

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

**DESARROLLO DE UN SISTEMA MULTIAGENTE  
PARA PROBLEMAS DE INVENTARIO**

**RICARDO ANDRES MUÑOZ OLGUIN**

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

DICIEMBRE 2011

Pontificia Universidad Católica de Valparaíso  
Facultad de Ingeniería  
Escuela de Ingeniería Informática

**DESARROLLO DE UN SISTEMA MULTIAGENTE  
PARA PROBLEMAS DE INVENTARIO**

**RICARDO ANDRES MUÑOZ OLGUIN**

Profesor Guía: **Claudio Cubillos Figueroa**

Profesor Co-referente: **Nibaldo Rodríguez Agurto**

Carrera: **Ingeniería Civil Informática**

DICIEMBRE 2011

# Índice de Contenidos

<b>ÍNDICE DE CONTENIDOS</b> .....	<b>I</b>
<b>LISTA DE FIGURAS</b> .....	<b>III</b>
<b>LISTA DE TABLAS</b> .....	<b>IV</b>
<b>RESUMEN Y ABSTRACT</b> .....	<b>V</b>
RESUMEN .....	v
ABSTRACT .....	v
<b>1. DESCRIPCIÓN DEL PROBLEMA</b> .....	<b>1</b>
1.1. INTRODUCCIÓN .....	1
1.2. DEFINICIÓN DE OBJETIVOS .....	2
1.2.1. <i>Objetivo General</i> .....	2
1.2.2. <i>Objetivos Específicos</i> .....	2
1.3. METODOLOGÍA DE DESARROLLO .....	2
1.4. ESTRUCTURA DEL DOCUMENTO.....	3
<b>2. SISTEMAS MULTIAGENTE</b> .....	<b>4</b>
2.1. DEFINICIÓN DE AGENTE .....	4
2.2. ARQUITECTURAS DE AGENTES.....	5
2.2.1. <i>Arquitecturas Deliberativas</i> .....	5
2.2.2. <i>Arquitecturas reactivas</i> .....	6
2.2.3. <i>Arquitecturas Híbridas</i> .....	7
2.3. SISTEMAS MULTIAGENTE .....	8
2.3.1. <i>Dominios de Aplicación</i> .....	9
2.4. ESTÁNDARES MULTIAGENTE .....	9
2.4.1. <i>Arquitectura FIPA</i> .....	9
2.4.2. <i>Infraestructura de Agentes</i> .....	10
2.4.3. <i>Ontologías</i> .....	11
2.4.4. <i>Lenguajes de Comunicación de Agentes</i> .....	12
2.4.5. <i>Protocolos de Interacción</i> .....	14
2.5. METODOLOGÍAS DE DESARROLLO .....	14
2.5.1. <i>BDI</i> .....	15
2.5.2. <i>MaSE</i> .....	15
2.5.3. <i>GAIA</i> .....	16
2.5.4. <i>PASSI</i> .....	17
<b>3. TEORÍA DE INVENTARIOS</b> .....	<b>19</b>
3.1. DEFINICIÓN .....	19
3.2. EVALUACIÓN DE LOS INVENTARIOS.....	19
3.2.1. <i>Argumentos a favor de los inventarios</i> .....	20
3.2.2. <i>Argumentos en contra de los inventarios</i> .....	20
3.3. TIPOS DE INVENTARIOS .....	20
3.4. OBJETIVOS DEL INVENTARIO .....	21
3.4.1. <i>Disponibilidad del producto</i> .....	21
3.4.2. <i>Costos pertinentes</i> .....	22
3.5. TIPOS DE DEMANDA .....	23
3.6. POLÍTICAS DE CONTROL DE INVENTARIOS.....	23
3.6.1. <i>Control de inventarios por incrementos (push)</i> .....	23
3.6.2. <i>Control básico de inventarios por demanda (pull)</i> .....	24
3.6.3. <i>Control avanzado de inventarios por demanda (pull)</i> .....	26
3.6.4. <i>Método del punto de reorden con costos conocidos de falta de existencias</i> .....	28

