidad de proceso y la habilidad requerida.

Todas estas características de los MAS cooperantes, hacen que las decisiones sean distribuidas, lo que permite que cada agente siga siendo autónomo, ya que decide sobre las tareas en función de su entorno; que las decisiones no estén en un único agente y que sean consensuadas entre ellos, permitiendo que haya adaptabilidad ante posibles cambios en el entorno en tiempo real y además la definición del mecanismo de comunicación que van a tener.

Dentro del problema de la coordinación entre los agentes son importantes dos procesos, el proceso de razonamiento y el proceso de comunicación. El primero consiste en la toma de decisiones y la identificación de la información a compartir. El segundo analiza el cómo, el qué y el cuándo debe producirse la comunicación entre los agentes.

Dentro de todas las características que poseen los sistemas multiagente, existe una gran variedad de ventajas [10], algunas de las más relevantes que posee son:

- **Velocidad y Eficiencia:** Los agentes pueden funcionar asincrónicamente y en paralelo.

- **Robustez y Confiabilidad:** La existencia de múltiples agentes presenta algunas características de tolerancia a las fallas.
- **Escalabilidad y Flexibilidad:** El sistema puede ser escalado naturalmente agregando nuevos agentes.
- **Costos:** El problema puede ser descompuesto en varios subsistemas simples con costo unitario menor.
- **Desarrollo y Reusabilidad:** Los agentes pueden ser extendidos, reutilizados, probados y mantenidos fácilmente.

Así como se nombraron las ventajas, también existen desventajas [10] que están presentes en estos sistemas, estas son:

- **Overhead:** es preferible elegir otra tecnología si ésta contiene los mismos beneficios que un MAS.
- **Ausencia de un sistema de control central:** Dificultad para representar restricciones globales
- **Ausencia de una perspectiva global:** Los agente tienen conocimiento local del sistema, por lo tanto las decisiones que toman se basan en ello, lo que puede afectar si se ve globalmente.
- **Delegación y Confianza:** Existe mucha delegación de tareas y extrema confianza de parte de las personas hacia los agentes.

## 2.7 Arquitectura FIPA

El desarrollo de los sistemas multiagente hizo que se investigaran nuevas formas para desarrollar esta tecnología. Con ello se presentaron diversos problemas, uno de ellos fue la carencia de una definición estándar de sistemas multiagente y la incompatibilidad de ellos. Esto promovió a la creación de organismos que se dedicaron a desarrollar una definición estándar para un MAS, algunos de ellos son: OMG (Object Management Group), KSE (Knowledge Sharing Effort) y FIPA (Foundation for Intelligent Physical Agents), entre otros; de los cuales FIPA es el que ha tenido más relevancia.

FIPA es el acrónimo de Foundation for Intelligent Physical Agents [11] (fundación para agentes inteligentes físicos) y está dedicado a la estandarización de agentes; esta organización es la que más se ha dedicado a construir estándares para facilitar el uso de tecnologías multiagente, resolviendo para ello los problemas de interoperabilidad (integración de sistemas) y de apertura (posibilidad de extensión).

La serie de principios que FIPA ha seguido para definir la plataforma de agentes son:

- Definir solo el comportamiento externo (interfaz), dejando las decisiones de diseño a los equipos de desarrollo.
- Conseguir un sistema totalmente abierto de tal forma que los sistemas heterogéneos puedan interactuar a nivel de sociedad de agentes.
- Establecer el modelo lógico referente a la creación, destrucción, registro, localización y comunicación de los agentes.

La plataforma de agentes proporciona la infraestructura para el desarrollo y uso de agentes. Esta plataforma contiene todos los recursos tanto de hardware como de software necesarios para poner en marcha esta infraestructura.

El modelo de referencia de FIPA [52] para la gestión de agentes está formado por los siguientes componentes lógicos, presentes a su vez en la Figura 1.1:

- **Agente:** Es el componente básico y principal del modelo, tiene que tener uno o más dueños, disponer de una identidad propia (nombre del agente) y puede registrarse con un número de direcciones en las cuales puede ser contactado.
- **Plataforma de Agentes (Agent Platform, AP):** Proporciona la infraestructura física y lógica y está constituida por el hardware, el sistema operativo, software de comunicaciones y software de agentes.
- **Facilitador de Directorios (Directory Facilitator, DF):** Es un agente que proporciona un servicio de directorio a los demás agentes, los cuales pueden utilizarlo para registrar sus servicios o para buscar los servicios ofrecidos por otros agentes.
- **Sistema de Gestión de Agentes (Agent Management System, AMS):** Es un agente de gestión que controla el estado, el acceso a la plataforma, que conoce en todo momento el estado de su plataforma y de los agentes de ella, y permite la localización de agentes a partir de sus nombres.
- **Agente de Canal de Comunicación (Agent Communication Channel, ACC):** Todos los agentes tienen el canal de comunicación por defecto.
- **Plataforma de Transporte de Mensajes Interna (Internal Platform Message Transport, IPMT):** Método de intercambio de mensajes dentro de la misma plataforma y de otras. Todos los agentes deben tener acceso a él.

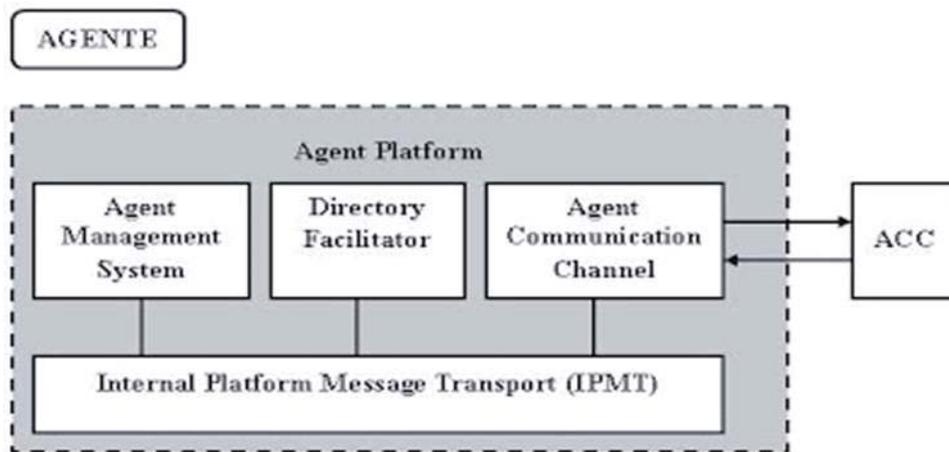


Figura 2.1 Arquitectura FIPA [52]

## 2.8 Infraestructura de Agentes

La comunicación en sí es muy importante para los humanos y también para los agentes, por ello se vio la necesidad de tener una infraestructura para la comunicación clara entre agentes. Esta infraestructura es la que permitirá que los agentes puedan entenderse, comunicarse y relacionarse en un ambiente y dominio de interés común, todo esto a través de las regulaciones que se definan, compuestas de [10]:

- **Ontologías:** Es la que permite que los agentes se entiendan a través del significado común de los conceptos que comuniquen.
- **Lenguajes de Comunicación (ACL's, Agent Communications Lenguaje):** se definen protocolos de comunicación para el intercambio de mensajes entre agentes y gracias a estos protocolos puedan comunicar información y conocimiento.
- **Protocolos de interacción (AIP's):** Un protocolo de interacción es una descripción detallada del tipo y orden de los mensajes involucrados en una conversación entre agentes.

### 2.8.1 Ontologías

La definición que aparece en wikipedia para este concepto es: “El término ontología en informática hace referencia al intento de formular un exhaustivo y riguroso esquema conceptual dentro de un dominio dado, con la finalidad de facilitar la comunicación y la compartición de la información entre diferentes sistemas”. Según esto y aplicado a los agentes, éstos deben tener una ontología común para poder entender los conceptos que se intercambian en los mensajes dentro de un dominio de aplicación, y con ello definir los términos que se van a usar.

Una ontología debe poseer usualmente:

- **Clases:** elementos generales en los dominios interesados.
- **Propiedades:** cualidades de estas clases.
- **Relaciones:** interrelaciones entre las clases involucradas.

En síntesis lo que busca es describir conceptos de manera no ambigua, para ello se fueron creando algunos lenguajes de ontología, donde los más comunes son los que aparecen en la web semántica. La Web Semántica [53] es una web extendida y está contenida de un mayor significado. Se desarrolla con lenguajes universales que permitirán a los usuarios encontrar respuestas a sus preguntas de una forma más rápida y sencilla gracias a la mejor estructuración de la información. Algunos de los lenguajes que proporciona son los que aparecen en la pirámide de la web semántica:

- **XOL:** Ontology eXchange Language, US bioinformatics community.
- **SHOE:** Simple HTML Ontology Extension, university of Maryland.
- **OML:** Ontology Markup Language, university of Washington.
- **DAML:** DARPA Agent Markup Language.
- **OIL:** Ontology Interchange Language, Onto Knowledge Project..
- **DAML + OIL:W3C.** es un el lenguaje estándar de representación de ontologías
- **OWL:** Web Ontology Language, W3C.

Los lenguajes mencionados basan generalmente su sintaxis en el lenguaje estándar XML y hacen que el contenido de la web sea entendible y procesable por agentes de software, servicios y modelos basados en conocimiento.

## 2.8.2 Lenguajes de Comunicación entre Agentes (ACLs)

Los agentes se comunican para entender y ser entendidos y el modelo de comunicación humana es la que imitan los agentes, en donde ellos perciben (reciben el mensaje) y actúan (envían un mensaje).

El estudio formal de la comunicación indica que existen tres aspectos importantes para poder entenderla:

- **Sintaxis:** es el cómo se estructuran los símbolos de la comunicación.
- **Semántica:** es la que indica que cosas los símbolos denotan.
- **Pragmática:** es la que indica cómo se debiesen interpretar dichos símbolos.

La combinación entre la semántica y la pragmática es lo que constituye el significado de lo que se desea comunicar.

Dentro de esto, existe una teoría muy común que se aplica a los agentes, denominada teoría del acto del habla. Esta teoría está desarrollada por lingüistas para ayudar en la comprensión del lenguaje humano, en donde estudia el lenguaje como acción, los hablantes no sólo expresan sentencias ciertas o falsas, informan, preguntan, sugieren, prometen. Por ende es relevante que se puedan distinguir e identificarlos para una correcta comunicación, en la cual se encuentran tres acciones:

- **Locución:** Contexto del hecho físico del acto del habla.
- **Ilocución:** Intención con la que se realiza.
- **Perlocución:** Acto que ocurre como resultado, el efecto causado en el receptor.

Esta teoría es la base en la comunicación entre agentes, y se utilizan los performativos para ver la fuerza ilocutoria del significado de la oración, esto para que no sufra de ambigüedad la ilocución de la oración. Todo lo anterior se utiliza para saber si el contenido del mensaje es una información, una respuesta, una consulta, etc.

Existen estructuras definidas para el envío de mensajes entre distintos agentes, dentro de los cuales se encuentra KQML y FIPA ACL.

### **2.8.2.1 KQM (Knowledge Query Manipulation Language)**

Es un lenguaje [1] y protocolo de un formato para los mensajes basados en paréntesis (similar a lisp), apoyando a los agentes en la identificación y conexión con los otros agentes en el intercambio de información (comunicación). Además posee mecanismos de traducción semiautomática dentro de los lenguajes de representación del conocimiento (como por ejemplo prolog), esta ventaja se debe a que está asociado al lenguaje de representación KIF (KnowledgeInterchangeFormat). Está dividido en tres capas: capa de mensaje, de contenido y de comunicación. En la tabla 2.1 que se muestra a continuación, se detallan las estructuras de los mensajes en KQML:

**Tabla 2.1 Elementos de un mensaje reservado de KQML [1].**

Elemento del mensaje	Descripción
:sender	Nombre del emisor del mensaje
:reciever	El receptor del mensaje
:in-reply-to	La etiqueta para la respuesta a un mensaje anterior
:reply-with	Si el emisor espera respuesta, la etiqueta para dicha respuesta
:language	Nombre del lenguaje de representación del parámetro :content
:ontology	Nombre de la ontología usada en el parámetro :content
:content	Información sobre la que el performative expresa una actitud

**Tabla 2.2 Elementos de un mensaje reservado FIPA ACL [1].**

Elemento del mensaje	Categoría de elementos
Performative	Tipo de acto comunicativo
Sender	Participante en la comunicación
Reciever	Participante en la comunicación
Reply-to	Participante en la comunicación
Content	Contenido del mensaje
Language	Descripción del contenido
Encoding	Descripción del contenido
Ontology	Descripción del contenido
Protocol	Control de la conversación
Conversation-id	Control de la conversación
Reply-with	Control de la conversación
in-reply-to	Control de la conversación
Reply-by	Control de la conversación

### 2.8.2.2 FIPA ACL

Es un lenguaje para el intercambio de mensajes [1] y se basa en la teoría del acto del habla (performative). Tiene una sintaxis similar a un mensaje en KQML y define la estructura que deben tener los mensajes que se envíen entre los distintos agentes. Los mensajes de este estándar contienen elementos del mensaje, en donde el elemento performative es obligatorio ocuparlo, ya que denota el tipo de acto comunicativo del mensaje, los demás elementos se pueden utilizar como opción. En la tabla 2.2 se exhiben los elementos de los mensajes en FIPA ACL.

### 2.8.3 Protocolos de Interacción

Un protocolo de interacción o de comunicación, describe las restricciones que debe haber dentro del contenido de un mensaje y cuáles de estos mensajes están permitidos entre los agentes. Lo que busca principalmente es representar los patrones que modelan las posibles comunicaciones en donde los participantes deben conocer el protocolo.

Dentro de la especificación de FIPA, están definidos ciertos protocolos en el que se encuentra el FIPA IPL, Interaction Protocol Library, que está abierta a que cada diseñador añada sus propios protocolos. Para que se definan estos protocolos FIPA utiliza una notación llamada AUML, la cual está basada en los diagramas de UML, que dan origen a los diagramas de protocolos. En ellos se pueden encontrar los siguientes:

- **Roles de los agentes:** identifican el papel de cada agente dentro de la comunicación. Su notación corresponde a un rectángulo con el nombre del rol o roles.
- **Línea de vida:** define el tiempo de participación de un agente en la comunicación.
- **Hilos de interacción:** muestra el período de tiempo en el que un agente realiza una tarea como reacción a un mensaje entrante. Se representa como un rectángulo alargado dibujado sobre la línea de vida.
- **Mensajes:** Es la acción de comunicación entre un agente y otro, representada por una flecha horizontal etiquetada con el nombre del acto comunicativo.

Los distintos protocolos de comunicación del FIPA y su descripción se muestran a continuación en la tabla 2.3:

**Tabla 2.3 Protocolos de comunicación estándar FIPA [1].**

Nombre protocolo	Descripción
Request	A un agente se le pide que realice cierta acción.
Requestwhen	A un agente se le pide que realice cierta acción siempre que se cumpla la precondición
Query	A un agente se le pide que informe sobre algo.
Contract Net	Un agente pide la realización de cierta tarea a un conjunto de agentes. Estos dan su propuesta basada en costes y el iniciador elige quién la realiza finalmente.
Brokering	Un agente (broker) ofrece las funcionalidades de otros agentes o reenvía las peticiones al agente apropiado
English auction	Varios agentes participan en una subasta que se inicia con un precio más bajo y progresivamente se va subiendo.
Dutch auction	Varios agentes participan en una subasta que se inicia con un precio más alto y progresivamente se va bajando.
Recruiting	Es como el brokering, pero las respuestas sobre el servicio van directamente al agente que lo necesita (no a través del broker).
Propose	El iniciador propone a una serie de agentes la realización de una tarea y estos aceptan o no.
Subscribe	Un agente pide ser notificado si cierta condición se vuelve verdadera.

## 2.9 Metodologías de Desarrollo

Las metodologías se pueden describir como medios para construir sistemas multiagente, aquí se explicaran algunas de las metodologías más conocidas, pero la que se utilizará en este proyecto será la metodología PASSI.

### 2.9.1 AAI/BDI

AAI (Australian Artificial Intelligence Institute) [1], esta es basada en la arquitectura BDI, creencias, deseos e intenciones, una creencia es información sobre el entorno, un deseo es algo que quiere conseguir y las intenciones son los objetivos elegidos. Los deseos

corresponden a las metas del agente, estas pueden ser adoptadas por él, o elegidas por el usuario. Para lograr sus metas debe usar un plan estático, dinámico o uno que convine a ambos. Esta metodología, ha influido mucho en la forma de concebir el control en los agentes.

### **2.9.2 MAS-CommonKADS**

Esta metodología extiende de CommonKADS [13], esta no fue diseñada para su utilización en sistemas multiagente, ya que gira en torno de modelos de experiencia y está pensado para desarrollar sistemas expertos centralizados que interactúan con usuarios, estos no tienen ninguna de las propiedades de los agentes, para lograr extender esta metodología para ser aplicable a los modelos multiagente, se define una serie de modelos.

### **2.9.3 GAIA**

Basado en agentes en una sociedad u organización [12], en que cada agente tiene un rol, este interactúan con otros roles, de esta manera toman parte en patrones de interacción. Cada rol es definido por 4 atributos, estos son: Responsabilidades del agente, Permisos, Actividades y Protocolos.

Su objetivo es obtener un sistema que maximice alguna medida de calidad global [12], para esto no se preocupa de definir modelos computacionales, sino más bien, describir como coopera la sociedad, propone trabajar inicialmente con un análisis de alto nivel, para esto utiliza el siguiente análisis: identifica los roles del sistema, para cada rol, identifica y documento los protocolos asociados y elabora el modelo de roles con la información anterior. Luego de esto se producen 3 modelos que son: el modelo de agente, el modelo de servicios y el modelo de conocidos.

### **2.9.4MaSE**

Sigue un paradigma orientado a objetos, los agentes son especializaciones de objetos [12], esta visión de agente consiste en que los agentes se coordinan vía conversaciones para actuar proactivamente y lograr sus metas. Estos agentes pueden ser o no inteligentes, pero pueden coexistir ambos en el mismo sistema.

El análisis de MaSE consta de tres pasos: captura de objetivos, los cuales estructura en una jerarquía de objetivos, luego un análisis de casos de uso, a partir de esto es posible definir y refinar los roles. Debido al análisis se crean diagramas de objetivos, diagramas de roles, tareas asociadas a los roles y casos de uso considerando la posibilidad de utilizar diagramas de secuencia para detallarlos.

### **2.9.5 PASSI**

Process for Agents Societies Specification and Implementation [14] es una metodología en la que el agente tiene uno o más roles, este siendo una función temporal asumida por el agente en la sociedad, a menudo se relaciona con un servicio ofrecido por el agente a esta. El agente utiliza tareas para ejecutar sus planes, estas son entidades que apuntan a alcanzar una sub-meta. Esta metodología integra modelos de diseño y conceptos.

En su etapa de análisis se encuentran las siguientes fases: descripción del dominio, esta utiliza diagramas de casos de uso; identificación de agentes, esta se puede realizar a partir del diagrama anterior; identificación de roles, esta es representada mediante diagramas de secuencias e identificación de tareas.

En la etapa de diseño, se encuentran las fases [14] de: descripción de ontologías; descripción de roles, descripción de comunicación, se modela la vida de los agentes a partir de sus roles; descripción de protocolos; definición de estructura y del comportamiento de los agentes.

## Capítulo 3: Berth Allocation Problem

### 3.1 Introducción al Problema

Desde tiempos remotos el transporte marítimo ha sido el método con más relevancia, tanto para transporte de personas y mercancías, medio de transporte para ampliar territorios, la guerra, y otros. Este medio impulso muchos de los descubrimientos geográficos hoy muy conocidos, todo esto con apoyo de los avances científicos de esta época. Con los avances actuales de la tecnología esto se ha hecho mucho más fácil, pero no por esto menos barato.

Existen entre 6000 a 7000 puertos alrededor del mundo, el diccionario de la RAE [2] los define como “Lugar en la costa o en las orillas de un río que por sus características, naturales o artificiales, sirve para que las embarcaciones realicen operaciones de carga y descarga, embarque y desembarco, etc.”.

Muchos de estos puertos han significado el desarrollo de varios países al ser éstos principalmente marítimos.

Las tendencias actuales en el transporte marítimo son: mayor importancia a los contenedores para almacenar mercancías, alianzas y fusiones entre empresas navieras, un aumento en el tamaño de las naves portacontenedores, con esto un aumento de las profundidades y eficiencia de los puertos, la concentración de carga en pocos puertos a nivel mundial, entre otros.

Chile cuenta con 57 puertos a lo largo de su costa, de los cuales 3 de los más importantes en el transporte de contenedores son [56]: primero el puerto de Valparaíso, seguido por el puerto de San Antonio y finalmente el puerto de Talcahuano, los cuales también están en el ranking de los principales puertos de América Latina con grandes cantidades de TEU (Twenty-Foot Equivalent Unit, unidad de medida de capacidad del transporte marítimo en contenedores) transportadas. El puerto de Valparaíso, asociado al sector económico, está localizado a 110 Km al noreste de la capital chilena, está protegido de vientos por una amplia bahía destinada al arribo de barcos.

Con la aparición de los contenedores en el siglo XX [15] el transporte marítimo se vuelve de gran importancia, ya que esto requería menor embalaje del producto, estableció un estándar internacional, se redujeron los daños, etc. Se produce un crecimiento del comercio mundial debido a los avances tecnológicos, acuerdos comerciales, y otros; involucrando un aumento del flujo de contenedores, en la importación y exportación de mercancías y un aumento en los tamaños de los barcos portacontenedores. En la actualidad se estudia la posibilidad de ampliar las capacidades de transbordo existentes de los puertos, lo que significaría construir nuevos terminales o ampliar los ya existentes, además de adquirir maquinaria nueva; sin embargo no siempre estos proyectos de expansión de los

puertos son factibles, ya sea por cuestiones socioeconómicas, gubernamentales y/o ambientales.

Al momento que un barco portacontenedores avisa su arribo a puerto se plantean 3 incógnitas de planificación, las que deben ser resueltas antes de que este portacontenedores arribe; estas son la asignación de este portacontenedores a algún sector del muelle, la asignación de grúas y la disposición de sus contenedores en el patio. Al arribo de los barcos a puerto [15] estos deben atracar en un espacio asignado para ello, que este equipado con las grúas apropiadas para cargar y/o descargar sus contenedores, luego los vehículos correspondientes los transportan al patio y otros los apilan para luego ser llevados a su destino final. La maquinaria utilizada para este propósito son diversas grúas como las Gantry, grúas de patio y camiones como el reachstacker, entre otros.

Con la finalidad de cumplir con estos y otras funciones que tiene el puerto este es dividido en distintas zonas, cuatro de las principales son:

- **Zona de muelle**, la cual sirve para el amarre de los barcos y se emplean las grúas para carga y descarga de contenedores.
- **Área para el transporte de contenedores** dentro del puerto mediante camiones y vehículos guiados automáticamente.
- **Área para el depósito de contenedores** a través de vehículos utilizados para la apelación de los contenedores dentro del patio.
- **Área para camiones** que sirven de traslado de los contenedores entre el puerto y las zonas exteriores.

Cuando se habla de los problemas asociados a puerto se pueden tomar decisiones de tres tipos [16]: estratégicas, tácticas y operacionales, y con ello analizar cómo debe llevarse a cabo la gestión de recursos que se tienen en el terminal portuario. Las decisiones de tipo estratégicas son aquellas que ven el tipo de equipamientos que se usará, la cantidad de ellos y cómo van a operar, estas decisiones pueden durar de uno a varios años y se toman teniendo como base las experiencias. Las decisiones tácticas se refieren a procesos de asignación de recursos, a qué estrategia seguir para almacenar los contenedores, el uso del espacio del muelle y del patio, éstas suelen durar algunos días y hasta meses. Por último las decisiones de tipo operativa son aquellas donde se realizan los planes de trabajo para la asignación de recursos, como dónde se va a colocar cada contenedor para su posterior carga o descarga; estas decisiones se deben tomar día a día. Los tres tipos de decisiones son influenciadas unas por otras; es por ello que la información sobre las características de un terminal portuario, el equipamiento y maquinarias que utilizan y además los conocimientos y usos que se tienen referentes a metodologías de problemas de asignación, planificación,

optimización y de manera general la investigación de operaciones, hace que se puedan tomar decisiones y resulten de ellas soluciones apropiadas.

Todo lo antes mencionado conlleva costos y tiempo, como las operaciones de carga y descarga, los tiempos de espera que se producen, y algunas otras variables, para optimizar todos estos tiempos, y a su vez costos, existe la necesidad de tener sistemas de ayuda a la decisión y procedimientos que encuentren soluciones apropiadas para los problemas que deben resolverse diariamente. Para esto hay una vasta cantidad de problemas de optimización, como lo son el *Quay Crane Assignment Problem*, el *Yard Allocation Problem*, entre otros, todos estos para apoyar la toma de decisiones, cada uno de estos apoyando a distintas áreas y tareas realizadas en el puerto, como los son la descarga de barcos, asignación de espacio de almacenamiento de contenedores, determinación de rutas para los vehículos que transportan los contenedores dentro y fuera del puerto, etc. El área que interesa a este informe es el área de muelle y el problema de optimización para esta área es el de asignación de barcos a muelles, este es llamado *Berth Allocation Problem* (BAP).

En este problema se tiene en cuenta un conjunto de atraques y uno de barcos que han de ser atendidos dentro del horizonte de planificación, a los que hay que asignarles una localización en el muelle y un tiempo de servicio, de manera de optimizar alguna función objetivo, por esto se deben considerar ciertas características, como lo son su tamaño, cantidad de contenedores que se deben descargar, los que se deben cargar, entre otros. Con esto el tiempo que debe permanecer un barco en el puerto debe ser mínimo, al igual que el uso de maquinarias, esto significa que una gran cantidad de contenedores debe ser cargada, descargada y transbordada en un breve espacio de tiempo.

El BAP es considerado uno de los principales problemas de optimización, este se ve como una descripción de dos dimensiones, una de estas es espacial, el largo del muelle, mientras la otra es un horizonte de decisión temporal [20]. Los barcos pueden ser representados por rectángulos los cuales tienen como dimensiones el largo y tiempo de manejo, este último sería el tiempo que el barco está en el muelle, mientras el tiempo de servicio es el tiempo total que resulta al barco no ser atendido inmediatamente. Estos rectángulos deben ser posicionados en el espacio de decisión sin ser uno puesto encima del otro, para así no alterar el largo del muelle ni el horizonte de decisión. A continuación se presenta un modelo de la representación mencionada anteriormente (Figura 3.1).