3.6.5. Método del punto de reorden con tiempos de demanda y entrega inciertos.....	29
3.6.6. Modelo de revisión periódica con demanda incierta.....	29
3.7. CONTROL MULTINIVEL .....	32
3.8. SISTEMAS MULTIAGENTE EN LA CADENA DE SUMINISTRO.....	33
3.8.1. R. Sarker - A Multi-agent Simulation Study for Supply Chain Operation .....	34
3.8.2. Sara Saberi - Multi Agent System For Negotiation In Supply Chain Management .....	37
3.8.3. Manuel D. Rossetti – An Object-Oriented Framework For Simulating Multi-Echelon Inventory Systems .....	39
<b>4. SOLUCIÓN PROPUESTA .....</b>	<b>42</b>
4.1. DESCRIPCIÓN DE LA PROPUESTA .....	42
4.2. JUSTIFICACIÓN .....	44
4.3. ALCANCES Y LIMITACIONES.....	44
4.4. RESULTADOS ESPERADOS .....	45
4.5. ESPECIFICACIONES DE DISEÑO .....	45
4.5.1. Modelo de Requerimientos del sistema .....	45
4.5.2. Modelo de Sociedad de Agentes .....	54
4.5.3. Modelo de Implementación de Agentes.....	56
<b>5. IMPLEMENTACIÓN Y RESULTADOS DE LA ARQUITECTURA PROPUESTA .....</b>	<b>62</b>
5.1. ARCHIVOS DE ENTRADA Y SALIDA.....	62
5.2. IMPLEMENTACIÓN DEL SISTEMA .....	64
5.3. RESULTADOS .....	65
<b>6. ASPECTOS FINALES .....</b>	<b>68</b>
6.1. CONCLUSIONES .....	68
6.2. TRABAJO FUTURO .....	69
<b>7. REFERENCIAS .....</b>	<b>70</b>

## Lista de Figuras

Figura 2.1: Arquitectura Deliberativa .....	6
Figura 2.2: Arquitectura de subsunción .....	7
Figura 2.3: Arquitectura Híbrida (Interrap).....	7
Figura 2.4: Modelo de Arquitectura FIPA. ....	10
Figura 2.5: Modelos y fases de PASSI .....	17
Figura 3.1: Los inventarios se sitúan en cada nivel (o escalón) de la cadena de suministro. ...	19
Figura 3.2: Equilibrio de los costos pertinentes del inventario con la cantidad de pedido. ....	22
Figura 3.3: Modelo de control básico de inventarios de demanda (pull) para una parte de reaprovisionamiento.....	26
Figura 3.4: Control de inventario de revisión periódica con incertidumbre para un artículo. ..	30
Figura 3.5: Cadena de suministro, desde el proveedor (S) hasta el minorista (R). ....	35
Figura 4.1: Descripción del Dominio.....	46
Figura 4.2: Identificación de Agentes .....	48
Figura 4.3: Realizar pedido al retailer con stock suficiente .....	49
Figura 4.4: Realizar pedido al supplier .....	50
Figura 4.5: Inventory Agent .....	51
Figura 4.6: Dealer Agent.....	52
Figura 4.7: Supplier Agent .....	53
Figura 4.8: Descripción de la ontología de dominio.....	54
Figura 4.9: Descripción de Roles.....	55
Figura 4.10: Definición de la Arquitectura Multiagente.....	57
Figura 4.11: Inventory Agent .....	58
Figura 4.12: Dealer Agent.....	59
Figura 4.13: Supplier Agent .....	59
Figura 4.14: Monitor Agent.....	60
Figura 4.15: Descripción del Comportamiento Multiagente.....	61
Figura 5.1: Archivo Demand.txt.....	62
Figura 5.2: Archivo Inventories.txt.....	62
Figura 5.3: Archivo Suppliers.txt .....	63
Figura 5.4: Archivo Items.txt .....	63
Figura 5.5: Archivo Results.txt.....	63
Figura 5.6: Interfaz gráfica del sistema: Pestaña del Retailer 1 .....	64
Figura 5.7: Cadena de Suministro de Prueba .....	65
Figura 5.8: Items de la simulación de prueba.....	65
Figura 5.9: Inventarios de la simulación de prueba .....	66
Figura 5.10: Nivel de servicio de la simulación de prueba. ....	66
Figura 5.11: Nivel de inventario del ítem 1 de la simulación de prueba. ....	67

## Lista de Tablas

Tabla 2.1: Elementos de mensaje FIPA ACL. ....	12
Tabla 2.2: Parámetros reservados en un mensaje KQML.....	13
Tabla 2.3: Protocolos de comunicación FIPA.....	14

# Resumen y Abstract

## Resumen

Uno de los problemas principales de las empresas corresponde a tratar la logística, y más específicamente, el problema de la cadena de suministro, donde el control y manejo de inventarios tiene un rol fundamental, ya que permite lograr un equilibrio entre el nivel de servicio al cliente y los costos asociados de mantener el inventario. Este proyecto tiene por objetivo entregar una propuesta de sistema para resolver el problema de control de inventarios, tomando en cuenta la multiplicidad de niveles que pueden alcanzarse, y realizándose según alguna de las políticas de inventario encontradas en la literatura, todo bajo la metodología PASSI de desarrollo de sistemas multiagente. Como resultado se ha obtenido una simulación de una cadena de suministro, utilizando la política de revisión periódica con demanda incierta, la cual cumple con entregar altos niveles de servicio al cliente.