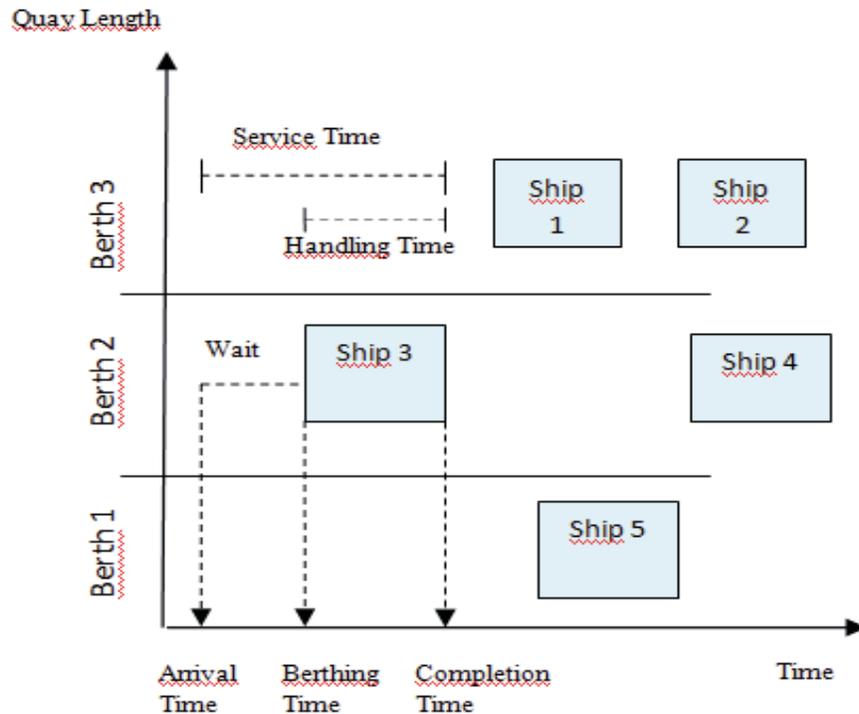


Figura 3.1 Representación espacio tiempo [17]

### 3.2 Formulación Matemática

Como se ha mencionado anteriormente, en este informe se tratara la versión discreta del BAP. Se encuentra una asociación de BAP con una variante en el problema de Multi-Depot Vehicle Routing Problem With Time Windows (MDVRPTW) [17]. En este los barcos son vistos como clientes, y los puntos de atraque como depósitos de vehículos, donde cada uno está ubicado. Existen  $M$  vehículos, uno para cada depósito. También se debe considerar que cada vehículo parte y termina en el mismo depósito. Los barcos son considerados vértices de un multigrafo, mientras que los depósitos, como divisiones en vértices de origen y destino y ventanas de tiempo pueden ser consideradas en cada vértice, es decir, estas pueden estar tanto en el origen como en el destino, siendo estas consideradas a periodos disponibles de determinado punto de atraque.

Las variables son:

$a_i$ : límite inferior de la ventana de tiempo de servicio del barco  $i$ .

$b_i$  : límite superior de la ventana de tiempo de servicio del barco  $i$ .

$v_i$ : el valor, o costo, del tiempo de servicio para el barco  $i$

Como recién se mencionó, el problema es modelado como un multigrafo  $G_k = (V_k, A_k)$ ,  $\forall k \in M$ , donde  $V_k = N \cup \{o(k), d(k)\}$  y  $A_k \in V_k \times V_k$ .

Las siguientes variables y constantes son definidas:

- $X_{ijk} \in \{0,1\}$   $k \in M$  (i,j)  $A_k$ ,  $X_{ijk} = 1$ , sí y solo sí el barco j está programado después que el barco i.
- $T_{ik} \in M$ ,  $i \in N$ : el tiempo de atraque del barco i en el atraque k, o sea, el tiempo que se demorara el barco i en amarrarse al atraque k
- $T_{o(k)-k}$   $k \in M$ : el tiempo de comienzo de operaciones del punto de atraque k, dado por el primer barco que atraca en ese punto.
- $T_{d(k)-k}$ ,  $k \in M$ : el tiempo de cese de operaciones del punto de atraque k, dado por el tiempo de salida del último barco en ese punto.
- $M_{ijk} = \max \{b_i + t_{ik} - a_j, 0\}$ ,  $k \in M$ , i y j  $\in N$ .

Este problema se modela de la siguiente manera:

Minimizar

$$\sum_{i \in N} \sum_{k \in M} v_i \left[ T_i^k - a_i + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k \right] \quad (1)$$

Sujeto a:

$$\sum_{k \in M} \sum_{j \in N \cup \{d(k)\}} x_{ij}^k = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{o(k),j}^k = 1 \quad \forall k \in M \quad (3)$$

$$\sum_{i \in N \cup \{o(k)\}} x_{i,d(k)}^k = 1 \quad \forall k \in M \quad (4)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{ij}^k - \sum_{i \in N \cup \{o(k)\}} x_{ji}^k = 0 \quad \forall k \in M, \forall i \in N \quad (5)$$

$$T_i^k + t_i^k - T_j^k \leq (1 - x_{ij}^k) M_{ij}^k \quad \forall k \in M, \forall (i, j) \in A^k \quad (6)$$

$$a_i \leq T_i^k \quad \forall k \in M, \forall i \in N \quad (7)$$

$$T_i^k + t_i^k - \sum_{j \in N \cup \{d(k)\}} x_{ij}^k \leq b_i \quad \forall k \in M, \forall i \in N \quad (8)$$

$$s^k \leq T_{o(k)}^k \quad \forall k \in M \quad (9)$$

$$T_{d(k)}^k \leq e^k \quad \forall k \in M \quad (10)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall k \in M, \forall (i, j) \in A^k. \quad (11)$$

La función objetivo es la minimización de la suma de los tiempos de servicio con un cierto *peso o valor* asignado. Cuando la nave  $i$  no está asignada al punto de atraque  $k$ , el término correspondiente en la función objetivo es 0 puesto que  $T_{ik} = a_i$ , por lo tanto, el objetivo es minimizado. La restricción (2) indica que para cada barco  $i$  existe exactamente un arco activo  $(i; j) \in A^k$ ,  $\forall k \in M$ . Las restricciones (3) y (4) definen el *grado* de los depósitos, mientras que la conservación del flujo para los restantes vértices está asegurada por la restricción (5). La consistencia de las  $T^k$  variables con la secuencia en el punto de atraque se logra mediante la restricción (6). Las ventanas de tiempo de servicio en los barcos son fijadas por las restricciones (7) y (8), y las ventanas de tiempo de la disponibilidad en los puntos de atraque por (9) y (10).

### 3.3 Análisis de la Literatura

El BAP ha sido estudiado ampliamente dentro de la literatura. Para el BAP se han definido una gran cantidad de modelos matemáticos, divididos en los modelos destinados para el BAP [18] discreto, para el modelo continuo y el modelo híbrido, que dependen básicamente de la estructura espacial del muelle.

- **Modelo Discreto:** En el problema discreto, el muelle se divide en un número finito de secciones o espacios, con una longitud de espacio más o menos uniforme y donde cada espacio está destinado para recibir un solo barco a la vez, véase las Figura 3.2 a) y b) que corresponden a este diseño. Para este caso, el BAP puede ser modelado como un problema de scheduling en máquinas paralelas no relacionadas (Unrelated Parallel Machine Scheduling), donde cada barco es visto como un trabajo y cada punto de atraque como una máquina.

- **Modelo Continuo:** En el caso continuo, el muelle no está dividido en secciones y cada barco puede atracar en cualquier lugar arbitrariamente a lo largo del muelle, dependiendo de la posición de los demás barcos que estén amarrados, véase la Figura 3.2 c). El hecho de considerar el muelle continuo lo hace mucho más flexible al dar una mejor utilización del espacio del muelle, pero a la vez mucho más complejo de abordarlo computacionalmente. En este caso, el BAP se puede considerar como un problema bidimensional del Cutting Stock Problem, pero se deben agregar las restricciones correspondientes al caso.
- **Modelo Híbrido:** Además de los dos casos más comunes nombrados anteriormente, existe el modelo híbrido (Bierwirth y Meisel, 2010). Acá el muelle se puede dividir en un conjunto de secciones, en donde los barcos más grandes que el espacio asignado pueden ocupar más de una sección a la vez, indicado en la Figura 3.2 d), mientras que los más pequeños pueden compartir su espacio con otro barco en un mismo tiempo, demostrada en la Figura 3.2 e); y cuando el barco es demasiado grande, puede ocupar el espacio que necesite, como se muestra en la Figura 3.2 f).

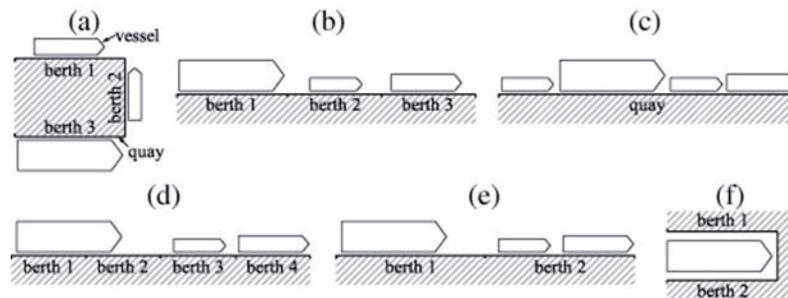


Figura 3.1 Modelo espacial del muelle [18].

La elección del modelo depende de muchos factores, algunos de los más relevantes son el tipo de puerto, tamaño del puerto, el volumen de tráfico a transportar, y tipo de mercancía. Cualquiera de éstos, se considera como un problema con dificultad NP- Hard.

A la vez, existen restricciones temporales para medir el tiempo de atraque del barco y para los horarios de salida del muelle. Para ello se distinguen dos modelos: el BAP estático y el BAP dinámico.

- **Modelo Estático:** En el Static Berth Allocation Problem (SBAP), no hay tiempos de llegada dado para los buques o los tiempos de atraque no tienen restricción. En el primer caso se supone que los barcos ya están esperando en el puerto, por lo tanto se permite el atraque de inmediato. En el último caso, se supone que un barco puede apresurarse para cumplir con su tiempo de atraque a la hora prevista de llegada.

- **Modelo Dinámico:** El Dynamic Berth Allocation Problem (DBAP) no considera la dimensión espacial de los barcos ni los puntos de atraque, ya que se representa el muelle como un sistema finito de puntos de atraque y la llegada de los barcos se considera en forma aleatoria. Aquí se considera el tiempo, los horarios de llegada que se les da a los barcos son fijos, por lo tanto, no se les permite que atraquen antes de la hora prevista de llegada. Esto permite que se mantengan los horarios previstos. En algunos casos, se considera el tiempo de espera máxima más el tiempo de manipulación para determinar el horario de salida. En este modelo, todo el servicio del barco debe ser ejecutado dentro de una ventana de tiempo.

Muchos autores han realizado variadas adaptaciones del BAP para dar una solución al problema, proponiendo diferentes métodos. Algunas de las propuestas que se han analizado como métodos de solución para el BAP son [19]:

- Propuestas de algoritmos genéticos para el DBAP.
- Resolución mediante el método Húngaro para el SBAP, al verlo como un problema de asignación bidimensional.
- Consideración del BAP y el QCSP como un único problema de asignación, ya que el tiempo que un barco se encuentre en el muelle dependerá directamente del número de grúas que se encuentren disponibles para realizar las tareas de carga y descarga.
- Propuesta de una heurística basada en la relajación Lagrangiana para el BAP.
- Algoritmos de Tabú Search para el SBAP y con adaptaciones para el DBAP.

Dentro de los métodos propuestos para solucionar la problemática del BAP, se toma en cuenta si dicha solución quiere minimizar la función de tiempo, la función de costo o de ambas a la vez (minimizar tiempo y costo) [15].

A continuación se nombran algunos de los autores que han abordado la problemática del BAP, junto con el método de solución que han propuesto y que utilizaron en sus investigaciones, ellos son [19] [15]:

- **Lim, 1998 [31]:** Este autor considera el BAP continuo. Lim supone que se debe dividir el muelle en secciones de longitud predeterminada y en cada uno de ellos pueden arribar uno o más barcos según su tamaño y supone que los barcos llegan a momentos diferentes al puerto. Postula además que cuando dos barcos estén amarrados en espacios vecinos, debe existir una distancia de seguridad determinada por sus tamaños. Para ello lo representó gráficamente en un plano cartesiano, en donde el eje horizontal corresponde a la dimensión de tiempo y el eje vertical a la longitud barcos; también se puede representar mediante rectángulos, en donde la base

corresponde al tiempo y la altura representa la longitud. Para solucionar su modelo, el autor propone un algoritmo heurístico.

- **Imai et al, 2001 [32]:** Para el modelo de asignación estática, lo resuelve en tiempo polinomial y para el caso dinámico, propone la solución de búsqueda heurística basado en Lagrange.
- **Legato y Mazza, 2001 [33]:** Analizaron el BAP por medio de una simulación, describiendo el proceso de llegada, atraque y salida y propusieron un modelo de red de colas.
- **Nishimura et al, 2001 [34]:** Resume lo postulado por Lim 1998 e Imai et al, 2001, considerando el BAP dinámico. El algoritmo utilizado para la solución corresponde a una meta heurística.
- **Imai et al, 2003 [35]:** Consideró el BAP discreto con un modelo similar al anterior realizado por el autor, en donde lo solucionó a través de un algoritmo genético.
- **Kim y Moon, 2003 [36]:** Los autores consideran el BAP continuo y dinámico tratado como un problema bidimensional y el algoritmo de solución se basa en meta heurística de búsqueda temporal.
- **Park y Kim, 2003 [37]:** Es el primer intento en que quieren juntar el BAP con el problema de programación de las grúas de muelle, al darse cuenta que están estrechamente relacionados entre si. Para la solución se Utiliza el problema de Lagrange y el Lagrange doble.
- **Guan y Cheung, 2004 [38]:** Los autores consideran en caso dinámico y continuo del BAP. Utilizan para la solución el método de Lagrange y un algoritmo de branch and bound.
- **Cordeau et al, 2005 [39]:** Se considera el BAP discreto y dinámico, en donde utilizan ventanas de tiempo para el planteamiento del problema, modelando el problema como un MDVRPTW. Se resuelve basado en la técnica resolutive búsqueda tabú, que es un algoritmo heurístico de búsqueda y demuestran que es muy eficaz. También el algoritmo diseñado sirve para el caso continuo con una pequeña modificación.
- **Imai et al, 2005 [40]:** Se considera el BAP en sus modelos dinámico y continuo, vistos como un problema bidimensional. El algoritmo de resolución es heurístico.
- **Hansen et al, 2006 [41]:** Se considera el BAP discreto y dinámico en donde se componen de tres términos: el tiempo de flujo (espera más servicio), tiempo de servicio de carga y tiempo de avance (precocidad) o retraso (delay). El algoritmo propuesto por los autores es una búsqueda heurística.
- **Moorthy y Teo, 2006 [42]:** Presentan el análisis del problema desde un punto de vista táctico, en el que se consideran dos factores: el tiempo de espera de los barcos y los costos operativos relacionados al manejo de los contenedores. Para la solución propusieron un algoritmo genético.

- **Imai et al 2007, parte A [43]:** Se considera en particular el puerto de Ceres-Paragon, Amsterdam. Se propone un algoritmo genético como solución.
- **Monaco y Sammarra, 2007 [44]:** Los autores ven el BAP como un problema de programación dinámica, con el objetivo de minimizar el tiempo total de respuesta. Utiliza el problema de Lagrange para la solución.
- **Lokuge y Alahakoon, 2007 [45]:** Ellos propusieron como solución para el BAP realizar un sistema multiagente para minimizar tiempo, todo esto mediante un algoritmo matemático.
- **Imai et al 2007, parte B [46]:** Este trabajo presenta un primer intento en integrar la asignación de muelles y la asignación de grúas. Los autores proponen un modelo mixto de programación lineal para la minimización del tiempo de respuesta de los barcos. Se utiliza un algoritmo genético.
- **Imai et al, 2008 [47]:** Se aborda el problema de BAP en el puerto de Colombo. El problema se analiza y se formula en los modelos estáticos y dinámicos utilizando programación lineal entera y mixta. La resolución se hace mediante un algoritmo genético más una heurística.
- **Meisel y Bierwirth, 2009 [48]:** analizaron el modelo híbrido del BAP para minimizar costos, para ello buscaron una solución basada en meta heurística.

### 3.4 Datos de Pruebas

Dentro de la literatura y de los modelos que se han utilizado para la resolución del BAP se necesitan datos de entrada para la función objetivo y comprobar que la salida sea una solución factible.

Para ello los autores de distintas fuentes han utilizado datos simulados, o sea, ficticios, para probar su propuesta, sin embargo estos datos se relacionan directamente con la realidad. Otros en cambio ocupan datos benchmark [18] para instanciar y también existen autores como [48], que utilizan datos reales para dar solución al BAP, por ejemplo dentro del puerto de Amsterdam, y luego en el 2008 utilizó datos del puerto de Colombo.

### 3.5 Algoritmos de Solución

Como se vio anteriormente en el estudio de la literatura, existen una gran cantidad de métodos de resolución para el BAP, en los cuales han incursionado varios autores. Estos métodos pueden ser en exactos (garantizan encontrar la solución óptima) o heurísticos (busca buenas soluciones a un coste computacional razonable), y están basados fundamentalmente en la optimización de una función objetivo; sin embargo, debido a la complejidad de algunos de estos problemas de optimización, se desarrollaron algoritmos metaheurísticos (basadas en heurísticas pero se pueden adaptar a un problema en particular) para poder resolverlos. A continuación se presentan algunas de las técnicas más comunes utilizadas para solucionar el BAP:

### **3.5.1 Mixed Integer Linear Programming**

Cuando todas las variables de decisión son números enteros se habla de programación lineal entera, pero cuando algunas de estas son enteros y las demás reales, se habla de programación lineal entera y mixta [36]. En programación matemática es muy utilizado para llegar a la solución factible, sin embargo en este contexto no es la más apropiada, ya que los tiempos de ejecución computacional son muy elevados, por lo que no es aconsejable para problemas de tiempo real. Éste corresponde a un método de solución exacto.

### **3.5.2 Branch and Bound**

Ésta consiste en crear un árbol dinámicamente con N niveles (éstos corresponden al número de barcos) y cada una de sus ramas corresponde a las posibles soluciones, cuando una de estas ramas no sea la que lleve a la mejor solución, se deja de construir el árbol por esa rama, de esta manera llegará a la solución óptima sin necesidad de recorrer todas las ramas con las posibles soluciones, llegando así más rápidamente a la mejor solución. Éste corresponde a un método de solución exacto.

### **3.5.3 Genetic Algorithms**

Los algoritmos genéticos [58] se utilizan para resolver problemas de búsqueda y optimización. Está basado en el proceso genético de los seres vivos e imitan el proceso de evolución y supervivencia de las poblaciones. A través de esa imitación, van creando sucesivas generaciones de individuos que representan las posibles soluciones al problema; dichos individuos compiten y se cruzan con parejas seleccionadas por la factibilidad de la solución que brindan, de esta manera se llega a la probabilidad de obtener el mejor valor de la función objetivo tras las generaciones creadas. Los algoritmos genéticos no garantizan que se encuentre la solución óptima al problema, pero sí que esta solución sea aceptable.

### **3.5.4 Simulated Annealing**

Sistema de generación eléctrica con pila de combustible de óxido sólido alimentado con residuos forestales y su optimización mediante algoritmos basados en nubes de partículas

Traducido como algoritmo recocido (o enfriamiento) simulado [56]. Este término proviene de una técnica ocupada en metalurgia y consiste básicamente en simular el proceso de recocido de los metales y el cristal. El objetivo de este algoritmo es encontrar una aproximación al óptimo en un espacio de búsqueda muy grande. Para lograr aquello, se itera el procedimiento para encontrar una solución y luego se va comparando y sustituyendo con otras soluciones encontradas hasta hallar la mejor.

### **3.5.5 Método de Relajación Lagrangeana**

Según [26] “Las metaheurística de relajación se refieren a procedimientos de resolución de problemas que utilizan relajaciones del modelo original [...] cuya solución facilitara la solución del problema original.

Desde su creación en el año 1979 por LaGrange, este método ha sido la técnica de acotación preferida. Consiste en relajar un problema de optimización, mediante la búsqueda de la remoción de la restricción “mala” y su colocación en la función objetivo para la resolución más fácil del problema, a esta restricción se le asigna un peso y se convierte en una penalización con el multiplicador de Lagrange. Se usa principalmente en problemas de optimización discreta, como lo es el BAP, y en problemas de redes.

### **3.5.6 Heurística Constructiva**

Esta metaheurística se orienta a los procedimientos que tratan la obtención de una solución a partir del análisis y selección paulatina de las componentes que forman. Éstas adoptan soluciones del problema por medio de un procedimiento que incorpora iterativamente elementos en una estructura vacía, que representa la solución. Establecen estrategias para la selección de los componentes con las que se construye una buena solución del problema.

#### **3.5.6.1 Algoritmo Greedy**

Este algoritmo es una heurística constructiva. Según [25], éstas son utilizados generalmente para resolver problemas de optimización, toman decisiones basados en función a la información que está disponible en cada momento, y una vez que la decisión ya fue tomada esta no vuelve a plantearse en el futuro. No siempre garantizan una solución óptima, por lo tanto siempre se debe plantear el estudiar la corrección del algoritmo para demostrar si las soluciones obtenidas son óptimas.

### **3.5.7 TabuSearch (Búsqueda Tabú)**

Es una metaheurística [55] que busca una solución global del óptimo del problema. Posee memoria adaptativa y se concentra en explorar soluciones que tengan las características buscadas. Consiste en un proceso iterativo, donde se empieza con una solución inicial al azar y se va cambiando a medida que se va encontrando una solución mayormente aceptada, este proceso finaliza cuando se ha iterado un número determinado de veces y no se mejora la solución, o cuando la solución encontrada está cercana al óptimo buscado.

En lo que respecta a la heurística descrita, tiene algunas variantes definidas para el problema que se esté abordando. En el caso del BAP, y específicamente el definido como MDVPRTW, se han desarrollado dos heurísticas que tienen como base la búsqueda tabú,

éstas son [17]: Time based Tabú Search (T2S) para el Caso Discreto del BAP y Time and Space based Tabú Search (TS2) para el Caso Continuo del BAP

### **3.5.8 Algoritmo ADARTW**

Esta heurística puede ser utilizada como una adaptación para el problema de BAP, considerando los clientes como barcos y los vehículos como puntos de atraque.

Se presenta el algoritmo de Advanced Dial a Ridewith Time Windows (ADARTW) [21], todos los servicios DRT son modelados comúnmente de esta forma, esta es una extensión del Vehicle Routing Problem (VRP). El ADARTW es una heurística constructiva basada en una técnica de inserción. Esta es conveniente para el manejo de peticiones dinámicas. Las aproximaciones de heurísticas constructivas pueden proporcionar soluciones menos óptimas rápidamente lo que es una ventaja cuando se trata de problemas con un gran espacio de soluciones posibles.

Existe un conjunto de clientes y un conjunto de vehículos. Los clientes imponen intervalos de tiempo con lo más pronto y lo más tarde posible de recogida y entrega, con esto considera ventanas de tiempo de recogida y ventanas de tiempo de bajada, en que los clientes tendrían tiempo de servicio. Cuando un cliente especifica su tiempo de bajada es asignado al límite superior de la ventana de tiempo. Con esto dando un tiempo al vehículo de servir al cliente, alcanzar el nodo destino antes del tiempo más tardío de recogida y no después.

En el modelo estático las solicitudes de un cliente ya son conocidas, en el modelo dinámico una nueva petición de un cliente puede hacerse para consideraciones inmediatas.

La búsqueda debe incluir todos los bloques de planificación contenidos en el horario de trabajo de un vehículo, el procesamiento de la factibilidad de las ventanas de tiempo se encuentra relacionado de gran manera con el modelo de horario de trabajo. Para un resultado potencial es necesario considerar diferentes restricciones, las más importantes son las ventanas de tiempo, las restricciones de capacidad y los límites de la duración de la estadía de los clientes a bordo del vehículo.

Por lo tanto, un horario de trabajo de un vehículo contendrá periodos de vehículos en utilización y periodos inactivos en el cual el vehículo se encuentra disponible y esperando.

## **3.6 Arquitectura Multiagente para el BAP**

En esta sección, se analizan las arquitecturas para el problema de asignación de barcos a muelles. Dentro de ellas se encuentra la arquitectura multiagente MADARP utilizada para el Dial A Ride Problem (DARP), que es la que se abordará con mayor detalle ya que es la arquitectura multiagente para implementar el BAP en este proyecto. Además de

la nombrada, se analiza una arquitectura multiagente específica para el BAP, aunque no es la que se va a utilizar.

### 3.6.1 Arquitectura MADARP

El BAP se puede implementar como VRP (Vehicle Routing Problem), sin dejar de tener en consideración los cambios pertinentes para el desarrollo de la arquitectura MADARP [22]. Esta arquitectura fue planteada por y diseñada principalmente para el desarrollo de sistemas multiagente para el transporte de pasajeros, sin embargo, se puede adaptar fácilmente para ocuparla en el BAP. MADARP es una arquitectura que sirve para planificar y programar en este caso las ubicaciones de los barcos a muelles en un ambiente dinámico y se constituye de una interfaz básica para el manejo de los agentes que participan.

En la Figura 3.3 se muestra la arquitectura en la que se puede apreciar dos perspectivas, la primera se define en relación a las distintas capas funcionales que posee, mientras que la segunda, lo hace en referencia a los diferentes agentes que son parte de ella.

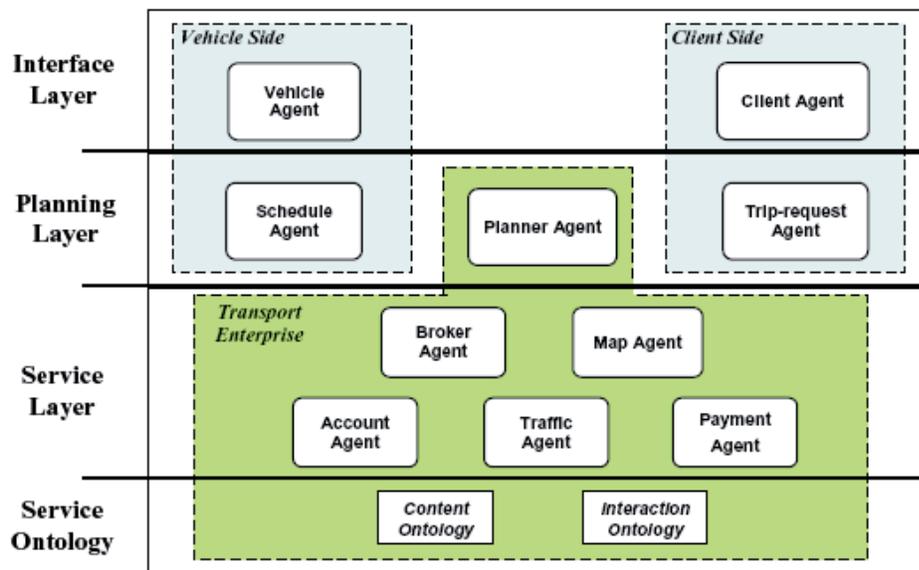


Figura 3.1 Arquitectura de agentes MADARP [22]

En lo que respecta a las capas que posee la arquitectura empleada, éstas están diseñadas para distribuir dentro de ellas todo el control de la arquitectura, y es posible identificar cuatro capas, agrupadas de acuerdo a la funcionalidad que tiene cada una de ellas, estas son:

- **Capa de Interfaz (Interface Layer):** es la capa que se preocupa de conectar y hacer que interactúe el sistema con el mundo real, que en este caso son los actores cliente y operadores de vehículo; en el caso de BAP sería barco y punto de atraque.
- **Capa de Planeamiento (Planning Layer):** Esta capa corresponde al lugar en donde se ubican los agentes que están dedicados a procesar y planear las rutas de los vehículos para el viaje y la asignación de clientes a ellos de una manera distribuida. Para el BAP, es en esta capa donde se va a planear y asignar cada barco a un muelle determinado (o punto de atraque).
- **Capa de Servicio (Service Layer):** Esta capa se preocupa de apoyar a su capa superior (capa de planeamiento) para poder cumplir con aquello que debe proporcionar funcionalidades complementarias (a través de los agentes de esa capa) que apoyen a las decisiones de los agentes de la capa de planeación.
- **Servicio de Ontología (Service Ontology):** Ésta corresponde a la capa inferior dentro de la arquitectura. Este servicio proporciona un medio de integración y cooperación entre los agentes de las diferentes capas para que puedan interactuar de manera transparente y coherente.

En la otra perspectiva del modelo, se encuentran los agentes y los actores involucrados en el sistema. En general, los agentes de interfaz entregan la entrada y la señal para que los agentes de planeamiento puedan cumplir con su labor de planear los vehículos; por su parte, los agentes de servicio ofrecen y dan la información necesaria y la ontología proporciona el estándar de conceptos para controlar las interacciones entre los agentes.

Además, es posible visualizar a los tres actores involucrados, que son los clientes, vehículos y la empresa de transporte y también a los diferentes agentes que pertenecen a cada uno de los actores. En este caso, cada actor vehículo está compuesto por el agente vehículo y el agente Schedule (o agente programador); cada actor cliente es representado por el agente cliente y el agente trip-request (o agente solicitador de viaje); y por último, el actor correspondiente a la empresa de transporte, está caracterizado por una serie de agentes y estructuras que brindan soporte a diversos servicios que sirven para la planificación del servicio de transporte y el control de éste.

Dentro de las funcionalidades que tienen cada uno de estos agentes, se encuentran las siguientes:

- El Agente Cliente es el responsable de obtener todos los requerimientos de los clientes respecto al transporte que desean y a sus preferencias en caso de imprevistos, como atrasos. Está encargado de brindar una interfaz de usuario adecuada y personalizada.

- El Agente Trip-request, toma los requerimientos y preferencias del cliente para actuar a favor de él y puede actuar como un asistente personal de viajes o como un intermediario de las decisiones que tome el cliente. Tanto el agente cliente como el trip-request hacen que haya una total comunicación e interoperabilidad con el usuario.
- El Agente Vehículo es visto como interfaz, con ello se puede comunicar con el vehículo y su conductor, entregando información acerca de contingencias, cambio de planes, etc. Además debe monitorear el estado de los vehículos y su progreso.
- El Agente Schedule tiene la tarea de manejar la ruta del vehículo y realizar cualquier ajuste en la ruta de los vehículos o en la programación de los viajes, ante una eventualidad.
- Tanto el Agente Vehículo como el Agente Schedule proporcionan la interoperabilidad entre los vehículos que ellos representan y el sistema de transporte al cual pertenece. Ambos son dependientes uno del otro, ya que el agente vehículo actúa como un sensor de lo que ocurre en el ambiente y en el vehículo, y el otro, razona para tomar las decisiones con respecto a la planificación, comunicadas mediante agente vehículo al vehículo.
- El Agente Planner hace el servicio de transporte a los clientes.
- El Agente Broker y el Agente Map colaboran para manejar bien la planificación y control, como acceso a datos, a la contabilidad, acceso a rutas, etc.

Además de las funcionalidades que se detallaron, esta arquitectura forma parte de un modelo de capas que permite ocuparlo para implementar otros desarrollos del problema de transporte de pasajero, adaptándola al nuevo modelo de problema a resolver. El hecho de adaptarla, implica el hecho de poder añadir agentes nuevos, nuevas estructuras y ontologías que permitan ofrecer lo necesario para el problema a solucionar.

En la siguiente Figura (Figura 3.4), se presenta la infraestructura para agentes, en la que se puede ver que está compuesta de tres capas: sistema de transporte concreto, arquitectura multiagente y plataforma de agentes.

En la primera capa (superior), se encuentra el sistema de transporte concreto, que es el sistema final que se desea implementar, éste se logra integrando las características que sean necesarias para el funcionamiento del sistema; para ello se puede agregar funciones a los agentes, creando nuevos agentes, etc.

En la segunda capa, se encuentra la arquitectura a utilizar, MADARP, la cual fue descrita anteriormente.

Y la última capa (capa del último nivel), es donde se encuentra la plataforma de agentes JADE (Jade Agent Platform). Esta plataforma es donde se ejecuta la arquitectura y entrega las herramientas para construir agentes, permite que se puedan comunicar (mediante el lenguaje FIPA ACL), y tiene tres componentes esenciales:

- **Sistema de Gestión de Agentes (Agent Management System):** Este sistema se encarga de brindar un servicio de “páginas blancas” en donde se identifica a cada uno de los agentes presentes, sus nombres y donde se ubican.
- **Facilitador de Directorio (Directory Facilitator, DF):** Éste, a diferencia del anterior, entrega un servicio de “páginas amarillas” donde se listan los servicios que brinda cada agente.
- **Sistema de Transporte de Mensaje (Message Transport System):** Este servicio se dedica a dar soporte a las comunicaciones entre agentes y contenedores.

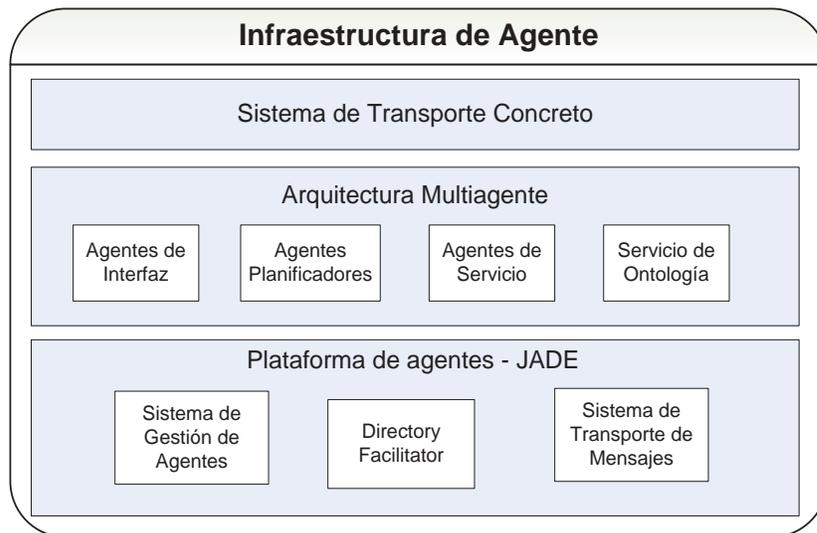


Figura 3.2 Infraestructura para agentes. [29]

La arquitectura MADARP, junto con la infraestructura de agentes descritas recientemente, son las necesarias para poder modelar e implementar una solución buscada para el BAP, lo cual se verá en el posterior capítulo.

### 3.6.2 Arquitectura MA en BAP

Existe una alternativa para la aplicación de agentes que puede dar solución al BAP. En ella se plantea una arquitectura multiagente diseñada específicamente para solucionar a través de la automatización el problema de operaciones en un terminal portuario. Ésta arquitectura [23] permite dividir el problema en varios de pequeña magnitud para disminuir la complejidad inicial, cada uno de estos sub-problemas son resueltos por agentes

específicos, los cuales tienen la particular característica de ser independientes unos de otros, tienen la capacidad de coordinarse y de comunicar los eventos y decisiones tomadas. La comunicación entre estos agentes se realiza mediante mensajes propuestos por FIPA ACL.

Dentro de esta arquitectura, se pueden diferenciar cinco tipos de agentes relacionados directamente con el puerto:

- **Agentes de Nave (Ship Agents):** Lo que este agente busca es minimizar al máximo el tiempo muerto(o de espera u ocio) de carga y descarga del barco, de las grúas, y de los costes asociados al proceso de estiba.
- **Agentes Estibadores (Stevedore Agents):** Estos agentes desean obtener la programación más apropiada para que las grúas posicionen los contenedores. Para ello debe tener conocimiento de todo el proceso de carga y descarga que afecta a la asignación de las grúas, de los barcos y de los vehículos que se dedican a transportar los contenedores; además de la información que les brindan los agentes de servicio acerca de la posición de los contenedores dentro del puerto.
- **Agentes de Servicio (Service Agents):** Básicamente lo que los agentes de Servicio tienen que realizar, es asignar el espacio apropiado para los contenedores dentro del yard (campo o patio) del puerto. Para poder planificar bien dónde van a ser asignados, el agente debe conocer el estado del patio, las características del contenedor (tipo, longitud, peso, destino, nave) y saber dónde y cómo van a ser apilados los contenedores.
- **Agentes Transtainer (Transtainer Agents):** éstos, son agentes autónomos y su funcionalidad es apilar eficientemente los contenedores y minimizar a la vez los movimientos muertos de la máquina o grúa que se esté utilizando. Para poder cumplir sus tareas, obtiene la secuencia más eficiente para mover el contenedor desde o hacia su posición correcta.
- **Agentes de Pórtico (Gate Agents):** Un agente de pórtico debe controlar la llegada y salida de una carga que venga por tierra (camiones, grúas, etc.). Para ello informa al agente correspondiente de servicio, la llegada de nuevos contenedores que necesitan almacenamiento, y la llegada de vehículos que necesiten retirar contenedores.

### 3.6.3 Arquitectura DOSS (Distributed Omni Search Strategy)

Ésta también es una alternativa de arquitectura basada en multiagente para el BAP. Aunque esta arquitectura se diseñó como solución para las distintas variantes del VRP [24], se puede implementar como arquitectura de diferentes heurísticas como algoritmos genéticos, tabú search, simulated annealing y estrategias de búsqueda en vecindades. Éste

sistema puede ser implementado por el lenguaje de programación Java y por plataformas para programación de agentes móviles.

Se compone básicamente de dos agentes para la arquitectura:

- **Agente Subasta:** Este agente debe recibir todas las demandas de los clientes que estén guardadas en un archivo o en una base de datos. Toda esta información se le envía a cada uno de los agentes vehículos. Además de recibir las demandas, les llegan todas las ofertas, las que compara para poder asignarle el cliente a un vehículo de mejor oferta o de menor costo.
- **Agente Vehículo:** Es aquel que recibe la información de todos y cada uno de los clientes por parte del agente subasta. Tiene la capacidad de aceptar o no a un cliente, siempre y cuando sea asignado por el subastador y luego lo asigna en un listado de clientes aceptados y calcula la distancia de su ruta.

Cada uno de estos agentes, está compuesto por clases, éstas son: clase de entorno (environment class), clase de comunicación (communication class), clase de habilidad (skill class), clase de identidad (identity class) y clase de cliente (customer class).

## 3.7 Trabajos relacionados

### 3.7.1 BAP Estático con Sistemas Multiagente

En este punto del capítulo se presentará un resumen del trabajo realizado por Díaz [28], en el cual está basada la solución propuesta para éste proyecto, llevándolo a la versión dinámica del BAP. Para ello se detallan las especificaciones de diseño e implementación:

- Objetivo general: Desarrollar un sistema basado en una arquitectura multiagente para la resolución del Berth Allocation Problem (BAP) y de él desprenden los objetivos específicos.
- Implementó el caso discreto del BAP.
- Los barcos no tienen prioridad, que el tiempo de descarga de un barco es directamente proporcional a su tamaño, el cual no debe exceder el tamaño de la sección de atraque más grande del muelle, el tiempo mínimo de asignación de un barco es una semana, y su tiempo máximo, un mes, durante esos periodos no existen cambios en la planificación.
- Se tiene un cierto “dinamismo”, por ejemplo, modificaciones de petición, cancelaciones, etc.; sin embargo éstas no se pueden realizar en el tiempo real.
- Utiliza PASSI como metodología de desarrollo.

- Usa como base la arquitectura MADARP, la cual se ha adaptado al problema del BAP.
- Creó los Agentes Barco y Muelle, con sus pares correspondientes, agentes petición de atraque y programador de atraque. También se incluye un agente que represente la empresa portuaria, que actúa de intermediario entre los Agentes Muelle y los Agentes Barco, además del Agente Central, quien une todo.
- Se diseñaron algunos diagramas de PASSI, como: diagrama de identificación de agentes, algunos de identificación de roles, algunos de especificación de tareas y uno de descripción de la ontología del dominio.
- Para resolver este problema de optimización se utilizó el algoritmo de inserción que es utilizado para el problema de Multiple-Dial-a Ride with time windows, y éste es adaptado al problema del BAP.
- Los resultados obtenidos en las pruebas de todo lo antes mencionado, los presenta utilizando datos propios para probar en un formato .txt, para la recolección de datos se consideró un muelle con 10 secciones, y la llegada de 50 barcos en un horizonte de tiempo máximo de 60 unidades de tiempo. Cada sección o punto de atraque se demora 1 unidad de tiempo en estar lista, y no se consideran restricciones de espacio, por lo que cualquier barco puede ser atendido por cualquier punto de atraque.
- El archivo de salida, también un .txt, señala los resultados de asignación de barcos a los distintos puntos de atraque, mostrando cada muelle, su identidad, la cantidad de barcos asignados, el costo total de la secuencia, y el costo de los tiempos ociosos de cada muelle. Se generan 5 escenarios posibles basados en 2 costos básicos.
- Los resultados finales son presentados en cinco gráficos. En éstos, se puede observar que la heurística intenta mantener un equilibrio natural asignando entre 3 a 8 barcos a cada muelle, como media siendo unos 5 barcos por muelle.
- Para finalizar, todos los gráficos que muestran los resultados de simulación, llevan a la conclusión de que los resultados caen dentro de lo esperado, que es propuesta una forma simple de implementación que permite distintos escenarios, tanto como escenarios donde se ve una distribución equitativa de la carga de trabajo en los distintos muelles o el escenario de minimizar los tiempos de ocio de cada uno de ellos.

### **3.7.2 D-DARP con Sistemas Multiagente**

En este trabajo hecho por [30] se realiza una solución al problema de optimización del dynamic dial a ride with time Windows.

- Objetivo general: Desarrollar un sistema para el Dynamic Dial-A-Ride Problem (DDARP) utilizando la tecnología de agentes.
- Implementa el caso dinámico del dial a ride problem.
- Propone una nueva arquitectura para resolver este, que es la arquitectura MADARP, la diferencia de esta con la anterior es la implementación de una capa de monitoreo para lograr el dinamismo en la solución del problema
- El módulo de monitoreo tiene por finalidad llevar el seguimiento de la ejecución de los eventos dinámicos y de la simulación propiamente tal en tiempo real.
- Utiliza PASSI como herramienta de modelado y se presentan algunos modelos, como lo son descripción de dominio, casos de uso narrativos, diagrama de identificación de agentes, diagrama de identificación de roles, diagrama de especificación de tareas, el modelo de sociedad de agentes, modelo de implementación de agentes.
- Para la planificación de pruebas, realizó pruebas de caja negra y caja blanca, crea pruebas para las rutas estáticas y las dinámicas, los eventos externos al cliente y los eventos externos a los vehículos, coordinación entre componentes y monitoreo de los vehículos. Para esto se crean diversos archivos de formato txt, con datos de vehículos, clientes, un mapa, etc.
- Para las pruebas compara la heurística anterior con la heurística propuesta por ellos, que incluye los agentes de monitoreo, creando distintos escenarios, en dos plataformas, uno en plataforma local, y otro en plataforma distribuida, mirando distintos escenarios, como lo son: comparación de usuarios atendidos, de vehículos utilizados y comparación de costos totales.
- Al evaluar el monitoreo se simuló una mañana de trabajo normal, de 6:00 AM a 9:00PM, los resultados arrojados fueron satisfactorios.

## Capítulo 4: Solución Propuesta

### 4.1 Problema Estático o Dinámico

Como se definió en la sección de análisis de la literatura del documento, el BAP, al igual que otros problemas similares de planificación, existe en versión estática o dinámica, dependiendo de qué tipo de solución se quiera llevar a cabo y según las restricciones de tiempo que se consideren.

Para tomar una decisión respecto a con cuál de estas dos opciones se implementará el proyecto, es muy importante tener en consideración la información con la que se pretende trabajar. Respecto a esta información, se puede decir que si se conoce en su totalidad a los barcos que requieren atracar en algún momento, estamos en presencia de un problema estático; en cambio, si falta un poco de información o suceda algo inesperado en el transcurso en el que el barco solicite el servicio, se habla de que se trata de un problema dinámico, ya que en el dinamismo son considerados retrasos, cancelaciones, accidentes, o simplemente que un barco no llegue, situaciones en las cuales implica que se deba hacer una re planificación.

En el problema dinámico, es muy importante el factor tiempo, por lo que la solución propuesta debe poder ejecutarse rápidamente, teniendo en cuenta además que el barco que requiera y solicite el servicio, pueda tener una respuesta de su solicitud en un periodo de tiempo razonable. En el caso del problema estático, el factor tiempo no es relevante, esto se debe a que las peticiones de servicio son conocidas desde antes, como por ejemplo las reservas anticipadas de algún servicio.

Según lo anterior, se puede definir que en este trabajo, se abordará el BAP en su versión dinámica (DBAP), considerando con esto que no es necesario que los barcos hayan llegado para ocupar muelle (o estén esperando), sino que puedan realizar una petición en cualquier momento, fijando su hora de llegada al puerto y con ello asignarle previamente un espacio dentro del muelle.

### 4.2 Modelo Discreto, Continuo o Híbrido

Los distintos modelos del BAP, también fueron estudiados en el capítulo anterior, y se diferencian entre ellos en el ámbito espacial del muelle, o sea cómo este puede ser dividido para que los barcos atraquen.

Para determinar cuál de estos modelos es el que se utilizará en este proyecto, se ha tomado en cuenta la complejidad computacional que conlleva ejecutar cualquiera de ellos. En el caso del modelo continuo e híbrido, se utiliza mejor el espacio del muelle, sin embargo, es mucho más complicado llevarlo a cabo en el ámbito computacional, en comparación con el modelo continuo.

Según lo especificado, se ha decidido implementar el modelo discreto, el cual divide el muelle en un número de secciones de manera igualitaria, y cada espacio recibe un barco a la vez. El tamaño del muelle, está relacionado directamente con el tamaño de los barcos (el promedio de ellos), o sea, se asume que los barcos son relativamente del mismo tamaño, por lo que las secciones del muelle quedan acorde a ellos.

### **4.3 Metodología de Desarrollo Elegida**

Existe una gran variedad de metodologías de desarrollo para un sistema multiagente, de las cuales se debe definir cuál es la mejor propuesta a definir. Dentro de ellas, se evaluaron las características que las definen y se ha decidido optar por la metodología PASSI.

Uno de los motivos que llevó a la conclusión de elegir PASSI, es el respaldo que tiene esta metodología por parte de la organización FIPA, quien trabaja para estandarizar el desarrollo de proyectos con tecnología de agentes.

Otra razón para la elección, es que se cuenta con una herramienta de modelado disponible para los usuarios (gratis desde la página principal de PASSI), PASSI Toolkit (PKT), pudiendo utilizar mediante ello, el software Rational Rose para poder crear los modelos, herramienta que además permite el soporte de UML (lenguaje de modelado unificado).

No está demás mencionar que PASSI es comparativamente fácil de aprender y utilizar, lo que no es un obstáculo para implementarla. Además, personalmente se tiene un mayor conocimiento y experiencia de desarrollo acerca de esta metodología, lo que hace que ésta sea mucho más factible de implementarla y adoptarla al desarrollo de un MAS para el BAP.

### **4.4 Herramientas de Desarrollo**

Para poder implementar y desarrollar el sistema, se utilizarán las siguientes herramientas:

- Como se mencionó en la metodología propuesta, que es PASSI, y el software que se utilizará para el modelado del sistema es StarUML.
- Como entorno de desarrollo integrado, o IDE (Integrated Development Environment), será utilizado Netbeans versión 7.0.1, el cual posee como predeterminado el JDK (Java Development Kit) versión 1.7. para el desarrollo e implementación del programa.
- Para la infraestructura, se utilizará la plataforma para agentes JADE, en su versión más reciente (JADE versión 4.1.1.)[59]. JADE (Java Agent Development

Framework) es un software implementado en su totalidad en lenguaje Java, permite desarrollar sistemas multiagente y que además se cumplan las especificaciones FIPA, ocupa FIPA ACL como lenguaje para representar los mensajes.

## 4.5 Diseño de la Arquitectura Propuesta

Dentro de las arquitecturas mencionadas y definidas con anterioridad, se ha decidido implementar la arquitectura MADARP, arquitectura diseñada para sistemas multiagente del Dial a Ride Problem, la cual ya ha sido abordada por el trabajo relacionado al BAP [28], por lo que se han considerado ambas para el diseño de la nueva propuesta de arquitectura.

Para poder llevar a cabo el diseño de MADARP, se ha considerado hacer una adaptación específica de ésta, al problema de asignación de barcos a muelle. Esta adaptación se basa simplemente en cambiar algunos agentes (tanto nombres, como la funcionalidad que ellos deberán cumplir para la satisfacción del problema), pero tratando de mantener la estructura original de dicha arquitectura.

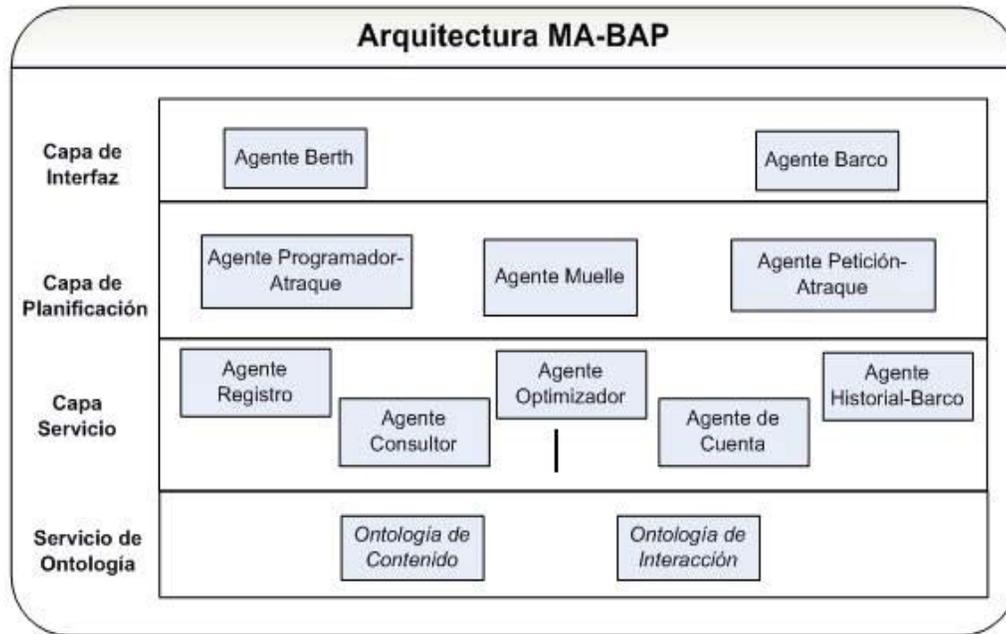
En lo que respecta a las capas que componen MADARP, éstas sufrirán una modificación en el modelo propuesto, por lo que afecta en cierta medida a su estructura inicial. Por lo tanto, se tiene la capa de interfaz, la capa de planificación, la capa de servicio y el servicio de ontología, pero además se agrega una capa de monitoreo, quedando la arquitectura propuesta con un total de cinco capas, lo que está afecto a diversos cambios son los agentes que se contienen dentro de ellas y los actores que van a interactuar con ellos de una u otra manera.

En cuanto a la capa de servicio de ontología, ésta no presenta ningún cambio, o sea sigue funcionando con una ontología de dominio y una de interacción, proporcionando los medios necesarios para la comunicación, integración y cooperación entre los agentes y el mundo real.

La capa de servicio tampoco se ve afectada según la propuesta inicial de [28], quedando con los mismos agentes: Agente Registro, Agente Consultor, Agente Optimizador, Agente de Cuenta y Agente Historial Barco, quienes apoyan a los agentes de la capa de planificación.

Sin embargo, la arquitectura especificada, no cumple con lo que se desea plantear como solución dinámica del BAP, por lo que nos hemos basado en ella, pero a la vez, se han propuesto varias modificaciones.

En la Figura 4.1, se presenta la antigua arquitectura, propuesta por [28] para el BAP.



**Figura 4.1 Arquitectura MA BAP**

La modificación más simple propuesta es considerar a los barcos como clientes, ya que son ellos quienes solicitan el servicio del muelle, o sea, el punto específico donde deberán atracar; y considerar a su vez, las secciones del muelle como los vehículos que ofrecen un servicio, que en este caso, el servicio corresponde al proceso completo en que el barco se encuentra en el muelle y se procede a la carga y/o descarga de los contenedores mediante las grúas. Es así como ha quedado compuesta la capa de interfaz, con el Agente Barco y el Agente Berth. Dichos agentes son los que cumplen la funcionalidad de interactuar directamente con los actores involucrados, conectándose con el mundo real, mostrando la ejecución del sistema mediante una interfaz gráfica. Para esta etapa del proyecto, dicha capa no será implementada.