*Palabras-claves: Sistemas Multiagente, Inventario Multinivel, Cadena de Suministro, PASSI.*

## Abstract

One of the main problems for companies is to deal with logistics, and more specifically, the problem of the supply chain where inventory control and management has a fundamental role, allowing a balance between service level to clients and the costs associated with maintaining inventory. This project aims to deliver a proposal for a system to solve the problem of inventory control, taking into account the multiplicity of levels that can be achieved, and following one of the inventory policies found in the literature, all under the PASSI methodology for multiagent systems development. As a result we have obtained a simulation of a supply chain, using the policy of periodic review under uncertain demand, which conforms to provide a high service level.

*Keywords: Multiagent System, Multi-echelon inventories, Supply Chain, PASSI*

# 1. Descripción del Problema

## 1.1. Introducción

Cuando se inicia la concepción de una empresa, una de las principales motivaciones para su creación es la oportunidad de lograr utilidades que den un mayor bienestar a todos quienes participan en el proyecto. Para llevar esto a cabo, se necesitan políticas empresariales capaces de lograr una maximización de las ganancias, reduciendo los gastos en los que se incurre. Si se quiere tener éxito en el negocio, se debe tener el control de la cadena de suministro, que consiste en todas las partes involucradas directa o indirectamente en el objetivo de satisfacer las necesidades del cliente. Esta cadena comienza desde la obtención de las materias primas, su procesamiento y la entrega al cliente final.

Uno de los puntos más sensibles en esta cadena es el control de inventario de la empresa, que son los bienes tangibles que se tienen para la venta en el curso ordinario del negocio o para ser consumidos en la producción de bienes o servicios para su posterior comercialización. Los inventarios comprenden, además de las materias primas, productos en proceso y productos terminados o mercancías para la venta, los materiales, repuestos y accesorios para ser consumidos en la producción de bienes fabricados para la venta o en la prestación de servicios; empaques y envases y los inventarios en tránsito.

La base de toda empresa comercial es la compra y venta de bienes o servicios; de aquí la importancia del manejo del inventario por parte de la misma. Este manejo contable permitirá a la empresa mantener el control oportunamente, así como también conocer al final del período contable un estado confiable de la situación económica de la empresa.

Debido a su importancia, se hace imprescindible para empresas que ven crecer su inventario tener sistemas de control que faciliten el manejo de todas las actividades que se van desarrollando, sobre todo si se trata del manejo de más de múltiples bodegas, como pueden ser las de abastecimiento externo, centros de distribución y bodegas del cliente. Mantener un buen control de inventario supone un gran ahorro para la empresa, y una ventaja importante frente a sus competidores.

El presente proyecto tiene por objetivo diseñar un prototipo de sistema para el problema de control de inventarios, tomando en cuenta la multiplicidad de niveles que estos pueden alcanzar, realizándose según las políticas de inventario que han sido instauradas a lo largo del tiempo de desarrollo de este.

Para llevar a cabo una solución capaz de satisfacer las características de un sistema de inventario en un ambiente distribuido y que actúa rápida y eficazmente frente a los cambios del ambiente, se contó con una tecnología inteligente que pudiese tomar las decisiones necesarias. La tecnología a utilizar ha sido la de sistemas multiagentes.

## **1.2. Definición de Objetivos**

### **1.2.1. Objetivo General**

El objetivo de este proyecto es diseñar un prototipo de sistema multiagente para el problema de gestión de inventarios.

### **1.2.2. Objetivos Específicos**

- Investigar la literatura sobre la teoría del control y gestión de inventarios, junto a las propuestas de solución a sus problemas, para poder recoger eficazmente los requerimientos del sistema.
- Investigar el estado del arte de sistemas multiagente para poder seleccionar la mejor plataforma y metodología de desarrollo.
- Diseñar un sistema multiagente que se adecue a los requerimientos del problema de control de inventarios, teniendo presente la estructura, comportamientos, interacciones y conocimientos de los agentes.
- Documentar el sistema, de tal forma que no resulte difícil ser retomando para su mejora en futuros proyectos.
- Validar el diseño realizando un prototipo con las partes claves del sistema para demostrar que este diseño se puede utilizar con confianza como base para la construcción de un sistema para problemas de inventario.