El hecho de plantear dinámicamente el problema, implica considerar la variable tiempo para la ejecución de todo el proceso en tiempo real. Por ende, un gran punto a evaluar dentro de la arquitectura propuesta, es el hecho de que ésta soporte modificaciones dentro del proceso de la asignación; una modificación drástica que podría ocurrir en tiempo real tener que hacer una re planificación debido a cualquier percance que pueda suceder. Debido a esto, se vio la necesidad de realizar un cambio en el sentido de las capas de la arquitectura, agregando una nueva: la capa de monitoreo. Dentro de ella, es necesario crear y agregar dos nuevos agentes, entre los que se encuentran el Agente Simulador Barco y el Agente Simulador Muelle.

La funcionalidad de la capa de monitoreo es muy importante, ya que es la que va a permitir observar el comportamiento del sistema, respecto a los eventos que sucedan en los barcos y en el berth, logrando simular las acciones de cada uno de ellos mediante los agentes simuladores, y tener un control sobre el proceso de planificación del BAP, todo esto conlleva a que el sistema funcione de una manera más efectiva en su dinamismo, permitiendo modificaciones y re planificaciones en tiempo real.

Respecto a la capa de planificación, también sufre un cambio, ya que en ella se consideran el Agente Programador Atrache, el Agente Central y el Agente Petición Atrache, quienes se van a encargar de llevar a cabo la funcionalidad de esta capa, o sea, todo el procesamiento, comunicaciones y la planificación de la asignación de barcos a muelles.

Los agentes involucrados dentro del sistema cumplen con diferentes funciones, a continuación se detallan las funcionalidades de cada uno de ellos:

#### **4.5.1 Agente Barco**

- Es el responsable de obtener todos los requerimientos de los clientes reales, peticiones, cancelaciones, preferencias, etc.
- Interactuar con el usuario, proporcionándole una interfaz personalizada.
- Informar acerca de estado de la petición al cliente real.
- Informar los eventos que ocurran al cliente real.

#### **4.5.2 Agente Muelle**

- Representar un rol de interfaz.
- Brindar toda la información necesaria respecto a la asignación, situaciones de contingencias, cambios, etc.
- Encargado de informar el estado del muelle.
- Informar acerca del progreso del muelle durante un periodo de tiempo.

#### **4.5.3 Agente Simulador Barco**

Los clientes reales, que para el BAP son los barcos, pueden tener los siguientes eventos:

- Una petición de atraque.
- Cancelar en cualquier instante una petición de atraque.
- Modificar una petición de atraque respecto al horario que le acomode llegar.

- Pueden atrasarse en su llegada.
- Pueden no llegar.

Algunos de estos eventos pueden suceder, provocando una re planificación, por lo que el Agente Simulador Barco, debe tener la capacidad de simular estos actos para comunicárselo a los demás agentes y no provocar conflictos en la asignación real de los barcos.

#### **4.5.4 Agente Simulador Muelle**

El Agente Simulador Muelle, también debe estar atento a todos los cambios que sufre el muelle dentro de un período, informando en tiempo real acerca de los espacios que estén disponibles dentro del muelle, si están ocupados, desocupados, los tiempos de llegada, de salida o de proceso de un barco, si los tiempos se encuentran dentro de los límites que se encuentran preestablecidos, si el muelle está en correcto funcionamiento, la inasistencia de un barco, etc.; para todo aquello lo cual involucre una re planificación de la asignación.

#### **4.5.5 Agente Programador Atraque**

La función principal de este agente, es servir de comunicación entre el Agente Muelle y el Agente Simulador Muelle, junto con comunicarse, a la vez, con el Agente Planificador, intercambiando información para una programación efectiva. A parte, entre sus tareas se encuentran:

- Generar una propuesta de atraque, la cual indica todas las posibilidades que se tienen para que el barco atraque dentro del muelle según su propia petición.
- Llevar a cabo la programación de la secuencia de barcos en un muelle, tarea en la que se lleva a cabo la inserción.
- Gestionar las eventualidades que pueden suceder dentro del muelle, como por ejemplo, que no esté en funcionamiento, que el servicio otorgado dentro de él a los barcos se atrase o adelante, etc.
- Gestionar las eventualidades externas al muelle, o sea, las que se informan acerca de las eventualidades que le han sucedido a algún barco.

#### **4.5.6 Agente Planificador**

- Gestionar todas las eventualidades, tanto las del muelle como las del barco, y posterior a ello informar de ellas.
- Gestionar peticiones del barco y enviar las propuestas, para que tengan conocimiento sobre qué petición desea el barco.
- Gestionar propuestas del muelle, eligiendo la mejor, y enviar la respuesta al barco acerca del estado de su petición.

#### **4.5.7 Agente Petición Atraque**

La función principal de este agente, es servir de comunicación entre el Agente Barco y el Agente Simulador Barco, junto con comunicarse, a la vez, con el Agente Planificador, intercambiando información para una programación efectiva. A parte, entre sus tareas se encuentran:

- Gestionar las eventualidades externas al barco, o sea, las que se informan acerca de las eventualidades que le han sucedido dentro de un muelle.
- Debe ser capaz de negociar la propuesta de atraque brindada por el Agente Programador Atraque, para ello toma en cuenta la petición realizada por el barco, en la cual se encuentran sus preferencias.
- Gestionar las peticiones del barco, en la cual se analizan sus preferencias de atraque como hora, lugar, etc.
- Gestionar las eventualidades del barco, como lo son cancelaciones de peticiones, modificación, retrasos en la llegada, etc., todas aquellas son monitoreadas por el Agente Simulador Barco.

Las funciones que cumplen cada uno de los agentes que componen la arquitectura propuesta para el BAP, son primordiales para el buen funcionamiento del sistema en general y para poder obtener los resultados esperados como solución al problema.

Para comprender mejor y como una explicación más específica de los agentes nombrados, se presenta la nueva arquitectura propuesta, la cual está dada de la siguiente manera (Figura 4.2):

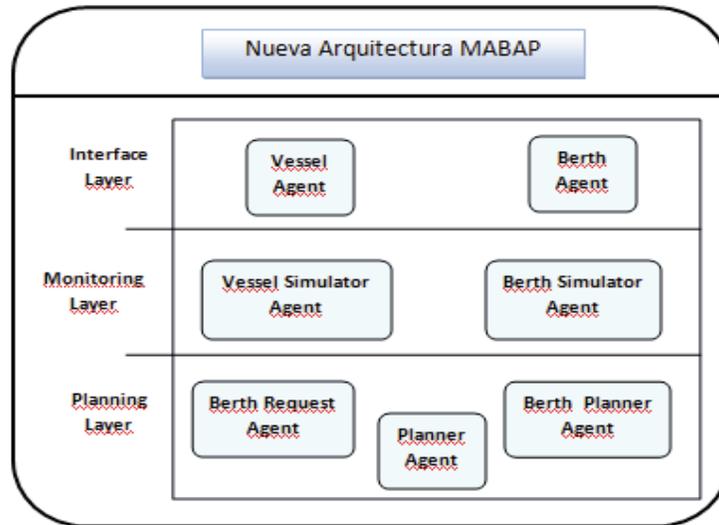


Figura 4.2 Arquitectura MABAP Propuesta

## 4.6 Diseño de la Heurística Propuesta

La heurística que se va a utilizar se basa en la propuesta de Jaw et al [27] y en una versión mejorada y distribuida de ella planteada por Cubillos et al [22]. ADARPTW es una heurística greedy constructiva, en la cual el algoritmo de planificación es una parte esencial, ya que permite evaluar y así encontrar todas las formas posibles de insertar un nuevo barco dentro de un muelle, eligiendo al que entrega la máxima utilidad de acuerdo a la función objetivo determinada; una vez que se ha seleccionado de manera factible la inserción del barco, se definen los tiempos reales de recogida y entrega, definiendo la posición exacta en donde será posicionado sin intervenir las ventanas de tiempo definidas para cada barco.

A continuación se explica en detalle el algoritmo de inserción definido por los autores ya nombrados para el DARP, y las adaptaciones que se realizaron para definir la heurística propuesta para el BAP.

### 4.6.1 Especificación de Conceptos

El principal objetivo del proyecto, es minimizar la función objetivo propuesta (ver capítulo 3.4), para ello, la inserción de un barco y elección del lugar de atraque, son una parte fundamental para la implementación del proyecto.

Para poder entender con mayor claridad el algoritmo de inserción que se explicará más adelante, hay que entender algunos conceptos referentes al DARP, que serán utilizados de manera distinta en el BAP, pero que son los utilizados como la base para el dilema de la

inserción de un barco; éstos son: clientes de entrada y de salida, estados del vehículo, bloques de planificación, ventanas de tiempo [29].

#### **4.6.1.1 Clientes de Entrada y Clientes de Salida**

Los clientes entrantes (inbound customers), corresponden a aquellos clientes que están esperando en un lugar, en un determinado tiempo (a una hora específica), que requieren ser recogidos y trasladados hacia un punto de destino. En este caso es aceptable que el vehículo llegue después de la hora estipulada (pequeña demora) para recogerlos, pero no se acepta que llegue antes de la hora estipulada, ya que el cliente no está preparado en ese instante para que sea recogido.

Los clientes salientes (outbound customers), corresponden a aquellos clientes que desean llegar a algún lugar específico a una hora determinada. Por lo tanto, al cliente no le importa la hora en que sea recogido, pero si le interesa que el vehículo que lo recoja lo deje en el lugar de destino necesariamente a la hora que él requiera.

En el BAP, los clientes corresponden a los barcos, y se debe tomar en cuenta el caso en que los barcos son clientes de entrada, esto se debe a que el barco debe llegar al muelle a una hora determinada, que es lo relevante para este caso, y luego realizar su proceso de carga y descarga para posteriormente irse en cuanto su servicio concluya, el cual también está determinado por una hora, pero varía según el servicio.

#### **4.6.1.2 Estados del Vehículo**

En el transporte de pasajeros, se dice que un vehículo puede estar en tres diferentes estados: en depósito, en viaje o inactivo. Que el vehículo esté en depósito significa que no ha comenzado su servicio o el servicio ha terminado. Que el vehículo esté en viaje significa que está en su tiempo de servicio, o sea, que se dirige a recoger o dejar pasajeros. El tercer estado, es cuando el vehículo se encuentra inactivo, o sea, el vehículo se encuentra detenido y esperando a transportar a un pasajero. En el último caso, al tiempo de inactividad del vehículo se le conoce como slacks, pero en este contexto, un vehículo nunca estará inactivo con pasajeros a bordo, pues los slacks se ubican entre bloques.

En el BAP, un slack corresponde al tiempo que está entre en que berth está listo para recibir un nuevo barco y que el barco llegue, por lo tanto en un puerto (a menos que el puerto posea una tasa de llegada muy alta), siempre existirán slacks, por lo que es importante que éstos sean considerados en los bloques de planificación, ya que gracias a éstos se podrá identificar cuando una solución es mejor que otra.

Por otra parte, se puede definir si un servicio será llevado a cabo en un solo vehículo (conocido como 1-DARP), o el caso en que exista una flota de vehículos disponibles para realizar el servicio (m-DARP); a su vez se puede ver la posibilidad de atender a varios

pasajeros a la vez dentro de un mismo vehículo. Para el BAP, se utilizará la segunda variante, donde existen múltiples vehículos, ya que en un puerto existen múltiples berths que pueden llevar a cabo el servicio solicitado por un barco, y cualquiera de ellos puede ser el seleccionado para hacerlo; sin embargo, solamente se podrá atender a un barco por atracadero a la vez.

#### 4.6.1.3 Bloques de Planificación

La ruta que desarrolla un vehículo puede estar conformada por bloques de planificación. Los bloques de planificación corresponden a un horario de trabajo de un vehículo y se caracterizan por contener eventos de recogida de clientes (pickup) y eventos de entrega de clientes (delivery) en una ruta del vehículo, o sea, un bloque contiene todas las paradas que realiza el vehículo (como máximo puede existir dos paradas por cliente) y comienza siempre con el vehículo en camino a recoger un cliente y termina cuando el último cliente a bordo es dejado en su destino.

En consecuencia, dentro de la planificación se consideran los períodos de utilización del vehículo, que corresponde a los bloques de planificación, y además los períodos inactivos del vehículo o tiempo de slack, los cuales se ubican entre bloques, como se muestra en la Figura 4.3.

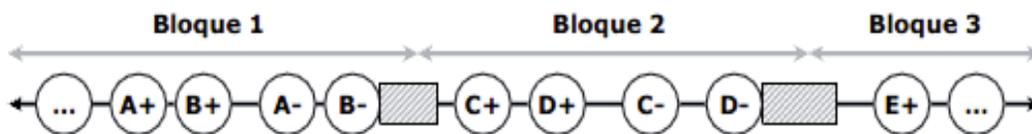


Figura 4.3 Estructura de un bloque de planificación [29].

#### 4.6.1.4 Ventanas de Tiempo

Una ventana de tiempo (time window), corresponde a un intervalo de tiempo dentro del cual un cliente es recogido o entregado en un lugar, en donde se deben definir sus rangos inferiores y superiores, esto dependerá de las variables a las que esté afecto el problema. Por lo tanto, la factibilidad de ellas se encuentra relacionada directamente con el modelo del horario de trabajo.

En la Figura 4.4 se puede apreciar las ventanas de tiempo que están presentes en el transporte de pasajeros (la figura denota una ventana de tiempo para clientes de salida), una de ellas corresponde al pickup del cliente, la cual tiene sus extremos definidos (límite inferior y límite superior), asignados según su tiempo más temprano de recogida (EPT<sub>i</sub>, earliest pickup time), y su tiempo más tardío de recogida (LPT<sub>i</sub>, last pickup time); mientras

que la segunda ventana concierne al delivery del cliente, también con sus extremos definidos, pero en este caso según sus tiempos de entrega ( $EDT_i$ , earliest delivery time y  $LDT_i$ , last delivery time). En la Figura 4.4, también se añade una restricción referente al tiempo de duración de viaje de un pasajero, mencionada como  $MRT_i$  (Maximum Ride Time), que corresponde al tiempo máximo de viaje aceptable para el cliente  $i$  y tiene un valor específico para cada cliente, y como  $DRT_i$  (Direct Ride Time), que es el tiempo directo para el viaje, visto mejor como el camino más corto necesario de viaje; además de ello, se encuentran los parámetros  $A$ ,  $B$  y  $WS$  (desviación máxima aceptable de un cliente frente a su tiempo deseado de recogida o entrega), que permiten que los tiempos no sean tan restringidos, variando la calidad de servicio prestada. A partir de esta explicación, se puede añadir que:

$$EPT_i = EDT_i - MRT_i \text{ y}$$

$$LPT_i = LTD_i - DRT_i.$$

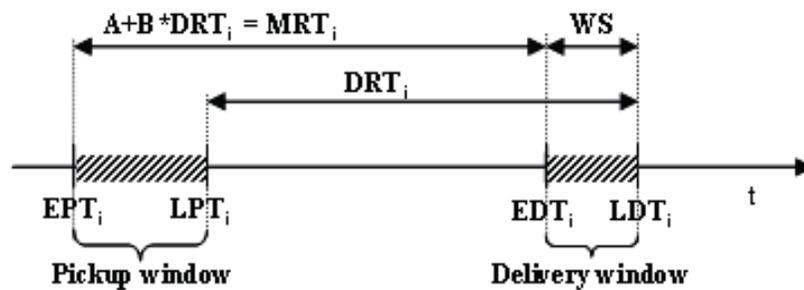


Figura 4.4 Construcción de una ventana de tiempo [22].

#### 4.6.2 Algoritmo de Inserción

La heurística de inserción que se utilizará está basada en una propuesta realizada por Jaw et al. [27], en la cual los autores evalúan en cuánto puede ser anticipado o postergado el tiempo de llegada de un vehículo, para así poder evaluar una posible inserción. Para el caso del BAP, el tiempo de llegada de un vehículo corresponde al tiempo en que un muelle esté disponible, puede ser anticipado cuando esté disponible antes y puede postergarse cuando se desocupe después del tiempo estipulado, por lo que el barco quedará esperando más tiempo por el muelle. El procedimiento para evaluar la inserción de un cliente, que se propone en el trabajo [22], se define así:

Para un bloque  $X$ , con eventos que representan una recogida o una entrega de clientes, los siguientes cálculos representan el tiempo que aquellos eventos pueden ser adelantados y retrasados en el tiempo.

$$\text{BUP}(X_i) = \text{Min}(\text{Min}(\text{AT}(X_i) - \text{ET}(X_i)); \text{SLK}_0)$$

$$\text{BDOWN}(X_i) = \text{Min}(\text{LT}(X_i) - \text{AT}(X_i))$$

$$\text{AUP}(X_i) = \text{Min}(\text{AT}(X_i) - \text{ET}(X_i))$$

$$\text{ADOWN}(X_i) = \text{Min}(\text{Min}(\text{LT}(X_i) - \text{AT}(X_i)); \text{SLK}_{w+1})$$

En donde se tiene que:

- $0 < i < w+1$ .
- $\text{SLK}_0$  y  $\text{SLK}_{w+1}$  representan los posibles períodos de slacks, antes y después del bloque.
- $\text{ET}(X_i)$  es el tiempo más temprano de llegada del evento  $X_i$  (early time), y corresponde al extremo inferior de la ventana de tiempo.
- $\text{AT}(X_i)$  es el tiempo actual de llegada del evento  $X_i$  (actual time), o sea, el tiempo que se programó para la llegada del vehículo.
- $\text{LT}(X_i)$  es el tiempo más tardío de llegada del evento  $X_i$  (latest time), y corresponde al extremo superior de la ventana de tiempo.
- $\text{BUP}$  y  $\text{BDOWN}$  se calculan a base a todos los eventos anteriores a  $X_i$ .
- $\text{AUP}$  y  $\text{ADOWN}$  se calculan a base a todos los eventos posteriores a  $X_i$ .

En la propuesta, además se agrega realizar operaciones de intersección a las restricciones de las ventanas de tiempo, para facilitar el proceso de evaluar la factibilidad de las mismas y revisar posibilidad de inserción de un nuevo cliente. La intersección de las ventanas consiste en comparar sus extremos, y evaluar su factibilidad por medio de ello.

En la Figura 4.5 se muestra la evaluación de inserción del cliente  $X$  en su evento de pickup ( $X+$ ) y delivery ( $X-$ ), se ve también que el bloque de planificación es dividido en tres sub-bloques de planificación, A, B y C, los cuales determinan los eventos que ocurren antes de la recogida del cliente  $X$ , entre los eventos y después de la entrega, respectivamente. Los eventos que unen cada uno de estos bloques, son puntos en los que se consideran las intersecciones de las ventanas obtenidas.

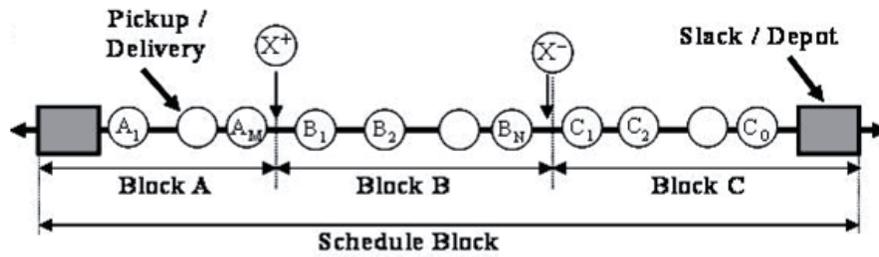


Figura 4.5 Evaluación para la inserción de un nuevo cliente. [22]

La Figura 4.6 presenta las ventanas de tiempo y las distancias necesarias para evaluar la inserción del cliente X y la intersección de las ventanas.

Al igual que en la Figura anterior 4.5, se divide el bloque de planificación en tres sub-bloques A, B y C. El intervalo  $[ETA, LTA]$ , indica los tiempos más tempranos y tardíos para que  $A_M$  sea desplazado sin intervenir en la restricción de las ventanas de tiempo de los eventos del bloque, o sea, los anteriores a  $A_M$ ; mientras que para el intervalo  $[ETB, LTB]$  y  $[ETC, LTC]$ , sucede lo mismo pero en referencia a sus sub-bloques, B y C respectivamente.

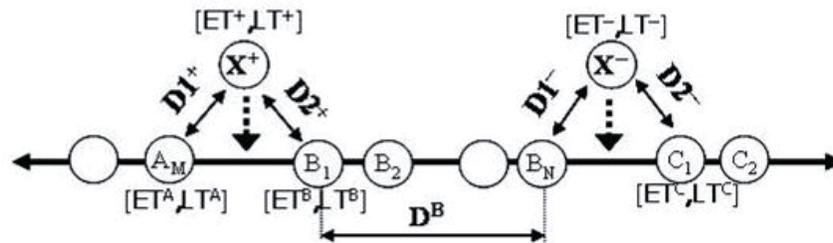


Figura 4.6 Representación de los elementos necesarios para la nueva inserción [22].

Es necesario identificar la factibilidad de desplazar hacia arriba y hacia abajo para cada uno de los bloques., para el bloque A se utiliza el BUP y BDOWN del evento  $A_M$ , en el que se consideran los eventos previos, para el bloque C se ocupa el AUP y ADOWN de  $C_1$ , y para el bloque B también se necesita el AUP y ADOWN para  $B_1$ , pero considerando solo hasta  $B_N$  y no los eventos del bloque C. entonces para el intervalo  $[ETA, LTA]$ , se tiene  $ETA=AT(A_M)-BUP(A_M)$  y  $LTA=AT(A_M)+BDOWN(A_M)$ , sucede lo mismo con los intervalos de b y c.

Adicional a los intervalos mencionados, se consideran  $D1+$ ,  $D2+$ ,  $D1-$ ,  $D2-$ , correspondientes a las distancias entre los nodos involucrados.

Definidas ya las ventanas y las distancias, el siguiente paso es la intersección de los intervalos de tiempo de los tres sub-bloques y las dos ventanas de tiempo provenientes de los eventos de recogida y entrega del cliente. Esta intersección necesita considerar las

distancias que separan los cinco intervalos. Por esta razón es usado como referencia un punto en la planificación obteniendo un solo intervalo de tiempo [ET, LT]. Al utilizar el caso del evento pickup del cliente (X+) como punto de referencia, los cálculos se pueden obtener según:

- $ET = \text{Max} (ETA+D1+; ET+; ETB - D2+; ET- - D1- - DB - D2+; ETC - D2- . D1- - DB - D2+)$
- $LT = \text{Min} (LTA + D1+; LT+; LTB - D2+; LT- - D1- - DB - D2+; LTC - D2- - D1- - DB - D2+)$

Entonces, si el intervalo [ET, LT] resulta válido, va a representar el área de factibilidad respecto al punto de referencia, que es el evento X+.

En síntesis, se tienen todos los elementos necesarios para la inserción de un nuevo cliente ya especificados, el procedimiento a utilizar para la intersección de las ventanas en cuestión es:

Primero se debe dejar fija la ventana de tiempo del evento X, y desplazar todas las demás en dirección a ese evento. Para la evaluación de eventos independientes, va a corresponder a DRT entre el nodo de la ventana desplazada y el nodo de X más el tiempo de servicio asociado y para el caso de las intersecciones en una ruta, se deben considerar las distancias de los nodos intermedios.

### 4.6.3 Algoritmo de Inserción para el BAP

Después de analizar y estudiar todos los conceptos relacionados al dilema del transporte de pasajeros, junto con la inserción de un nuevo evento dentro de una ruta de un vehículo (punto clave de la heurística), se ha podido originar una nueva heurística que abarque el problema de asignación de barcos a muelles, para ello se realizaron modificaciones a la heurística original creada para el DARP, la cual fue tomada como base para lograr el algoritmo de inserción para el BAP.

Para que sea posible llevar a cabo dicho algoritmo de inserción para el BAP, es necesario tener un agente (definido en el punto de arquitectura propuesta) que sea capaz de soportar toda la lógica para evaluar y posteriormente insertar un nuevo barco dentro de un muelle escogido, éste es el Agente Programador Atraque, al cual le llega toda la información necesaria para efectuar el algoritmo.

Para poder comprender y aclarar los cambios que se necesitaron para crear la nueva heurística, se explicará detalladamente el nuevo algoritmo de inserción de un nuevo barco para el BAP.

Como se ha mencionado en párrafos anteriores dentro de la explicación de la heurística para el DARP, se encuentran los siguientes conceptos definidos para el BAP que no se deben dejar de mencionar:

- Los clientes corresponden a los barcos y los vehículos corresponden a los muelles.
- En el BAP se considera el concepto de clientes de entrada, ya que se requiere una hora determinada para llegar al muelle, según la solicitud de atraque hecha por el barco.
- Un slack corresponde al tiempo que está entre que el muelle está listo para recibir un nuevo barco y que el barco llegue definitivamente.
- En un puerto existen múltiples muelles que pueden llevar a cabo el servicio solicitado por un barco. La diferencia con el m-DARP, es que un atracadero puede atender un solo barco a la vez.
- En la planificación se consideran los períodos de utilización del muelle y el tiempo de slack, ésta será considerada día a día.
- Un bloque de planificación, corresponde a todas las llegadas e idas de un barco dentro de un muelle en específico y en un determinado tiempo, dividida en sub-bloques que será la atención de un solo barco.
- La ventana de tiempo es un rango dado para que el barco llegue y se retire del muelle, en donde también se define el tiempo más temprano (ET) y más tardío (LT), necesario para cumplir con su proceso.
- A diferencia del problema de transporte, en el BAP, no se considera la variable de distancias, ya que no existen distancias asociadas entre muelles.

El primer cambio realizado es el bloque de planificación, ya que éste no es el mismo, esto se debe a que los dos casos, DARP y BAP, no son lo mismo, ya que los eventos en cuestión se manejan de manera distinta; en el primero, los clientes en sus eventos de pickup y delivery, si bien suceden dentro del mismo bloque, no necesariamente están juntos uno al lado del otro, o sea, se sube y se baja un cliente con o sin otro pasajero intermedio de estos dos eventos; pero en el BAP, estos eventos ocurren dentro de un mismo muelle, y necesariamente juntos, ya que un barco atraca, se le entrega el servicio (embarque y/o desembarque) y se va del muelle y el muelle está disponible en ese instante solamente para atender un solo barco. En lo que respecta al tiempo de slack, éste se producirá al momento de comenzar cada sub-bloque, por lo tanto el paso de un sub-bloque a otro, siempre generará un tiempo de inactividad (estado en que el muelle se encuentra sin servicio de un barco).

En la Figura 4.7, que se muestra a continuación, se ven los eventos de llegada e ida de un barco, A+ y A-, respectivamente, B+ y B-, C+ y C-, y así sucesivamente; los tiempos de

slack producidos antes del evento de llegada y un horizonte de planificación para un muelle en específico (bloque completo), el cual durará 24 horas.

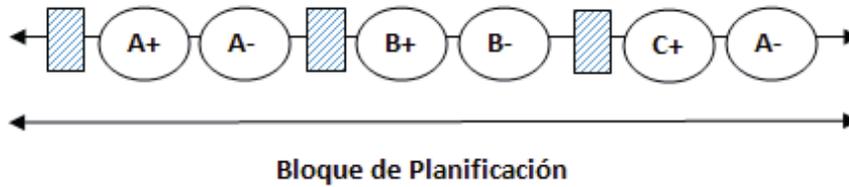


Figura 4.7 Bloque de Planificación para el BAP

El segundo cambio que se efectúa es en las ventanas de tiempo. Existen dos ventanas de tiempo asociadas a un determinado barco, éstas afectan a la llegada o arribo del barco al berth (momento del atraque) en su tiempo más temprano [EAT, earliest arrival time] y tardío [LAT, latest arrival time] y también a su ida o partida en su tiempo más temprano [EDT, earliest departure time] y tardío [LDT, latest departure time]. Estas ventanas se pueden visualizar en la Figura 4.8:

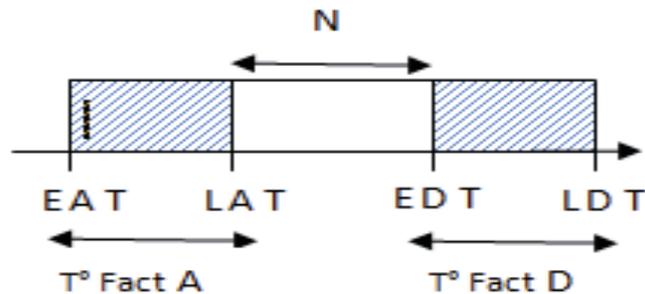
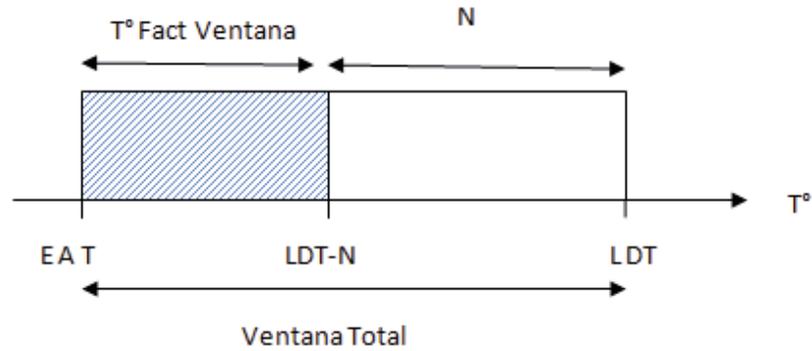


Figura 4.8 Ventanas de Tiempo asociadas al Barco

No obstante, éstas dos ventanas de tiempo delimitadas por [EAT, LAT] y [EDL, LDT], se van juntar formando una sola. Por lo tanto, quedará una sola ventana de tiempo [EAT, LDT], en donde se obtiene una ventana de tiempo factible, la cual corresponde al tiempo más temprano de llegada [EAT] y al tiempo más tardío de partida [LDT], menos el tiempo de servicio o procesamiento, denotado  $N$ , que necesita el barco, este tiempo factible es el que servirá posteriormente para la intersección. En la Figura 4.9, se muestra la ventana de tiempo resultante que se tiene para resolver el BAP.



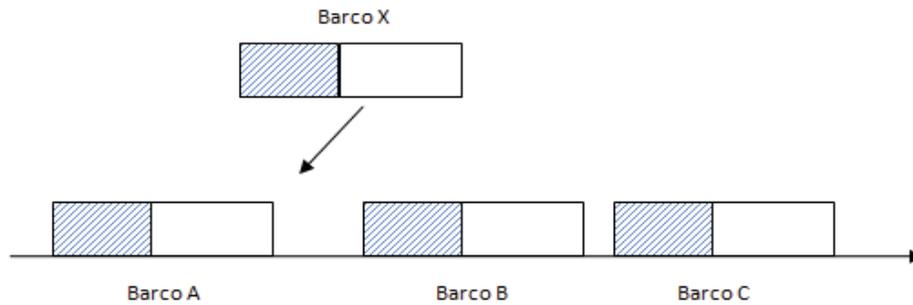
**Figura 4.9 Ventana de Tiempo resultante para el BAP**

La fórmula utilizada para calcularla se detalla a continuación:

- EAT = Corresponde al tiempo que es ingresado por el usuario, indicando la hora que desea arribar en el muelle.
- MRT = Corresponde al tiempo máximo de procesamiento, obtenido de  $A + (B * DRT)$ , donde A y B son números correlativos que permiten mayor flexibilidad a las ventanas.
- LDT = EAT + MRT.
- NFact = LDT – DRT (o N, tiempo de procesamiento ingresado por el usuario).

Teniendo ya definidos todos los elementos involucrados en la inserción de un nuevo barco a un muelle, a continuación se explica en detalle el algoritmo utilizado como heurística de inserción para el BAP.

Cuando un barco, denotado como X, quiere atracar en el puerto, quiere decir que se está en presencia de un nuevo evento de llegada de dicho barco (X+), y junto a éste, un evento de ida del barco (X-), por lo que se realiza una evaluación para poder insertarlo dentro de un muelle; esta evaluación lo que busca principalmente es minimizar los costos asociados al tiempo de servicio solicitado por el barco y cuando encuentra el muelle que menos costos otorga respecto al servicio, se asigna y se inserta en él. La Figura 4.10, demuestra gráficamente la llegada de un barco X, el cual desea insertarse entre los barcos A y B:



**Figura 4.10 Llegada de un Barco a un Muelle**

Para poder insertar efectivamente dicho barco, se evalúa lo siguiente:

- Se evalúan los muelles que estén disponibles y se escoge el que cumpla con las especificaciones de la petición y tomando en cuenta el que menos costos tenga asignado dentro de ellos, ya que este algoritmo de inserción se aplica en cada uno de los puntos de atraque que existen en el puerto.
- Se busca en el bloque de planificación asociado al muelle escogido, las secuencias que se tienen programadas, si no existe ninguna, quiere decir que la secuencia está vacía, por lo que se inserta el nuevo barco en ella, siendo este el primero dentro de ella.
- Si la secuencia no se encuentra vacía, se deben comparar las ventanas de tiempo factible entre los barcos que se encuentren al lado del barco que se desea insertar (barco A y barco B), y el barco X, para ello se debe buscar la factibilidad de la posible inserción; el procedimiento consiste en asumir que el barco X es insertado, por lo que se escoge la ventana de tiempo factible del evento de llegada y la del barco A son trasladadas (desplazadas) hacia ella para ver si se intersectan y comprobar la factibilidad de la inserción, este desplazamiento está dado por el tiempo necesario para el procesamiento (tiempo de servicio) del evento que se encuentra junto al evento del barco X, más el tiempo de slack asociado al evento X, lo mismo sucede con la ventana de tiempo del barco B.

## Capítulo 5: Diseño del Sistema

En este capítulo, se presentan todos los modelos que se consideraron como necesarios para la implementación del sistema, detallando mediante ellos la propuesta de la nueva arquitectura basada en MADARP, MABAP. Dichos modelos deben soportar la heurística propuesta en el capítulo anterior, el dinamismo del problema en las re planificaciones en tiempo de ejecución y también la nueva capa de monitoreo implementada.

Es importante señalar que los modelos fueron realizados a través de PASSI, intentando captar todas las funcionalidades del sistema en general y también las de los agentes participantes en él.

El hecho de tratarse de una continuación, implicó que se tuviese que comprender el diseño propuesto por [28]; sin embargo, se decidió realizar nuevamente todos los modelos, no siendo guiados por su idea, sino que con un aporte adicional, brindando una idea propia de diseño, adquiriendo con ello el dinamismo del sistema.

### 5.1 Modelo de Requerimientos del Sistema

Éste es el primero de los modelos otorgados en PASSI, cuyo objetivo es determinar los requerimientos del sistema, para lo cual se definen las funcionalidades de cada uno de los agentes involucrados en él. Para determinar dichas funcionalidades, se utiliza una notación en UML, lo que permite tener similitudes entre los diagramas de casos de uso, diagramas de secuencia y diagramas de actividad, y los modelos utilizados para identificar funcionalidades, roles y tareas, respectivamente.

Los diagramas propuestos por PASSI para el modelo de requerimientos del sistema se encuentran: Diagrama de Descripción de Dominio, Diagrama de Identificación de Agentes, Diagrama de Identificación de Roles y Diagrama de Identificación de Tareas; sin embargo, se va a modelar en base a los tres últimos, que son los modelos que se utilizaron para el diseño del sistema y su comprensión.

#### 5.1.1 Diagrama de Identificación de Agentes

Este diagrama es uno de los más importantes, está orientado principalmente a identificar y determinar las funcionalidades que posee cada uno de los agentes, ya que permite visualizar el sistema completo, incluidos los agentes que lo componen y además es aquel diagrama que sirve de base para la implementación de los demás modelos. Si por cualquier razón se modifican los diagramas posteriores, implica modificar el diagrama de identificación de agentes según los cambios que se hayan hecho.

Además, es en este diagrama en el que se permite visualizar, a parte de las funcionalidades, la interacción y comunicación que tienen los agentes, denotados por “communicate” si la comunicación es con otros agentes y “exclude” o “include” si se refiere a él mismo en el manejo de funciones.

Los agentes son representados dentro de un paquete, y en el cual define las funcionalidades de los agentes como macro tareas, ya que cada una de ellas puede desprender algunas más específicas.

A continuación, en la Figura 5.1, se muestra el diagrama de identificación de agentes, en la cual se incluyen los siguientes agentes: Vessel Agent (Agente Barco), Berth Agent (Agente Muelle), Vessel Simulator Agent (Agente Simulador Barco), Berth Simulator Agent (Agente Simulador Muelle), Berth Request Agent (Agente Petición Atraque), Berth Planer Agent (Agente Programador Atraque) y Planner Agent (Agente planificador). Todos ellos, junto a las funcionalidades e interacciones que poseen, explicadas en la arquitectura propuesta.

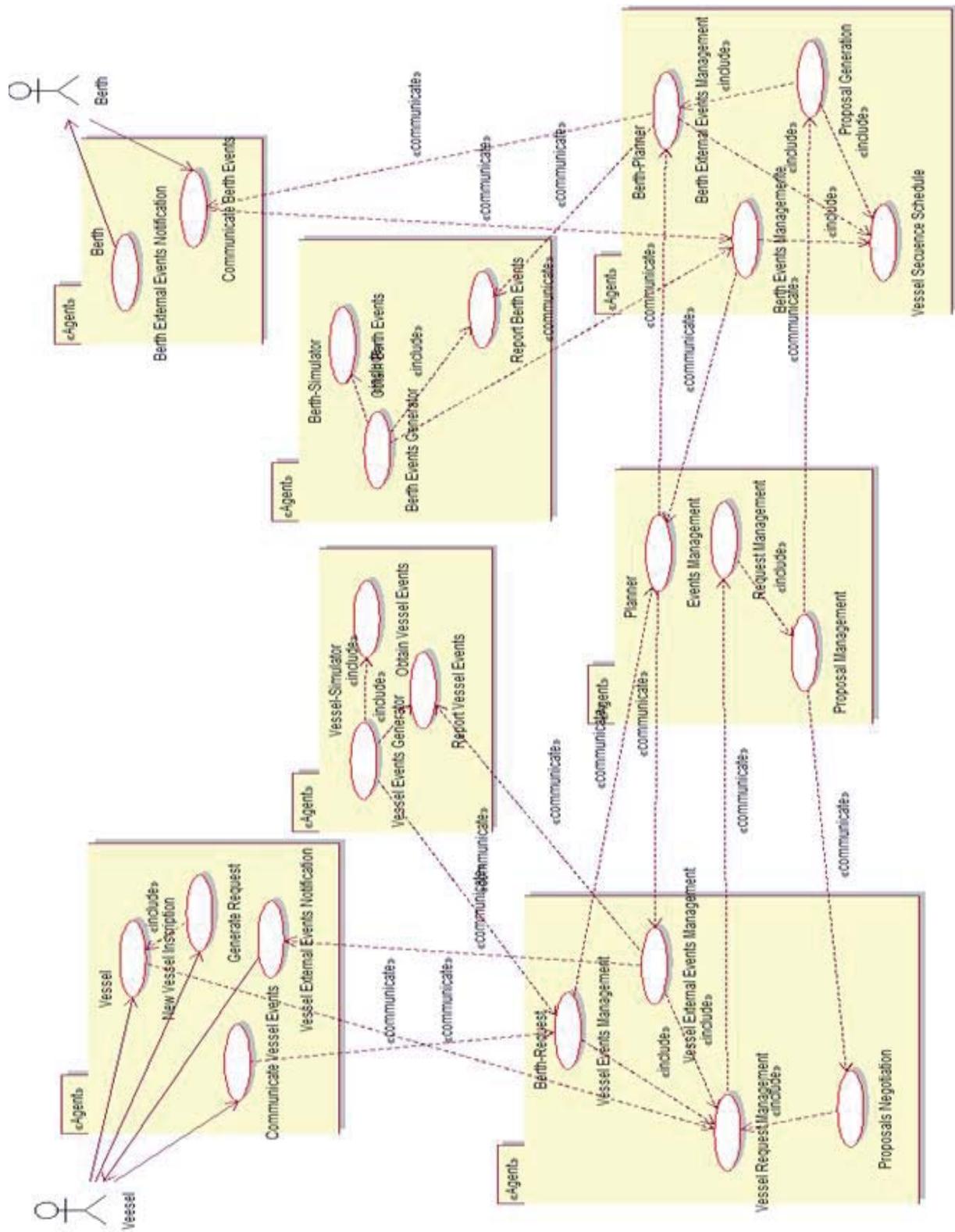


Figura 5.1 Diagrama de identificación de agentes

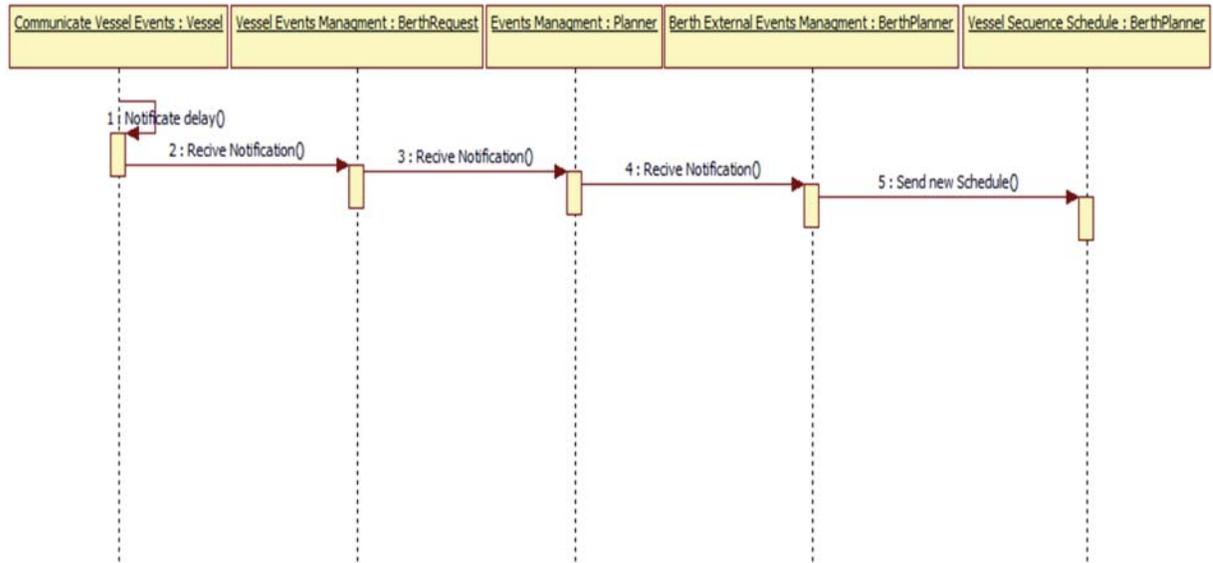
### **5.1.2 Diagrama de Identificación de Roles**

En este diagrama se especifica detalladamente el comportamiento que tiene un agente y explora todas las posibilidades dentro del diagrama de identificación de agentes, para efectuar un escenario importante.

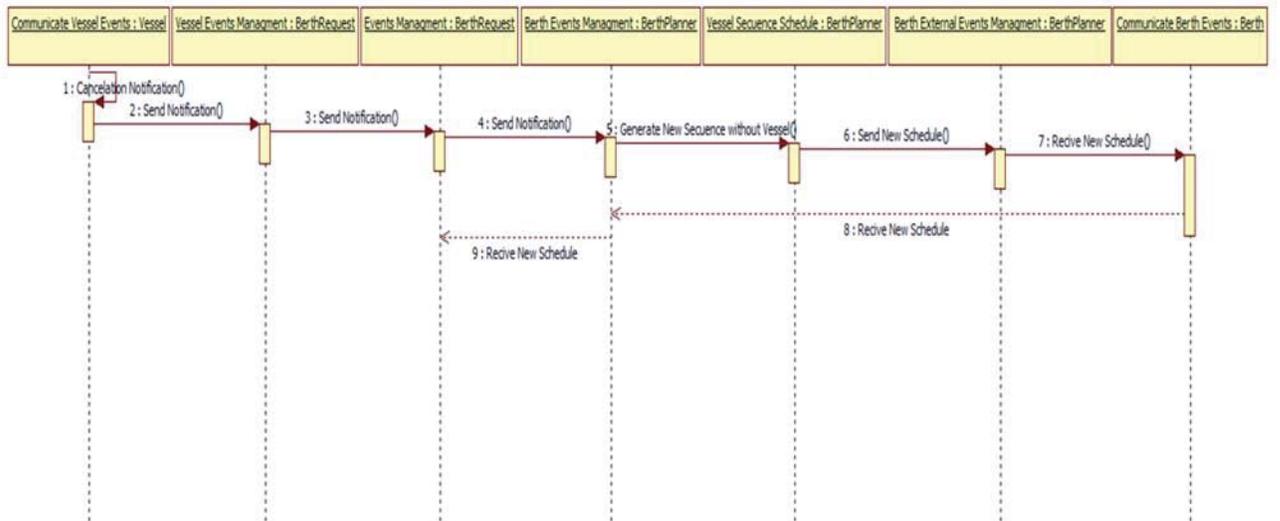
Lo que describe dicho diagrama es el escenario en que se encuentran interactuando y comunicándose los agentes, y cada uno de ellos puede participar en más de un escenario a la vez y también puede tener más de un rol. Su creación consiste en la producción de varios diagramas de secuencia que especifican los escenarios, pero con ciertas variaciones, como lo es el nombrar el rol.

A continuación, se muestran algunos diagramas de identificación de roles, correspondientes a los escenarios de: asignación de una petición exitosa (Figura 5.2), la cancelación de una notificación (Figura 5.3) y la re planificación por atraso (Figura 5.4).





**Figura 5.3 Escenario de cancelación de una notificación**



**Figura 5.4 Escenario de una re planificación por atraso.**

### 5.1.3 Diagrama de Identificación de Tareas

Este corresponde al último de los diagramas de modelo de requerimientos del sistema de PASSI. En este paso se va analizando un agente a la vez, representando cada uno mediante un diagrama de actividad compuesto por dos sectores o calles, en el sector de la derecha se encuentran las tareas del agente, y en el izquierdo, se encuentran los agentes con los cuales se interactúa. En este diagrama un nodo de actividad representa una tarea que el agente debe realizar. De manera que, este diagrama resume el comportamiento de cada agente, así como, sus interacciones con otros.

A continuación se presentan cuatro diagramas de identificación de tareas, entre los que se encuentran: especificación de tareas del agente muelle (Figura 5.5), especificación de tareas del agente barco (Figura 5.6), especificación de tareas del agente simulador muelle (Figura 5.7), especificación de tareas del agente simulador barco (Figura 5.8).

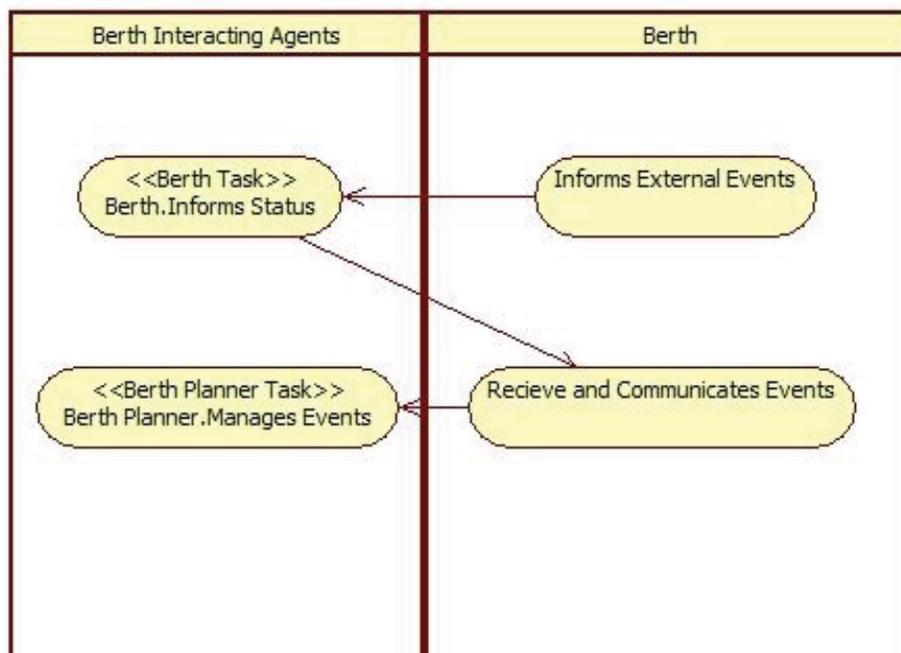


Figura 5.5 Especificación de tareas del agente berth

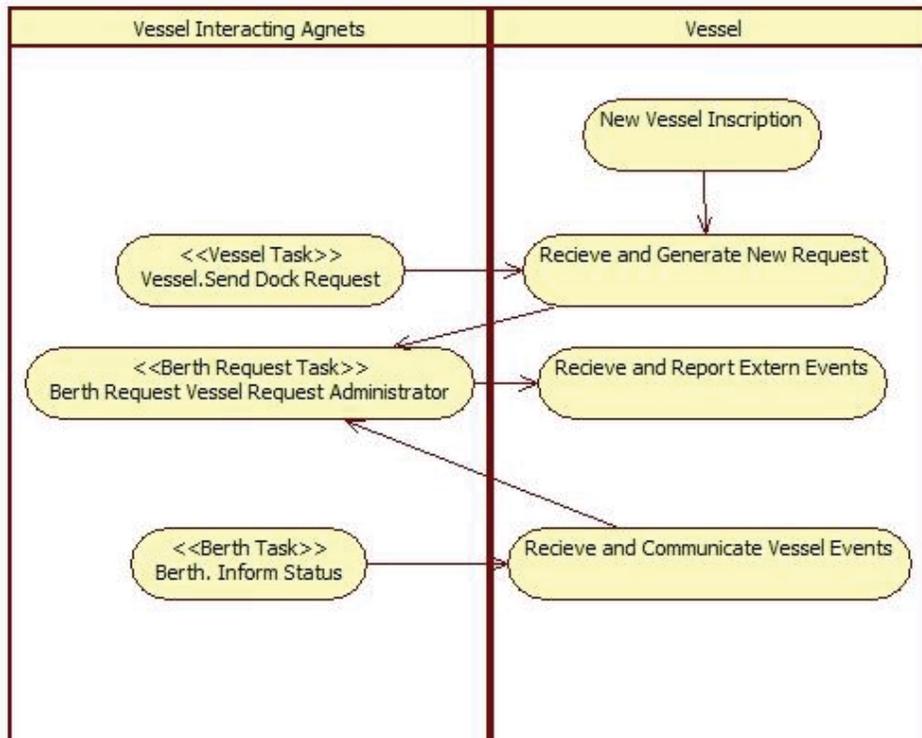


Figura 5.6 Especificación de tareas del agente barco

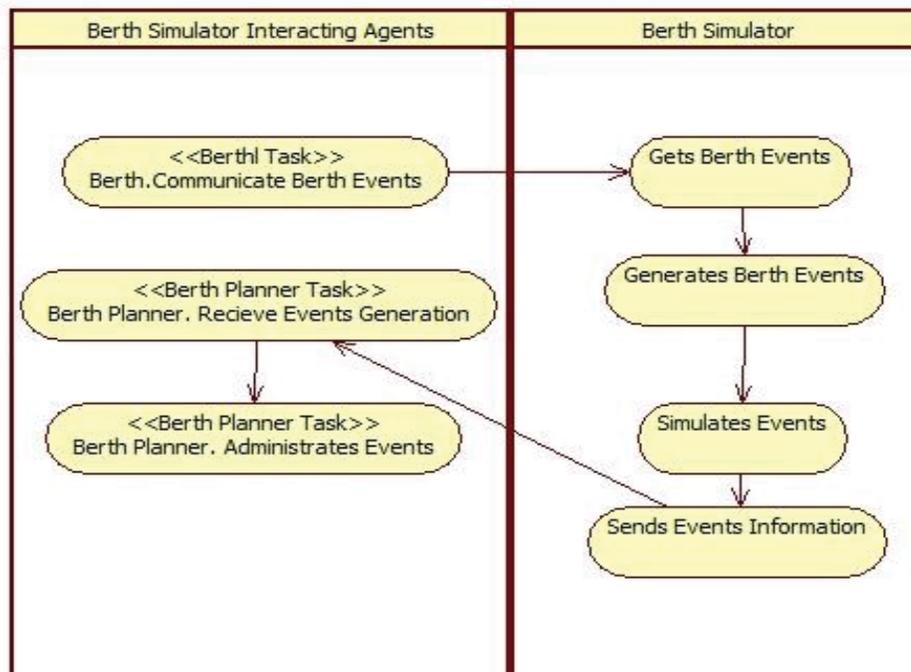


Figura 5.7 Especificación de tareas del agente simulador berth

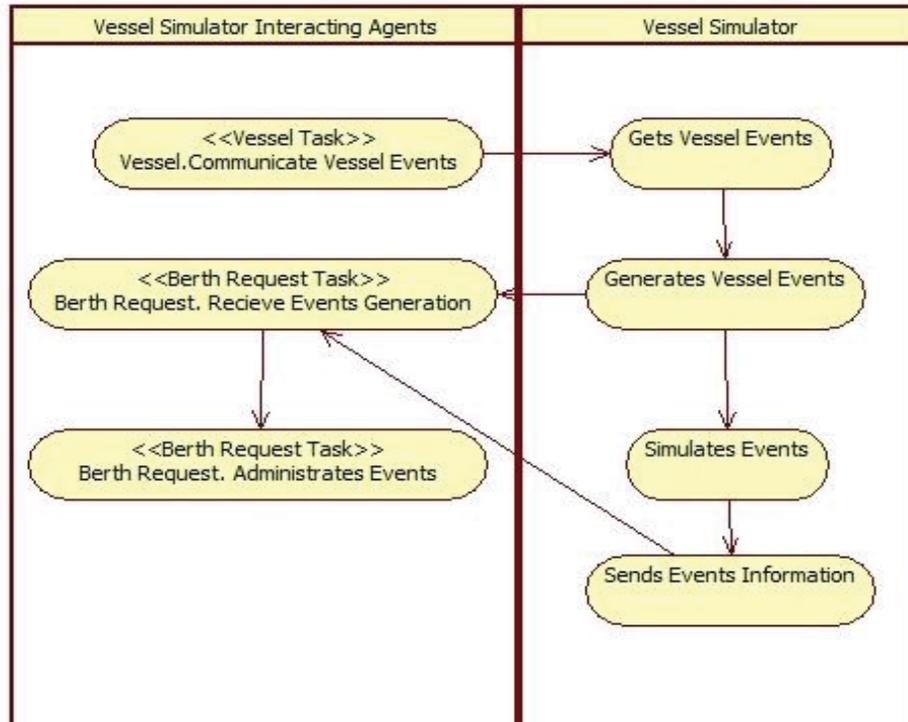


Figura 5.8 Especificación de tareas del agente simulador barco

## **5.2 Modelo de Sociedad de Agentes**

Pertenece al segundo de los modelos propuestos en PASSI, en donde se pueden diferenciar diversos diagramas que facilitan la comprensión del sistema, entre ellos se encuentran: descripción de ontología de dominio, descripción de ontología de comunicación, descripción de roles y descripción de protocolos.

### **5.2.1 Diagrama de Descripción de Ontología de Dominio**

La composición de este diagrama es mediante un diagrama de clases, en el cual se pueden especificar los conceptos, predicados y acciones que pudiesen existir en el dominio del problema. Éste permite que se pueda definir un vocabulario común entre los diferentes agentes para que se puedan comunicar. En la Figura 5.9, se muestra el diagrama de ontología de dominio que se ha propuesto, en éste se diferencian sólo los conceptos que se utilizan.

### **5.2.2 Diagrama de Descripción de Ontología de Comunicación**

El diagrama de ontología de la comunicación también se representa mediante un diagrama de clases. La característica principal de éste es que se compone de dos elementos, que son los agentes y las comunicaciones que ellos tienen. Esto se debe a que entre dos agentes, existe una relación, dicha relación es representada a través de la comunicación. En la Figura 5.10 se muestra el diagrama de dicha ontología en la cual se distinguen algunas de las comunicaciones más relevantes sucedidas en la interacción de los agentes.

### **5.2.3 Diagrama de Descripción de Roles**

Este diagrama también se compone mediante diagrama de clases, en donde los roles de los agentes se representan como clases, las cuales se agrupan en cajas para representar a un agente y cada rol puede tener involucradas varias tareas. Un agente puede tener uno o varios roles, y a su vez, éstos se pueden ir cambiando en sus determinadas tareas. A continuación se puede apreciar en la Figura 5.11 un diagrama de roles definido para el BAP.

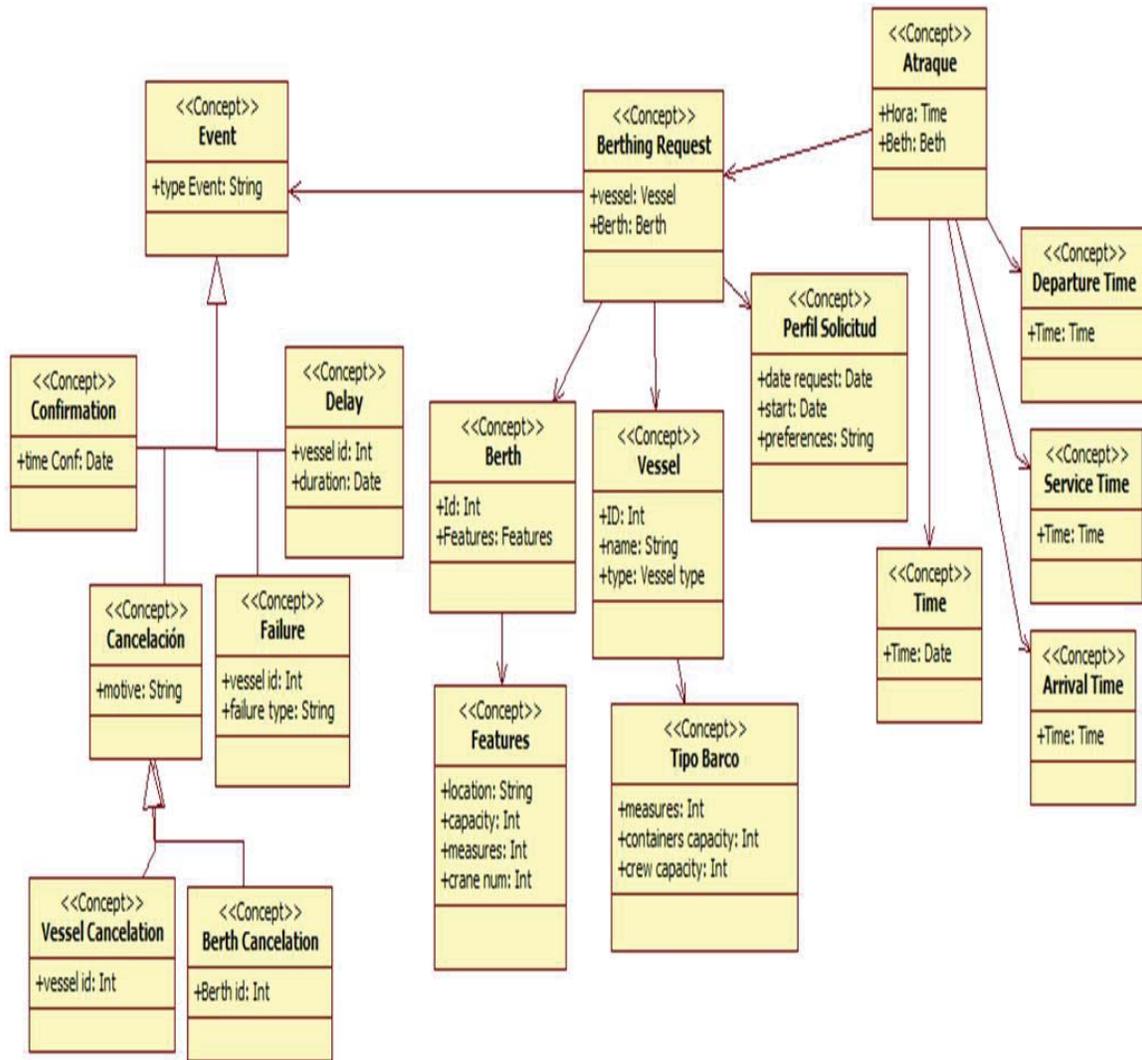


Figura 5.9 Descripción de ontología de dominio

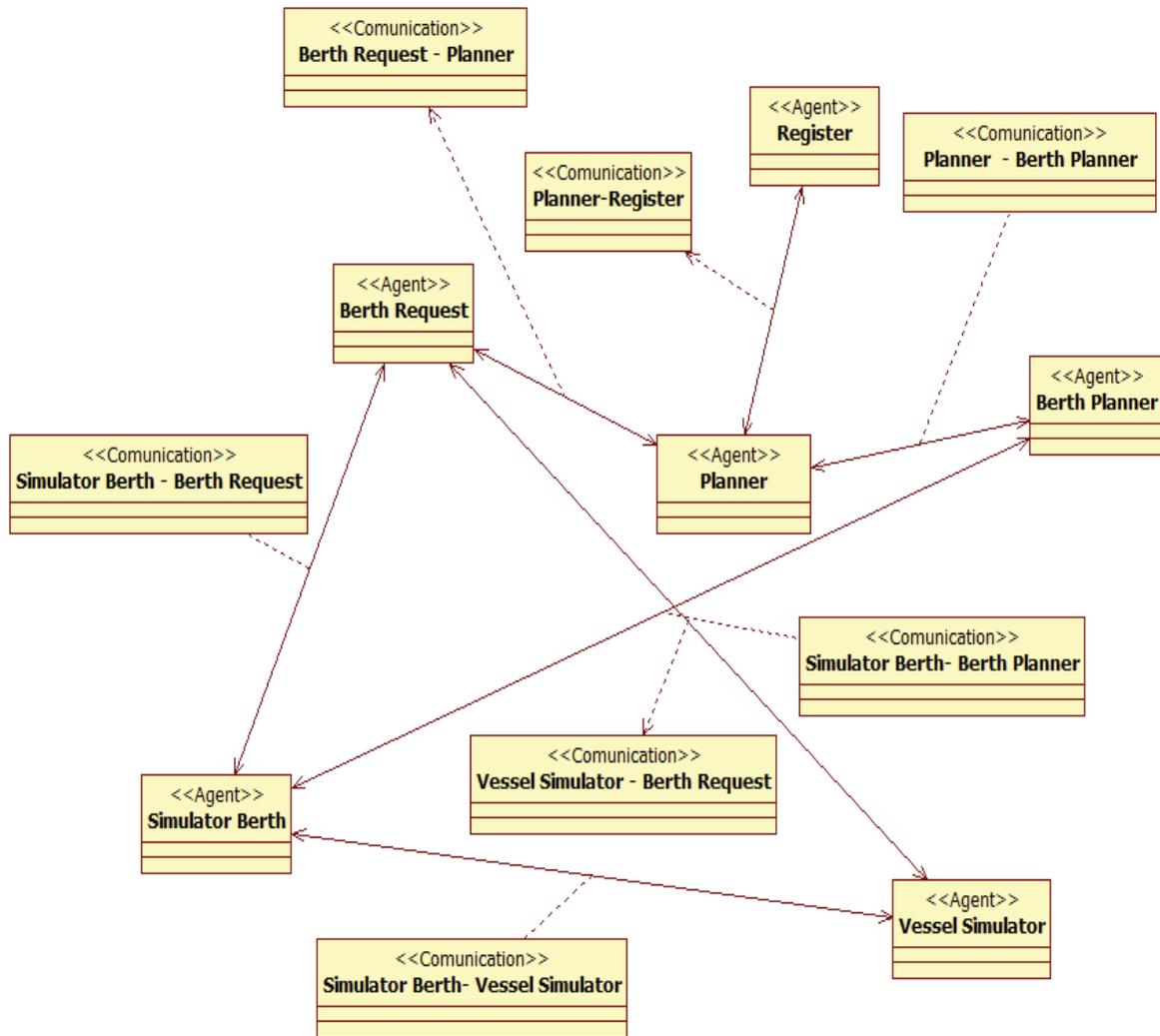


Figura 5.10 Descripción de ontología de dominio

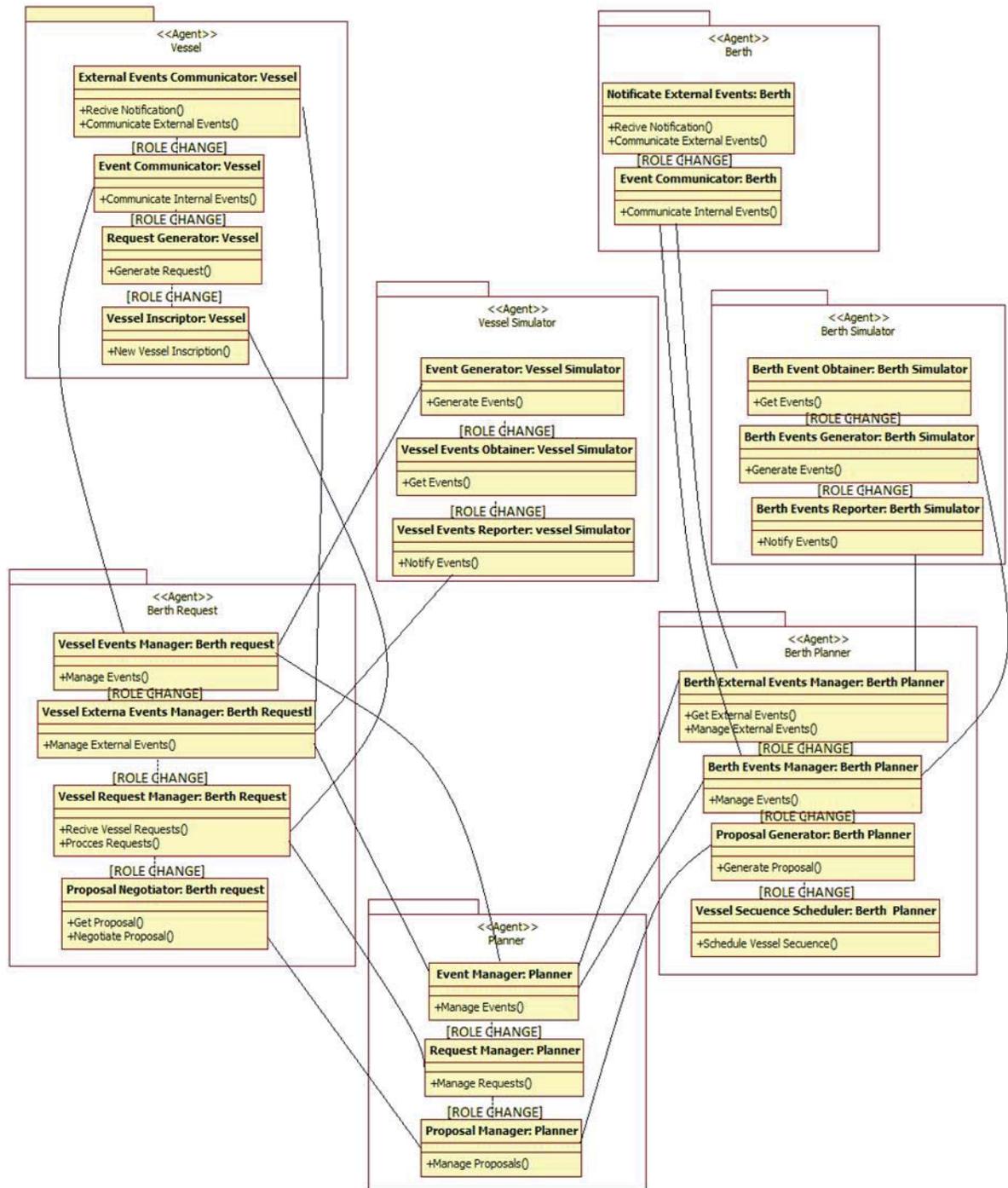


Figura 5.11 Descripción de roles

## **5.3 Modelo de Sociedad de Agentes**

Este modelo corresponde al penúltimo de los modelos de PASSI; sin embargo corresponde al último de los modelos realizados y presentados en este informe. Se incorporan en él, el diagrama de definición de la estructura multiagente y el diagrama de descripción de la estructura de los agentes individualmente.

### **5.3.1 Diagrama de Definición de la Estructura Multiagente**

Acá se representa todo el sistema multiagente, mediante un diagrama de clases. Cada una de las clases incorporadas simbolizan a los agentes, en la parte de los atributos, se agregan los conocimientos del agente y en la parte de las operaciones de la clase, se agregan todas las tareas que son designadas para cada agente. En la Figura 5.12, se puede apreciar el diagrama de la estructura multiagente diseñado para el BAP.

### **5.3.2 Diagrama de Definición de la Estructura del Agente Individual**

En esta etapa de creación de diagramas, se debe crear un diagrama de clases para cada agente, describiendo en él su estructura interna y un diagrama de clases describiendo todas las tareas que le fueron asignadas (al agente creado con anterioridad en el diagrama de clases). A continuación, se pueden apreciar aquellos diagramas que se consideraron más relevantes de la estructura de agente individual, en la Figura 5.13, está representado el diagrama de la estructura de agente individual para el Planner Agent

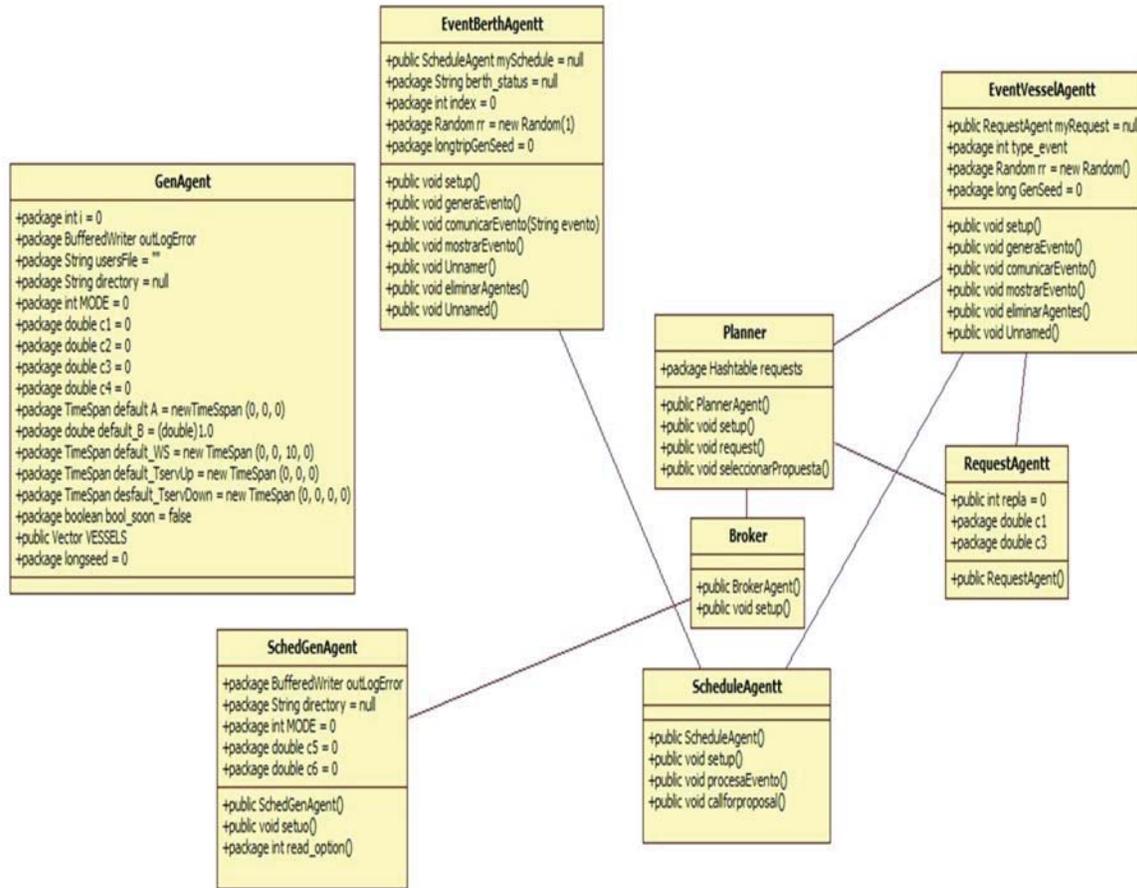


Figura 5.12 Diagrama de estructura multiagente

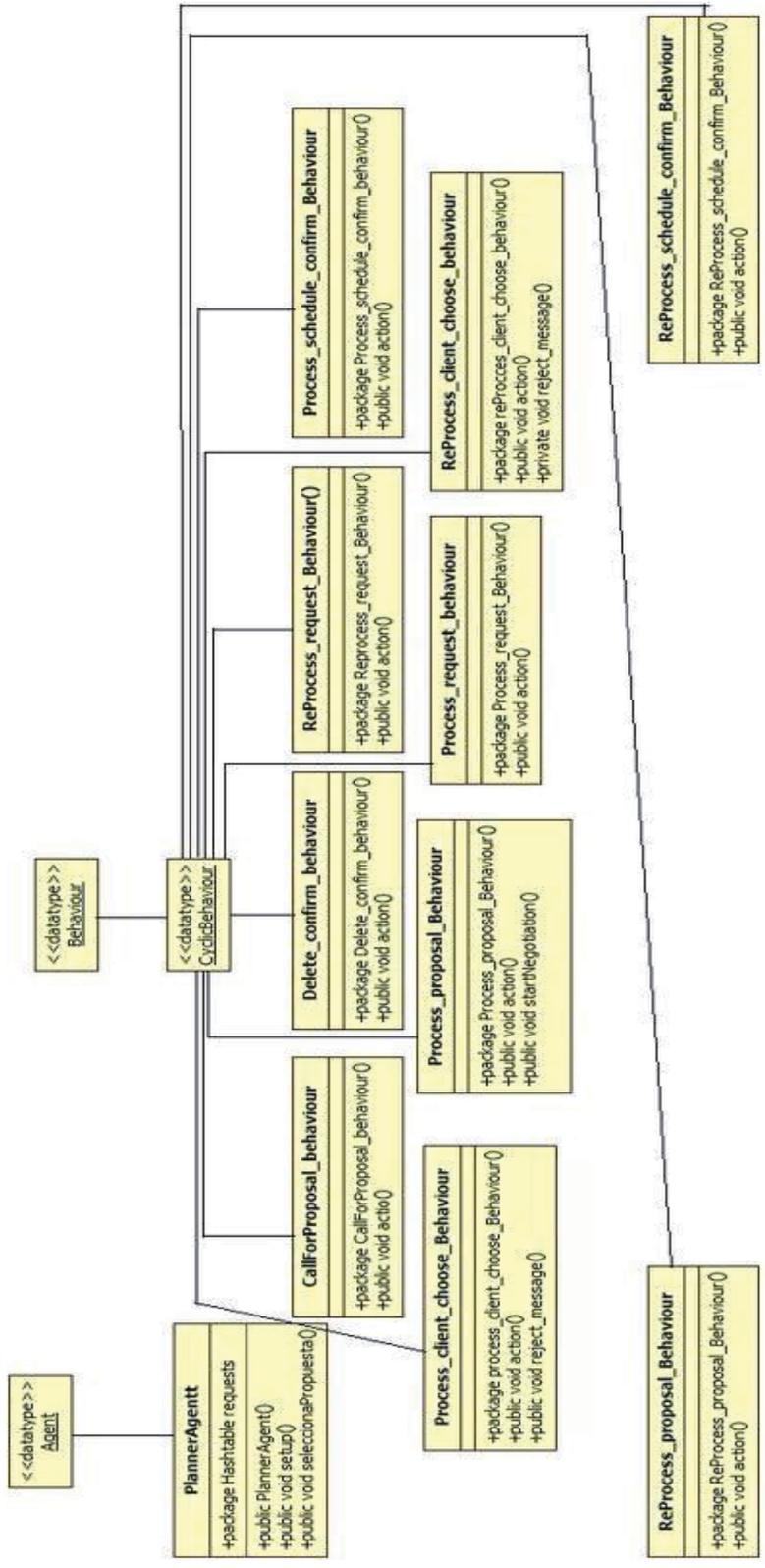


Figura 5.13 Diagrama de estructura individual agente planner

## Capítulo 6 Diseño de Pruebas

En todos los sistemas la parte de las pruebas es muy relevante, ya que gracias a ellas se puede verificar que el software cumple o no con las especificaciones y funcionalidades requeridas y con la integración de componentes.

Son muy variados los tipos de pruebas que se pueden ejecutar, entre ellas se encuentran las pruebas de usabilidad, de rendimiento, de funcionalidad, de seguridad, entre otras. Para este sistema en particular, se orienta básicamente al análisis y diseño de pruebas funcionales, comprobando con ellas el cumplimiento de las especificaciones planteadas.

Dentro de las metodologías de las pruebas de software, se pueden diferenciar las etapas de planificación, diseño y ejecución de las pruebas. A continuación, se muestran las etapas efectuadas para este informe.

### 6.1 Planificación de la Pruebas

Es necesario saber qué es lo que se probará en el sistema y en qué profundidad, para así poder ejecutar la siguiente etapa con mayor claridad. Lo primero que hay que tomar en cuenta es la planificación del proyecto, para ir separándolas por módulos y poder probarlos, para luego integrarlos en el sistema.

A continuación, se detallan las funcionalidades necesarias para que el software funcione correctamente:

- Extracción y reutilización de código.
- Planificación de solicitudes estáticas.
- Planificación de solicitudes dinámicas, mediante las eventualidades de los muelles y los barcos, cada una por separado y luego su integración.
- Coordinación del factor tiempo entre los componentes.
- Monitoreo efectivo del sistema.

Para ello, se harán las pruebas de caja blanca y de caja negra. Las primeras nos sirven para ver la lógica del programa (estructura interna); mientras que las otras nos permiten ver la consistencia de la información de entrada y salida (mediante la interfaz).

## 6.2 Diseño de las Pruebas

Acá, es donde se deben efectuar los casos de pruebas funcionales que se le realizarán al sistema para ver su validez. Se debe mostrar que es lo que se ingresará para ver si efectivamente funciona como corresponde.

- **Extracción y reutilización de código**

Se sacará el solver del algoritmo de inserción y se probará su funcionalidad, además de la reutilización de clases y archivos necesarios para la generación del código por completo.

- **Planificación de solicitudes estáticas:**

Se ingresarán los berths al sistema y varias solicitudes de atraque de barcos. Primeramente, se verificará que se realizan las asignaciones según corresponde a la heurística propuesta, es decir, que los eventos de llegada e ida del barco estén siempre juntos. Luego, se verifica la planificación de la secuencia de barcos que tendrá cada berth, según las solicitudes efectuadas y evaluaciones en que las peticiones superen la cantidad de berths disponibles y que no los superen.

- **Planificación de solicitudes dinámicas**

Para que el sistema funcione de manera dinámica, se realizarán pruebas de las eventualidades del berth y de los barcos por separado, y luego se verificará la integración de todas ellas en el sistema. Dichos eventos serán seleccionados de manera aleatoria y se probará el sistema para ver la respuesta de éste ante estas eventualidades.

- **Coordinación del factor tiempo entre los componentes**

El tiempo debe funcionar cronológicamente ante situaciones, por lo que se ingresarán tiempos erróneos para verificar que se esté cumpliendo.

- **Monitoreo efectivo del sistema**

Es la última prueba y se verifica cuando todos los escenarios mencionados anteriormente funcionen correctamente y se integren por completo.

## 6.3 Datos de Entrada

Ya definida la planificación y el diseño de pruebas, es necesario determinar los datos y el formato de los datos que se ingresarán al sistema y que servirán para llevar a cabo las pruebas.

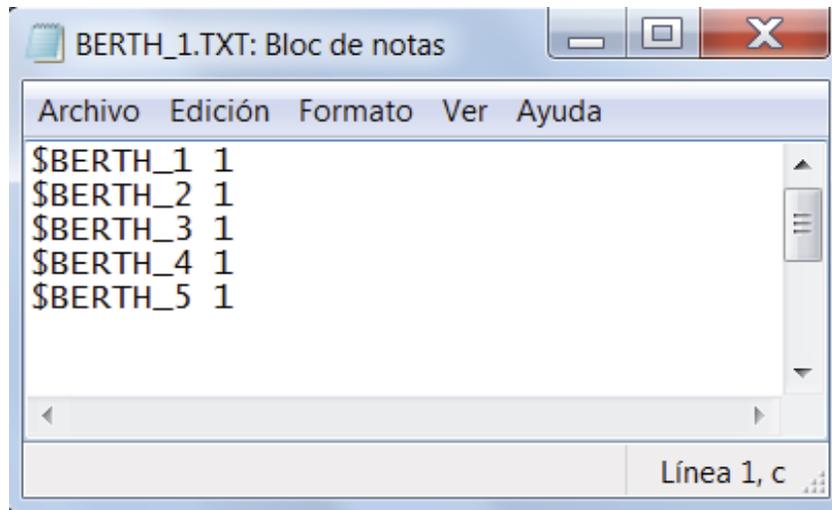
Se utilizarán tres archivos de entrada para la ejecución del programa, en el cual se cargarán automáticamente según se llamen. El primero corresponde a un .ini, denominado MABAP.ini, en el cual están integrados los datos necesarios para comenzar con la ejecución del software, allí se indican e inicializan valores de variables y archivos de texto que se ocuparán. Dentro de estos archivos de texto, se encuentra uno con el formato de los datos ingresados para un muelle, mientras que el segundo contiene el formato de los barcos.

Un ejemplo de archivo de entrada de un barco es el que se muestra a continuación (Figura 6.1), se puede ver en la primera línea de izquierda a derecha, que los datos guardados son: un identificador general de barco (V), un nombre o identificador propio para cada barco (VESSEL\_1), la hora de llegada del barco (11:42:51), el tiempo estimado de procesamiento que tendrá el barco, definido como milisegundos (503450), y por último un número que puede ser 1 o 0, que indican si se desea llegar o irse a la hora determinada, respectivamente.

Identificador	Nombre	Hora de llegada	Tiempo estimado (ms)	Estado
V	VESSEL_1	11:42:51	503450	1
V	VESSEL_2	11:49:47	258900	1
V	VESSEL_3	11:25:20	200000	1
V	VESSEL_4	11:36:49	398000	1
V	VESSEL_5	11:32:52	267000	1
V	VESSEL_6	11:40:03	123000	1
V	VESSEL_7	11:52:45	230000	1
V	VESSEL_8	11:28:48	330000	1
V	VESSEL_9	11:35:38	276000	1
V	VESSEL_10	11:19:58	301000	1
V	VESSEL_11	11:55:04	723000	1
V	VESSEL_12	11:25:44	222000	1
V	VESSEL_13	11:40:13	498000	1
V	VESSEL_14	11:03:35	487000	1
V	VESSEL_15	11:44:58	655000	1
V	VESSEL_16	11:14:18	234000	1
V	VESSEL_17	11:20:59	623000	1
V	VESSEL_18	11:39:56	845000	1
V	VESSEL_19	11:13:22	200000	1
V	VESSEL_20	11:09:00	220000	1
V	VESSEL_21	11:34:56	700000	1

**Figura 6.1** Archivo de datos de entrada para un barco

El otro archivo de entrada corresponde al muelle, un ejemplo de este se presenta a continuación (Figura 6.2), y lo único que contiene es un nombre o identificador del muelle (\$BERTH\_1), acompañado de el número 1, que indica la cantidad de barcos que pueden ser atendidos, o sea, un muelle puede atender un solo barco a la vez; y luego hacia abajo la cantidad de ellos que hay en el puerto, en este caso son cinco muelles.



**Figura 6.2** Archivo de datos de entrada para un muelle.

## Capítulo 7 Implementación de Pruebas

Como ya se mencionó en el capítulo anterior, se ha realizado ya una planificación y diseño de pruebas, por ende, falta la implementación de ellas, mediante la ejecución del código fuente. A continuación, se da un detalle de cómo exactamente se ejecutaron dichas pruebas y los resultados obtenidos a partir de los archivos de entrada determinados, obteniendo una visión más generalizada acerca del funcionamiento del sistema.

### 7.1 Pruebas Estáticas

Las pruebas estáticas realizadas se pensaron principalmente en base a la heurística propuesta, y comprobando que se cumple con la correcta asignación de los barcos, a la utilización de muelles y a los costos asociados a ellos.

**Tabla 7.1 Ejemplo de asignación de barcos a muelles en sus eventos de arrival y departure**

Eventos	Berth 1	Berth 2	Berth 3	Berth 4	Berth 5
	Start	Start	Start	Start	Start
Arrival	V-VESSEL_25	V-VESSEL_26	V-VESSEL_30	V-VESSEL_23	V-VESSEL_29
Departure	V-VESSEL_25	V-VESSEL_26	V-VESSEL_30	V-VESSEL_23	V-VESSEL_29
Arrival	V-VESSEL_16	V-VESSEL_3	V-VESSEL_28	V-VESSEL_24	V-VESSEL_20
Departure	V-VESSEL_16	V-VESSEL_3	V-VESSEL_28	V-VESSEL_24	V-VESSEL_20
Arrival	V-VESSEL_10	V-VESSEL_12	V-VESSEL_9	V-VESSEL_17	V-VESSEL_14
Departure	V-VESSEL_10	V-VESSEL_12	V-VESSEL_9	V-VESSEL_17	V-VESSEL_14
Arrival	V-VESSEL_22	V-VESSEL_21	V-VESSEL_1	V-VESSEL_4	V-VESSEL_19
Departure	V-VESSEL_22	V-VESSEL_21	V-VESSEL_1	V-VESSEL_4	V-VESSEL_19
Arrival		V-VESSEL_27	V-VESSEL_15	V-VESSEL_18	V-VESSEL_8
Departure		V-VESSEL_27	V-VESSEL_15	V-VESSEL_18	V-VESSEL_8
Arrival				V-VESSEL_11	V-VESSEL_5
Departure				V-VESSEL_11	V-VESSEL_5
Arrival					V-VESSEL_6
Departure					V-VESSEL_6
Arrival					V-VESSEL_13
Departure					V-VESSEL_13

<b>Arrival</b>					V-VESSEL_2
<b>Departure</b>					V-VESSEL_2
<b>Arrival</b>					V-VESSEL_7
<b>Departure</b>					V-VESSEL_7
	Stop	Stop	Stop	Stop	Stop

En la tabla anterior se muestra un ejemplo del resultado de la primera prueba realizada, visualizándose la secuencia de barcos que se asignaron a cada uno de los muelles existentes (5 muelles en este ejemplo); se puede comprobar que la heurística funciona correctamente, cumpliendo con la restricción de que se atiende un barco a la vez por muelle y luego se sigue con la siguiente petición, y que los eventos de llegada e ida permanecen siempre juntos.

A continuación, se muestran los gráficos asociados a los resultados experimentales de los diferentes escenarios planteados. Cada uno de ellos fue corrido 5 veces, por lo que los resultados corresponden al promedio de cada una de las corridas, a partir de 6 archivos de entrada de peticiones de barcos diferentes, que contenían 30 solicitudes de atraque cada uno, y con 3 archivos de muelles, con 4, 5 y 7 berths.

El primero, representa el promedio de la cantidad de barcos que fueron atendidos (de un total de 30), en donde se detalla que cuando hay 7 berths disponibles, todos son atendidos, mientras que en los otros dos escenarios no, ya que la cantidad de berths no satisface a todas las solicitudes generadas debido a que el tiempo de la petición de atraque y su tiempo de proceso no es compatible con la secuencia planificada.

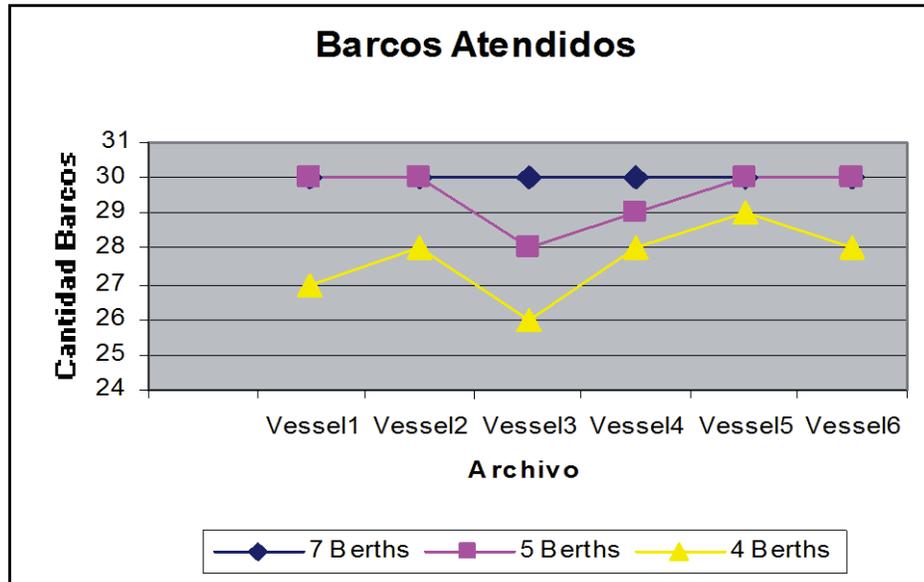


Figura 7.1 Gráfico de cantidad de barcos atendidos

El siguiente gráfico, muestra la cantidad de muelles que son usados, en los cuales se planifica la secuencia de barcos a atracar. Se puede verificar que el resultado varía, ya que en los escenarios en los que hay un total de 4 y 5 berths, todos son utilizados, mientras que en el que hay 7 berths, no necesariamente son todos los que se ocupan, ya que pueden ser atendidos todos los barcos en una cantidad menor de muelles que los existentes.

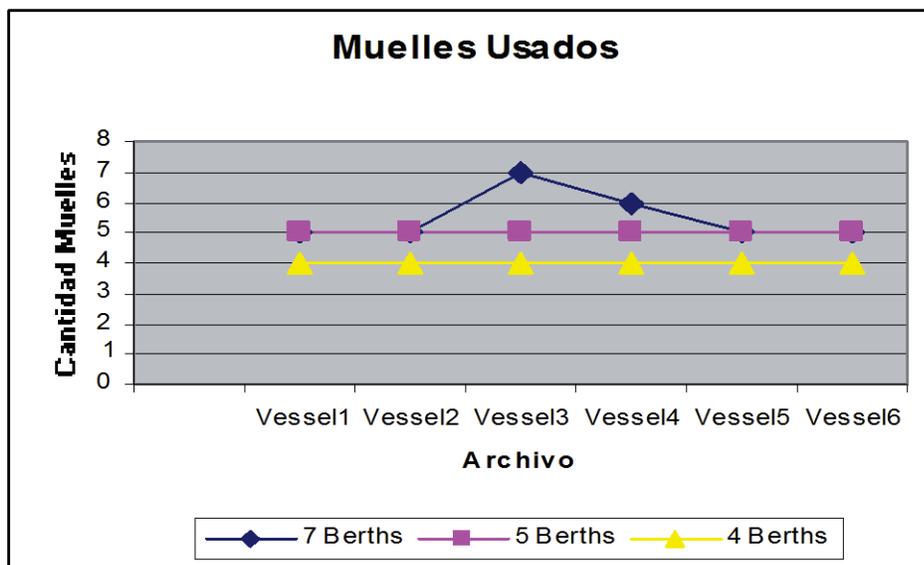


Figura 7.2 Gráfico cantidad de muelles usados

En el último gráfico, se puede ver que los costos totales oscilan entre 200 y 500, y varían dependiendo de varios factores, como por ejemplo los slacks, el tiempo de procesamiento, la cantidad de barcos atendidos y de muelles utilizados.

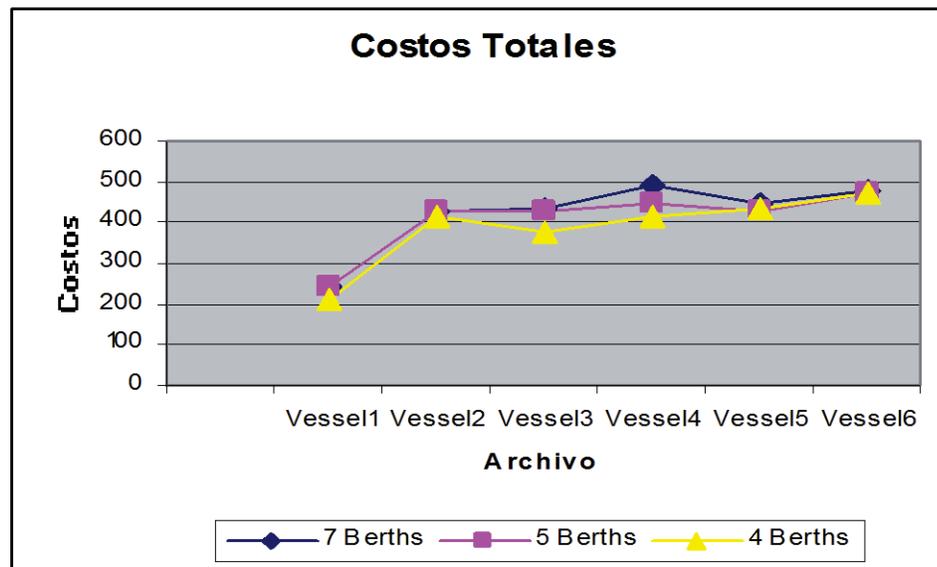


Figura 7.3 Gráfico resultados de los costos totales

## 7.2 Pruebas Dinámicas

Para la ejecución de este tipo de pruebas, se siguió con las características del escenario de pruebas estáticas, en donde la secuencia de barcos en un determinado muelle ya ha sido planificada, y posterior a ello, se ingresan eventualidades.

Las pruebas dinámicas se hicieron en base a sucesión de eventos por parte de los barcos y muelles, cada uno por separado y luego la integración de ellos. La manera de seleccionar las diferentes eventualidades a ocurrir en el transcurso del tiempo, se realiza de manera aleatoria.

Es así como se inició la ejecución del sistema tratando a las eventualidades de desperfectos del muelle, cancelaciones por parte del cliente y cambios de tiempo en las peticiones iniciales por separado, por lo tanto, cada una de ellas necesitó de una replanificación de la secuencia de barcos asignados a un muelle en los tiempos que se estipularon, utilizando para ello las ventanas de tiempo ya mencionadas.

### 7.2.1 Evento de Desperfecto en el Muelle

Este evento corresponde a cuando en algún berth ocurre algún tipo de desperfecto, por lo que no puede ser utilizado, por ende, la planificación de la secuencia de barcos que existe en él, debe ser replanificada y asignada a otro berth que esté en perfecto funcionamiento.

La siguiente tabla, muestra un resumen de cuatro corridas diferentes que se realizaron en un mismo archivo de datos de entrada de barcos (archivo Vessel 4), y con 7 muelles disponibles en su totalidad para atender a los barcos. Se puede apreciar que los barcos atendidos varía mucho, esto depende principalmente a que la cantidad de muelles utilizados también varía, debido a que algunos de ellos quedaron inutilizables por desperfectos; es por ello, que al hacer la replanificación de los barcos que dependían del berth con desperfecto, sus peticiones no necesariamente pudieron ser atendidas con total satisfacción para ellos, no pudiendo ser asignadas a ninguna secuencia disponible. En cuanto a los costos también varían mucho, al igual que en la prueba estática, por factores como cantidades de barcos atendidos, muelles utilizados, slacks generados. Finalmente, se concluye que los resultados son diferentes, debido al dinamismo de la cantidad de muelles disponibles para su uso.

**Tabla 7.2 Resumen de datos de muelles con desperfecto**

Barcos Atendidos	Muelles Usados	Costo Total	Nº Muelles con Desperfecto	Muelles con Desperfecto
13	1	157.62	6	Berths 1,3,4,5 y 6
27	4	391.7	3	Berths 3,4 y 6
28	4	334.23	3	Berths 3,4 y 6
29	5	471.27	2	Berths 5 y 6

La siguiente tabla muestra un ejemplo concreto de lo que sucede en las secuencias de los barcos asociadas a cada uno de los berths disponibles desde un comienzo en la planificación inicial (estática), y cómo es que ésta fue cambiando debido a la sucesión de eventos de desperfecto del muelle. Los números remarcados con color rojo, indican los identificadores de barcos que tuvieron que ser reasignados, según los desperfectos indicados.

**Tabla 7.3 Ejemplo de secuencias de barcos luego de algún desperfecto**

Secuencia de Barcos					
Muelle	Planificación Inicial	Desperfecto Berth 6	Desperfecto Berth 3	Desperfecto Berth 4	Costo
Berth 1	1	26,1,17,28,10	26,1,17,28,10,5,3	26,1,17,28,10,5,3,4	57
Berth 2		18,16	18,16,7	18,16,7,14,20	121
Berth 3	27,5,7	27,5,7,3	27,5,7,3		
Berth 4	13,2,0,14,4,11	13,2,20,14,4,11	13,2,20,14,4,11		
Berth 5	8,24,22,21,12,29	8,24,22,21,12,29	8,24,22,21,12,29	8,24,22,21,12,29	22
Berth 6	25,26,18,17,28,16,10,3				
Berth 7	19,30,23,6,15,9	19,30,23,6,15,9	19,30,23,6,15,9	19,30,23,6,15,9,11	123
		25 No Atendido	27 No Atendido	13,2 No Atendido	

### 7.2.2 Evento de Cancelaciones de Barcos

Evento que sucede cuando un cliente decide no atracar en un determinado momento, por lo cual cancela su petición de asignación a un barco, y este caso queda nula, o sea, se debe eliminar el barco de la secuencia ya planificada. En este tipo de eventualidad, no es necesaria una replanificación de los barcos a otros berths, sino que sólo son eliminados.

A continuación, se muestra una tabla con el resumen de cuatro ejecuciones de código diferentes sobre el mismo archivo de entrada, en la cual se muestran algunos datos, entre los que se encuentran los datos asociados a la cantidad de cancelaciones de barcos y a los identificadores de los barcos que realizaron una cancelación de solicitud. Las conclusiones que se pueden obtener de dicha tabla, es que la cantidad de muelles usados es menor a la cantidad de muelles disponibles (7 en total), debido a que ahora la cantidad de barcos por atender es disminuido significativamente debido a las cancelaciones realizadas y al igual que en pruebas anteriores los costos varían según factores.

**Tabla 7.4 Resumen de barcos cancelados**

Barcos Atendidos	Muelles Usados	Costo Total	N° Cancelaciones	Barcos
				Cancelados
13	6	426,52	17	3,4,5,6,7,8,9,11,15,17,19,21,24,25,26,28y29
29	6	426,52	1	13
23	5	425,62	7	5,7,8,11,12,16 y 23
23	6	415,52	7	24,21,18,16,3,20 y 14

A modo de ejemplo, en la tabla presentada se ve la planificación inicial de la secuencia de barcos en un muelle determinado y cómo es el resultado, luego de las cancelaciones realizadas por los barcos (denotadas con color rojo), además del costo final obtenido por secuencia.

**Tabla 7.5 Ejemplo de secuencia de barcos luego de cancelaciones de barcos**

Secuencia de Barcos			
Muelle	Planificación Inicial	Cancelaciones	Costo
Berth 1	27,5,7	27,5,7	13
Berth 2	8,24,22,21,12	8,22,12	88
Berth 3			0
Berth 4	1	1	8
Berth 5	25,26,18,17,28,16,10,3	25,26,17,28,10	76
Berth 6	13,2,20,14,4,29	13,2,4,29	84
Berth 7	19,30,23,6,15,9,11	19,30,23,6,15,9,11	113

### 7.2.3 Evento de Cambios de Tiempo en los Barcos

Este evento permite realizar un cambio en la petición inicial de atraque, ya sea para adelantar o atrasar el arribo del barco; a pesar de ello, al momento de hacerv esta solicitud, la planificación inicial de la secuencia ya está hecha, por lo que se necesita hacer una replanificación del barco, eliminandolo de la secuencia de muelle a la que pertenecía y asignarla a una nueva, según las petición del tiempo que solicitó. La determinación del tiempo, si el barco desea adelantar o atrasar su tiempo de llegada, se realizó de manera aleatoria.

En la tabla presentada a continuación, se aprecia un resumen con los datos asociado a la cantidad de barcos atendidos y muelles usados luego de la solicitud de cambios de tiempos de algunos barcos, mostrados también en la tabla. A partir de ellos se puede concluir que en la mayoría de los casos, no todos los barcos pudieron ser atendidos satisfactoriamente según sus requerimientos, ya que los nuevos tiempos asignados no siempre eran compatibles con los tiempos de las secuencias ya planificadas; mientras que en otros casos si lo eran.

**Tabla 7.6 Resumen de barcos con cambios de tiempo**

Barcos Atendidos	Muelles Usados	Costo Total	Nº Cambios	Barcos Replanificados por Cambios de Tiempo
28	7	542.58	9	22,11,2,24,22,1,17,19,23
29	7	461.13	7	24,7,24,22,17,6,30
30	6	441.12	7	3,20,23,17,17,27,7
20	7	442.13	11	11,17,15,19,30,21,27,12,26,18,10

Para un mejor entendimiento acerca de lo que ocurre al momento de hacer un cambio en la petición, por ende, en la planificación inicial, es que se muestra la siguiente tabla en donde se remarcan con rojo los barcos que cambiaron su petición y cómo es que fueron reasignados en una secuencia, del mismo berth o de otro diferente; integrándose a ello el costo por cada secuencia.

**Tabla 7.7 Ejemplo de secuencia de barcos luego de cambios de tiempo en los barcos**

Secuencia de Barcos			
Muelle	Planificación Inicial	Cambios por Tiempo	Costo
Berth 1	17	24,17	23
Berth 2	13,2,20,14,4,11	13,2,20,14,4,11	84
Berth 3	8,24,12,29	8,12,29	126
Berth 4	1,22,16	16	48
Berth 5	27,7	27,7	8
Berth 6	25,26,18,21,28,10,5,3	25,26,1,18,21,28,10,5,3	86
Berth 7	19,30,23,6,15,9	19,30,6,15,9	86
23,22 No atendidos			

## 7.2.4 Integración de Eventos

La integración de eventos se refiere principalmente a ejecutar el código fuente tomando en cuenta todas las eventualidades que pudiesen ocurrir de manera integrada, o sea, que pudiesen suceder cancelaciones, cambios de tiempos de barcos y desperfectos de muelles, en una misma corrida.

Para una mejor entendimiento de lo planteado y revisarlo de manera más objetiva, es que se diseñó la siguiente tabla, en la que se muestran los resultados experimentales de cuatro corridas diferentes de un mismo archivo de entrada. Se da un detalle de los identificadores de los muelles que sufrieron desperfectos, de los barcos que realizaron cancelaciones de sus petición, y de aquellos que mandaron una nueva solicitud para cambiar su tiempo de arribo. Finalmente se puede concluir que la cantidad de barcos atendidos y la cantidad de muelles usados varía mucho, ya que dependen directamente de los eventualidades que tengan los muelles y barcos, por ende, también son variables los costos totales.

**Tabla 7.8 Resumen completa con todos los eventos posibles**

N° Barcos Atendidos	N° Barcos No Atendidos	N° Muelles Usados	Muelle con Desperfectos	Barcos con Cancelaciones	Barcos con Cambios de Tiempo	Costo Total
23	26,24	6	6	19,7,4,3,14	24,20,18,26,11	381.62
26		6	5	30,24,9,5	11,27,23,19,17,25,1,3,27,8	411.23
23	27,28,7,25,12	3	5,6,7,4	5,6	9,25,12,28	297.77
22	13,2	6	7	18,12,24,19,25,27	20,17,13	434.73

## 7.3 Pruebas de Coordinación de Tiempos

Las pruebas de coordinación de los tiempos se hacen en base a dos criterios, los tiempos reales y simulados, y la asignación real según las ventanas de tiempo creadas.

La primera prueba, se verifica al observar la compatibilidad de los tiempos reales y de simulación. El cumplimiento de esta prueba, se puede verificar al observar los tiempos impresos en la interfaz del módulo de monitoreo. La segunda, corresponde a que las asignaciones se realicen efectivamente dentro de las ventanas de tiempos asignadas para ello. Ambas pruebas internas se realizaron con éxito, verificando que los tiempos funcionan acorde a lo esperado.

## 7.4 Pruebas al Sistema de Monitoreo

La principal función del módulo de monitoreo es ir mostrando la ejecución de todos los eventos ocurridos en un periodo de tiempo. Para que esto funcione de una manera más intuitiva y fácil de entender, se creó una interfaz gráfica simple en la cual se puede visualizar dichos eventos y mediante ella, se pudieron realizar las pruebas necesarias para dicho módulo.

### 7.4.1 Interfaz Gráfica

La interfaz gráfica se despliega al ejecutar el código del Berth Allocation Problem. Consiste en una pantalla, la cual contiene tres paneles principales y cuatro botones, así como también algunos indicadores.

Dentro de los paneles se visualizan los eventos sucedidos de los barcos, los muelles y el planificador, denominados: Vessel Events Panel, Berth Events Panel y Shedule Panel, respectivamente.

Los botones agregados son Start (comienza la ejecución del código), Stop (finaliza la ejecución del código antes del tiempo estipulado en la simulación), Settings y Exit, cada uno cumpliendo la función según el significado de su nombre.

Por último, los indicadores que existen son el Real Time Clock, que muestra el tiempo simulado, el cual está definido código fuente; en este caso es en un tiempo transcurrido entre 10:00:00 y 13:00:00, pudiendo cambiar estos tiempos según se estime conveniente, y el System Time Clock, que corresponde al tiempo real del sistema. Cada uno de los paneles, además nos muestra los tiempos reales y simulados, a medida que transcurre el código; además, en el panel de eventos del muelle, lo primero que se ejecuta es la determinación de los tiempos reales y de simulación: real start time, simu start time, real stop estimated time y simu stop estimated time; y al finalizar la ejecución: real stop time y simu stop time.

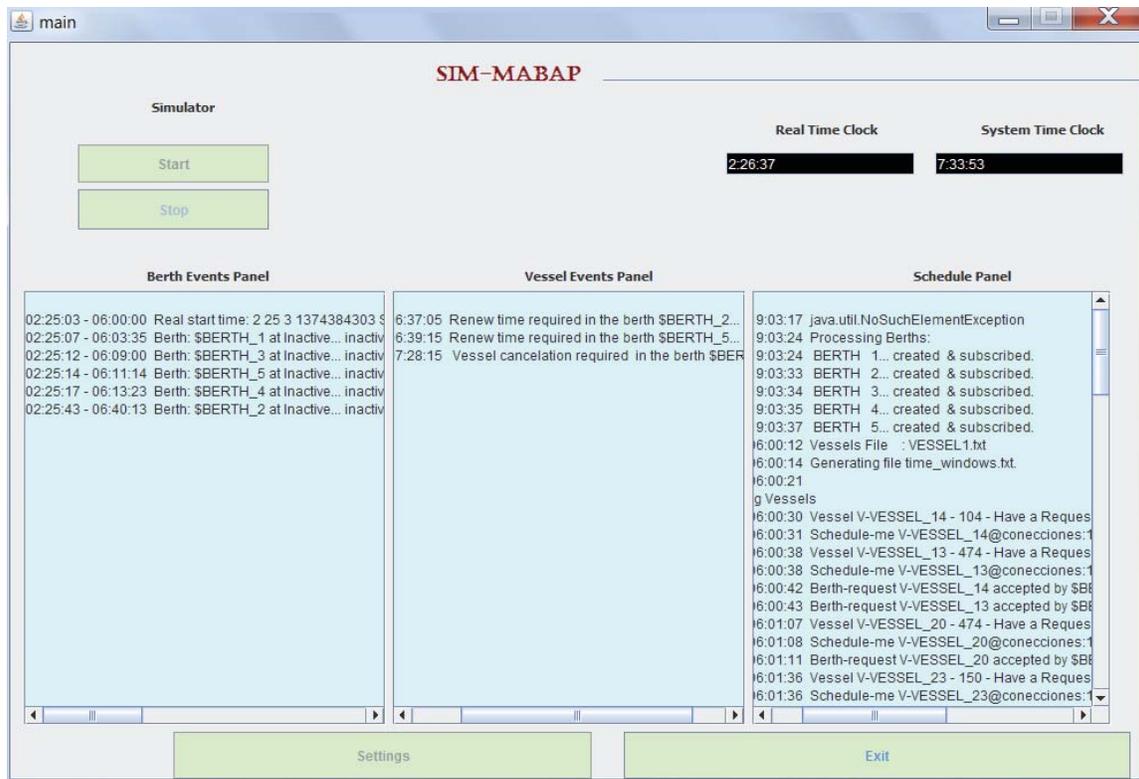


Figura 7.4 Interfaz del módulo de monitoreo

## 7.4.2 Evaluación del Módulo de Monitoreo

Para el correcto funcionamiento del sistema, y por ende, el funcionamiento del módulo de monitoreo y los eventos que lo componen, se realizaron variadas pruebas, como ya se mencionó anteriormente. Todos los resultados de los archivos de pruebas llevados a cabo fueron satisfactorios, y con ello se pudo verificar el comportamiento de los distintos eventos gatillados; reflejándose así mismo, que durante todas las ejecuciones, el sistema fue realizando satisfactoriamente el seguimiento de las peticiones de cada uno de los barcos y la simulación de la asignación de cada uno de ellos dentro del berth, cumpliendo con su objetivo final.

La siguiente tabla muestra un resumen de lo visualizado en la interfaz del módulo de monitoreo, haciendo un seguimiento completo a cada uno de los eventos gatillados y según lo observado en cada panel.

**Tabla 7.9 Ejemplo de eventos gatillados y respuesta según el monitoreo**

Tipo de Evento	Evento	Respuesta al Evento
<b>Evento de Cancelación de una Petición</b>	02:07:31 - 06:12:11 Cancel required on berth \$BERTH_1...	El barco es removido de la planificación del Berth, en este caso BERTH_1
	02:07:31 - 06:12:13 Vessel was removed	
	02:07:31 - 06:12:18 Vessel V-VESSEL_6 removed from vector	
	02:07:31 - 06:12:18 Vessel V-VESSEL_6 was deleted	
<b>Evento de Cambio de Tiempo en la Planificación</b>	02:07:32 - 06:13:42 Renew time required on berth \$BERTH_5...	El Barco es removido y se genera una nueva petición de asignación de dicho barco a un muelle
	02:07:32 - 06:13:44 Vessel was removed	
	02:07:32 - 06:13:44 New time assigned...	
	02:07:32 - 06:13:44 Vessel V-VESSEL_11 was send to new scheduling	
	02:07:32 - 06:13:44 Re-Schedule-me V-VESSEL_11	
	02:07:32 - 06:13:56 Berth-request V-VESSEL_11 accepted by \$BERTH_5	
<b>Evento por Desperfecto del Muelle</b>	02:08:00 - 06:41:05 Berth \$BERTH_3 breakdown...	El Berth es removido, por lo que todos los barcos que tenía asignado, se replanificaron
	02:08:00 - 06:41:05 Berth is removing...at inactive	
	02:08:00 - 06:41:06 Removed	
	02:08:00 - 06:41:06 Vessel V-VESSEL_30 was send to new scheduling	
	02:08:00 - 06:41:06 Vessel V-VESSEL_28 was send to new scheduling	
	02:08:00 - 06:41:06 Vessel V-VESSEL_1 was send to new scheduling	
	02:08:00 - 06:41:06 Vessel V-VESSEL_30 from breakdown berth...	
	02:08:00 - 06:41:06 Vessel V-VESSEL_28 from breakdown berth...	
	02:08:00 - 06:41:06 Vessel V-VESSEL_1 from breakdown berth...	

02:08:00 - 06:41:06 Re-Schedule-me by breakdown V-VESSEL_28
02:08:00 - 06:41:06 Re-Schedule-me by breakdown V-VESSEL_30
02:08:00 - 06:41:06 Re-Schedule-me by breakdown V-VESSEL_1
02:08:00 - 06:41:27 Berth-request V-VESSEL_1 accepted by \$BERTH_2
02:08:01 - 06:41:57 Berth-request V-VESSEL_28 accepted by \$BERTH_5
02:08:01 - 06:42:08 Berth-request V-VESSEL_30 accepted by \$BERTH_5

## 7.5 Archivos de Salida

A partir de la ejecución del código, se obtuvieron variados resultados de las pruebas como se visualizó en los puntos anteriores. Para poder interpretar dichos resultados, se extrajeron los datos de variados archivos de salida obtenidos.

El primer archivo de salida obtenido es aquel que nos muestra las ventanas de tiempo. En él se da el detalle de cada uno de los barcos, junto a sus tiempos de llegada e ida, de procesamiento, visualizando cómo resultó el armado de ellas.

```

time_windows.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
$Name, DRT, MRT, EPT, Nfac, LDT, TSup
V-VESSEL_14, 487, 715, 11 3 35 1375020215, 11 7 23 137502044
V-VESSEL_13, 498, 727, 11 40 13 1375022413, 11 44 2 13750226
V-VESSEL_20, 220, 422, 11 9 0 1375020540, 11 12 22 137502074
V-VESSEL_23, 230, 433, 11 11 12 1375020672, 11 14 35 1375020
V-VESSEL_19, 200, 400, 11 13 22 1375020802, 11 16 42 1375021
V-VESSEL_6, 123, 315, 11 40 3 1375022403, 11 43 15 137502259
V-VESSEL_17, 623, 865, 11 20 59 1375021259, 11 25 1 13750215
V-VESSEL_4, 398, 617, 11 36 49 1375022209, 11 40 28 13750224
V-VESSEL_18, 845, 1109, 11 39 56 1375022396, 11 44 20 137502
V-VESSEL_15, 655, 900, 11 44 58 1375022698, 11 49 3 13750229
V-VESSEL_9, 276, 483, 11 35 38 1375022138, 11 39 5 137502234
V-VESSEL_2, 258, 464, 11 49 47 1375022987, 11 53 12 13750231
V-VESSEL_11, 723, 975, 11 55 4 1375023304, 11 59 16 13750235
V-VESSEL_24, 780, 1038, 11 10 51 1375020651, 11 15 9 1375020
V-VESSEL_12, 222, 424, 11 25 44 1375021544, 11 29 6 13750217
V-VESSEL_29, 430, 653, 11 0 13 1375020013, 11 3 56 137502023
V-VESSEL_26, 700, 950, 11 12 23 1375020743, 11 16 33 1375020
V-VESSEL_10, 301, 511, 11 19 58 1375021198, 11 23 28 1375021
V-VESSEL_25, 110, 301, 11 0 55 1375020055, 11 4 6 1375020246
  
```

Figura 7.5 Resultado de ventanas de tiempos



ejecución de cada uno de los archivos corridos, junto con algunos detalles como lo son barcos atendidos, berts usados, costos totales.

File	Serv. Vessels	Used. Berth	Cost
VESSEL1. txt	30	5	241,42
VESSEL2. txt	30	5	428,95
VESSEL3. txt	30	7	434,32
VESSEL4. txt	30	6	490,10
VESSEL5. txt	30	5	447,07
VESSEL6. txt	30	5	480,22
VESSEL1. txt	30	5	241,42
VESSEL2. txt	30	5	428,95
VESSEL3. txt	28	5	428,28
VESSEL4. txt	29	5	449,08
VESSEL5. txt	30	5	425,07
VESSEL6. txt	30	5	469,22
VESSEL4. txt	13	1	157,62
VESSEL4. txt	27	4	391,70
VESSEL4. txt	28	4	334,23
VESSEL4. txt	29	5	471,27
VESSEL5. txt	30	4	474,25
VESSEL4. txt	13	6	426,52
VESSEL4. txt	29	6	426,52
VESSEL5. txt	27	5	496,33

Figura 7.8 Archivo resumen de ejecuciones

A continuación, se puede apreciar un archivo que denota los barcos que fueron atendidos o no atendidos, junto con el tiempo utilizado para ello.

VesselsFile	Run	Vessel	Status	Time
VESSEL3. txt	/1/V-VESSEL_14/request-served/			94
VESSEL3. txt	/1/V-VESSEL_13/request-served/			78
VESSEL3. txt	/1/V-VESSEL_20/request-served/			62
VESSEL3. txt	/1/V-VESSEL_23/request-served/			47
VESSEL3. txt	/1/V-VESSEL_19/request-served/			63
VESSEL3. txt	/1/V-VESSEL_6/request-served/			31
VESSEL3. txt	/1/V-VESSEL_17/request-served/			47
VESSEL3. txt	/1/V-VESSEL_4/request-served/			31
VESSEL3. txt	/1/V-VESSEL_18/request-served/			31
VESSEL3. txt	/1/V-VESSEL_15/request-served/			47
VESSEL3. txt	/1/V-VESSEL_9/request-served/			78
VESSEL3. txt	/1/V-VESSEL_2/request-served/			141
VESSEL3. txt	/1/V-VESSEL_11/request-served/			94
VESSEL3. txt	/1/V-VESSEL_24/request-served/			78
VESSEL3. txt	/1/V-VESSEL_12/request-served/			32
VESSEL3. txt	/1/V-VESSEL_29/request-served/			31
VESSEL3. txt	/1/V-VESSEL_26/request-served/			47
VESSEL3. txt	/1/V-VESSEL_10/request-served/			46
VESSEL3. txt	/1/V-VESSEL_25/request-served/			47
VESSEL3. txt	/1/V-VESSEL_3/request-served/			47

Figura 7.9 Archivo con el resultado de la solicitud hecha por un barco

## Capítulo 8: Conclusión

En el documento presentado, se planteó como objetivo principal resolver el problema de asignación de barcos a muelles, desarrollándolo mediante un sistema multiagente, para ello se consideró el problema en su versión dinámica. Sin embargo, para lograrlo se dio a conocer en detalle dicho problema de optimización, que se encuentra presente en cualquier puerto, el Berth Allocation Problem (BAP) y además, las características que cumplen los sistemas multiagente. Con ello, se plantearon los objetivos específicos, de los cuales, se han logrado cumplir solo los dos primeros, lo que es un avance importante del proyecto, ya que los demás, se verán en la etapa de implementación del sistema.

El problema de asignación de barcos a muelles dentro de un puerto, tiene una extensa literatura debido a la complejidad de él, cada una de ellas enfoca el problema en su diversidad de alternativas para su resolución. Todos estos estudios hechos por distintos autores, se han considerado al momento de explicar la problemática, como lo es el modelo matemático, métodos de resolución y soluciones obtenidas que ya han sido propuestas, tanto en arquitecturas utilizadas como los métodos de resolución.

Para poder desarrollar el BAP en un sistema multiagente, se analizaron las características, funcionalidades, definiciones y tipologías de los agentes, de manera tal de introducir todos estos conceptos en la arquitectura, metodología y heurística propuesta. Se conoce por lo demás, que la tecnología de agentes dentro de un sistema, ha permitido la descentralización del problema pudiendo delegar tareas a otros agentes, lo que ha logrado que el problema del BAP se simplifique al utilizar como solución para él la tecnología de multiagente.

Respecto a la metodología de desarrollo de agentes, se puede decir que PASSI brinda diferentes diagramas que permiten visualizar gráficamente lo que se desea implementar, por lo que el trabajo de diseño ha sido más simple de identificar y entender, además el hecho que ésta permita la utilización de notación UML, permite que no se ocupe tiempo en aprender una nueva. Todos los modelos que se hicieron, sirven como base efectiva para el comienzo de la etapa de implementación del sistema. Además éstos nos entregan una visión más clara del sistema, definiendo a cada uno de los agentes involucrados y a las tareas que ellos cumplen, acercándonos de manera real a la automatización.

En cuanto a la heurística, se concluye que a pesar de la dificultad que se tuvo para estudiarla y comprenderla, ésta es bastante útil en el sentido de la ejecución, y que las modificaciones que se realizaron para implementar la nueva heurística propuesta, fueron las necesarias para permitir dar una solución a la versión dinámica del BAP y que el algoritmo de inserción utilizado, también lo permite.

Este proyecto busca implementar una nueva arquitectura, en donde la arquitectura propuesta señala un cambio drástico respecto a la original, si bien permite realizar toda la planificación de los barcos y los berths, también permite realizar la simulación de ellos mediante la capa de monitoreo que se agregó, lo cual va a permitir en la etapa de implementación, probar la heurística escogida.

En lo que respecta al diseño, se han completado la mayoría de los modelos propuestos por PASSI, realizando así de mejor manera esta etapa, quedando para el final algunos diseños referentes básicamente al código del sistema.

Las pruebas son una parte esencial del sistema, para ello se planificaron y se diseñaron las pruebas necesarias para el correcto funcionamiento del sistema y junto con ellas, el modelo del archivo a utilizar.

El mayor desafío que encontramos en esta primera etapa, fue el hecho de proponer y establecer una nueva arquitectura, que abarcara todo lo que queremos demostrar en el sistema final y que a la vez, fuese aceptable dentro de la solución del problema de optimización del BAP, y junto con ello, la heurística.

A modo de conclusión en esta etapa de pruebas, se puede decir que se verificó el cumplimiento total de la funcionalidad del sistema, probándolo primero por módulos, cada uno por separado y luego su integración. Respecto a los resultados de los datos entregados, se puede decir que existe una gran diferencia entre el sistema visto en su escenario estático y en su escenario dinámico; en el escenario dinámico, al ir variando las solicitudes, y las condiciones de los berths y barcos, es que los resultados de los costos varían al ejecutar varias corridas en un mismo archivo, mientras que en el escenario dinámico no hay muchas diferencias entre ellos, obteniendo un promedio cercano a los resultados de cada corrida. Se puede decir además, que el sistema planteado es bastante escalable, pudiendo seguir realizándole mejoras e integrando más funcionalidades.

## Capítulo 9: Referencias Bibliográficas y Bibliografía

- [1] Mas, Ana. “Agentes Software y Sistemas Multi-Agente: Conceptos, Arquitecturas y Aplicaciones”. 2005, Pearson Education.
- [2] RAE “Real Academia de la Lengua Española”. <http://www.rae.es>.
- [3] Michael Wooldridge, “An Introduction to Multiagent Systems”. Dept. of Electronic Engineering, Queen Mary & Westfield College, 2002.
- [4] Dra María del Carmen Romero Ternero, Sistemas Multiagente, Programa de Doctorado Informática Industrial, Universidad de Sevilla.
- [5] Juan de Lara, Sistemas Multiagente, Escuela Politécnica Superior Ingeniería Informática.
- [6] Haddadi, A; y Sundermeyer K. Belief-Desire-Intention Agent Architectures. Wiley-Interscience Publication , 1996.
- [7] Rao, A.S.; y Georgeff, M.P.: BDI Agents from Theory to Practice. Proceedings of the First International Conference on Multi-Agents Systems (ICMAS-95), 1995.
- [8] Brooks, R.A.: Intelligence without Representation, Artificial Intelligence, 47, 139-159, 1991.
- [9] Weiss, G. “Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence”. MIT Press, Massachusetts, USA. 1999.
- [10] Cubillos C., Agentes Inteligentes y Sistemas Multiagente, Material Docente, Escuela de ingeniería Informática, Pontificia Universidad Católica de Valparaíso, 2012.
- [11] Foundation of Intelligent Physical Agents (FIPA), “FIPA ACL Message Structure Specification”. <http://www.fipa.org/>
- [12] Jorge J. Gómez Sanz. Metodologías para el Desarrollo de Sistemas Multiagente Inteligencia Artificial. Departamento de Ciencias e Ingeniería de la Computación, 2005.
- [13] Inglesias C. Definición de una Metodología para el Desarrollo de Sistemas Multiagente, Universidad Politécnica de Madrid, 1998
- [14] Marchetti, y García. Metodologías de Desarrollo de Sistemas Multi-Agente: Un Análisis Comparativo. Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA), Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur.

- [15] Arango et al. Operativa de Buques en las Terminales Marítimas de Contenedores: Estado del Arte. 5th International Conference on Industrial Engineering and Industrial Management. XV Congreso de Ingeniería de Organización. Cartagena, 7 a 9 de Septiembre de 2011
- [16] Melián et al. Una Primera Aproximación al Problema de Asignación de Atraques con Tiempos de Llegada Difusos. Dpto. de Estadística, IO y Computación. Escuela Técnica Superior de Ingeniería Informática. Universidad de La Laguna
- [17] Cordeau et al. Models and Tabu Search Heuristics for the Berth Allocation Problem, Paper publicado en Abril de 2005
- [18] Frank Meisel. "Seaside Operations Planning in Container Terminal". Publicado en 2009
- [19] Sammarra. "La Logistica Portuale". Letteratura Scientifica sulla Gestione Operativa <http://www.rdlog.it/storico/osservatorio-st13.html>. Revisado por última vez el 1 de Diciembre de 2012
- [20] DTU Management Engineering. "Models for the Discrete Berth Allocation Problem: A Computational Comparison", Department of Informatics and Mathematical Modelling, Technical University of Denmark
- [21] Cubillos, C. et al: "On user requirements and operator purposes in Dial-a-Ride services". Proceedings of the 9th Meeting of the EURO Working Group on Transportation.
- [22] Cubillos, C.: MADARP: Multi-Agent Framework for Transportation Systems. Tesis para optar al grado de Dottore di Ricerca in Ingegneria Informatica e de Sistemi (cicloXVII), Politecnico di Torino, 2005.
- [23] Botti, Universidad Politécnica de Valencia. *Multi-Agent System Technology in a Port Contenedor Terminal Automation*. Ercim News, Número 56 Enero ,pags 37-39, 2004.
- [24] Thangiah, S, et al.: An Agent Architecture for Vehicle Routing Problems. 2001.
- [25] Departamento de ciencias de la computación e I.A. Universidad de Granada. Algoritmos Greedy. Análisis y Diseño de algoritmos.
- [26] Moreno. Metaheurísticas: Conceptos y propiedades. Materail Docente Universidad De Los Lagos. Año 2000
- [27] Jaw et al. A Heuristic Algorithm for the Multi-Vehicle Advance Request Dial A Ride Problem with Time Windows.

- [28] Díaz R. “Desarrollo de un Sistema Multiagente para el Berth Allocation Problem”. Tesis para optar el título de Ingeniero en Informática, 2007.
- [29] Urra E. “Estudio de Algoritmos Genéticos para el Problema de Transporte de Pasajeros (DARPTW)”. Tesis para optar el título de Ingeniero Civil en informática, 2007.
- [30] Donoso M., Sandoval D. Desarrollo de un Sistema para el Dynamic Dial a Ride Problem utilizando Tecnología de Agentes. Tesis para optar el título de Ingeniero de Ejecución en informática, 2007.
- [31] Lim A. et al. The Berth Planning Problem. Paper publicado el año 1998.
- [32] Imai A.; Nishimura E.; Papadimitriou S. The Dynamic Berth Allocation Problem for a Container Port. Paper publicado el año 2001
- [33] Legato P y Mazza R.. Berth Planning and Resource Optimisation at a Container Terminal Via Discrete Event Simulation. Paper publicado el año 2001.
- [34] Nishimura E et al.. Berth Allocation Planning in the Public Berth System by Genetic Algorithms. Paper publicado el año 2001.
- [35] Imai A et al.. Berth Allocation Problem with Service Priority. Paper publicado el año 2003.
- [36] Kim K. y Moon K.. Berth Scheduling by Simulated Annealing. Paper publicado el año 2003.
- [37] Park Y.-M y Kim K. H.. A Scheduling Method for Berth and Quay Cranes. Paper publicado el año 2003.
- [38] Guan Y Cheung R. The Berth Allocation Problem: Models and Solution Methods. Paper publicado el año 2004.
- [39] Cordeau J et al.. Models and Tabu Search Heuristics for the Berth Allocation Problem. Paper publicado el año 2005
- [40] Imai A. et al.. Berth Allocation in a Container Port: Using a Continuous Location Approach. Paper publicado el año 2005.
- [41] Hansen P et al.. Variable Neighborhood Search for Minimum Cost Berth Allocation. Paper publicado el año 2006.

- [42] Moorthy R. y Teo C.-P.. Berth Management in Container Terminal: The Template Design Problem. Paper publicado el año 2006.
- [43] Imai A. et al.. Berth Allocation at Indented Berths for Mega-containerships. Paper publicado el año 2007, parte a.
- [44] Monaco M y Sammarra M. The Berth Allocation Problem: a Strong Formulation Solved by a Lagrangean Approach. Paper publicado el año 2007.
- [45] Lokuge, P., Alahakoon, D. Improving the Adaptability in Automated Vessel Scheduling in Container Ports Using Intelligent Software Agents. Paper publicado el año 2007.
- [46] Imai A et al. The Simultaneous Berth and Quay Crane Allocation Problem. Paper publicado el año 2007, parte b.
- [47] Imai A et al.. Berthing Ships at a Multi-user Container Terminal with a Limited Quay Capacity. Paper publicado el año 2008.
- [48] Meisel y Bierwirth. A Fast Heuristic for Quay Crane Scheduling with Interference Constraints. Paper publicado el año 2009.
- [49] Ismael Rihawi Aragón, Poiritem, un Planificador Orientativo Inteligente de Rutas e Itinerarios Turísticos en el Entorno Metropolitano, <http://poiritem.wordpress.com/2009/11/16/6-4-2-agentes-inteligentes-y-la-naturaleza-de-su-entorno/> Revisado por última vez el 1 de Diciembre de 2012
- [50] <http://members.fortunecity.com/avzqznweb/multiagents/definicion.htm>. Revisado por última vez el 1 de Diciembre de 2012
- [51] [http://www.fortunecity.com/meltingpot/estonia/427/multiagents/clasif\\_agentes.htm](http://www.fortunecity.com/meltingpot/estonia/427/multiagents/clasif_agentes.htm). Revisado por última vez el 1 de Diciembre de 2012
- [52] José Luis Coronel Cortés, Introducción a FIPA. Disponible vía web en <http://dukechile.blogspot.com/2008/02/introduccion-fipa-parte-1.html> Revisada por última vez el 28 de Julio de 2013
- [53] <http://personales.upv.es/ccarrasc/doc/2004-2005/LaWebSemanticaFernandezRoigSetal/www/tresHerr.html>. Revisado por última vez el 1 de Diciembre de 2012

[54]Infraestructura y medios de transporte, Disponible vía web en <http://www.skyscrapercity.com/showthread.php?t=654714>, Revisado por última vez el 1 de Diciembre de 2012

[55]Oscar Lozano, Búsqueda Tabú, Disponible vía web en <http://dis.unal.edu.co/~fgonza/courses/2003/pmge/present/tabusearch.pdf>, Revisado por última vez el 1 de Diciembre de 2012

[56][http://e-spacio.uned.es/fez/eserv.php?pid=tesisuned:IngIndMgomez&dsID=TESIS\\_MGG1\\_Capitulo3.pdf](http://e-spacio.uned.es/fez/eserv.php?pid=tesisuned:IngIndMgomez&dsID=TESIS_MGG1_Capitulo3.pdf), Revisado por última vez el 1 de Diciembre de 2012

[57]<http://www.mitecnologico.com/Main/DefinicionYModelosDeProgramacionEntera>, Revisado por última vez el 1 de Diciembre de 2012

[58] Natyhelem Gil Londoño, Algoritmos Genéticos <http://www.monografias.com/trabajos-pdf/algoritmos-geneticos/algoritmos-geneticos.pdf>, Revisado por última vez el 1 de Diciembre de 2012

[59] Telecom Italia S.A. <http://jade.tilab.com/>, Revisado por última vez el 1 de Diciembre de 2012