## **1.3. Metodología de Desarrollo del Proyecto**

La Metodología de Desarrollo utilizada en este proyecto es la Incremental junto con la Metodología PASSI para el Desarrollo de Sistemas Multiagente. La Metodología Incremental es una mezcla entre las Metodologías Cascada e Iterativa, dividiendo el proyecto en mini cascadas, donde se realiza todo el proceso de desarrollo en cascada para pequeñas partes del sistema. Con esto se espera tener un mayor control del proyecto que la Metodología en Cascada, ya que se van realizando las modificaciones que se requieran en cada uno de las mini cascadas realizadas, acotando los distintos problemas y desafíos que se presentan.[22]

En el caso particular del Sistema Multiagente propuesto, en primer lugar se definieron los agentes que debían participar, para luego diseñarlos e implementarlos, junto con la comunicación entre ellos Para definir los agentes y su comunicación, se diseñaron los diagramas pertenecientes al Modelo de Requerimientos del Sistema de PASSI. En una segunda etapa se trabajó en la generación de la demanda y en una política de inventarios básica (si el material solicitado está disponible se entrega, sino se solicita al proveedor) para

poder probar el funcionamiento del sistema en su totalidad, generando también los restantes diagramas PASSI utilizados. En la tercera fase del proyecto, se diseñó e implementó la política de inventario finalmente utilizada (revisión periódica con demanda incierta).

En la sección 2.5.4 del presente documento se puede encontrar mayor información sobre la Metodología de Desarrollo PASSI utilizada en este proyecto.

## **1.4. Estructura del Documento**

Los contenidos presentes en este documento corresponden a los obtenidos en el proyecto, desde la investigación del estado del arte hasta la implementación y obtención de resultados del sistema.

En el capítulo 1 se presenta la introducción del proyecto, junto a los objetivos y la metodología utilizada.

En el capítulo 2 se presentan los contenidos de la primera fase de investigación, asociado al paradigma de agentes, sistemas multiagente y a las tecnologías asociadas a su desarrollo e implementación, dando a conocer las alternativas de trabajo bajo este enfoque.

En el capítulo 3 se muestran los contenidos asociados a la teoría de inventarios, donde se mencionan las clasificaciones de inventarios, los costos asociados, las características de la demanda, los inventarios multi-nivel o multi-escalón y algunas investigaciones en el tema.

En el capítulo 4 se presenta la solución propuesta, donde se incluye la descripción de la propuesta, la justificación del proyecto, alcances y limitaciones, los resultados esperados, y las especificaciones de diseño, donde se presentan los distintos modelos realizados siguiendo la metodología PASSI.

El capítulo 5 contiene finalmente los resultados del trabajo y las conclusiones obtenidas en el desarrollo del proyecto, junto al trabajo futuro.

## 2. Sistemas Multiagente

### 2.1. Definición de Agente

Definir que es un agente no ha sido una tarea trivial, ya que no existe un real consenso entre los autores avocados en este tema para definirlo. Además, el nombre de agente es usado para múltiples áreas y disciplinas. Dentro de las áreas en que se utiliza están la gramática, la medicina, la economía y la inteligencia artificial. Inclusive, dentro de este último contexto, casi todos los investigadores difieren en su definición, ya que se deben tener en cuenta algunos aspectos no menores cuando se definen los agentes. ¿Qué es lo que realmente hace un agente? ¿Actúan solos los agentes en nuestro nombre? ¿Debemos confiar en ellos? ¿Estamos preparados para entregar datos privados para realizar una tarea en la red? ¿Queremos que sean totalmente autónomos, dirigidos, o sólo esclavos? ¿Queremos que aprendan de nosotros y de su entorno?

Algunas de las definiciones de agentes encontrados en la literatura son:

*“Toda entidad que de forma autónoma perciba su entorno (real o simulado) mediante sensores y que actúe en el mismo mediante efectores”*, según Russel & Norwig [1].

*“Un sistema informático que se sitúa en un cierto ambiente y que es capaz de acción autónoma en este ambiente para lograr sus objetivos de diseño”*, según Wooldbridge [2].

Las teorías de agentes son especificaciones para conceptualizar los agentes. Debido a que la definición de agente ha resultado ser controvertida, se ha optado por una definición de un conjunto de propiedades que caracterizan a los agentes, aunque un agente no tiene que poseer todas estas propiedades.

Algunas características de agentes, basados en el enfoque de la Ingeniería de Software Orientado a Agentes (AOSE) son:

- **Autonomía:** Los agentes deben ser capaces de actuar sin la intervención directa de seres humanos o de otros agentes, y deben tener el control sobre sus propias acciones y estado interno.
- **Reactividad:** Es la capacidad de percibir cambios en el ambiente y responder a ellos siguiendo sus objetivos de diseño.
- **Pro-Actividad:** Es la iniciativa demostrada por el agente para exhibir un comportamiento orientado por sus metas.
- **Habilidad Social:** Es la capacidad de interactuar con otros agentes y seres humanos como medio de alcanzar un objetivo o meta.

La cuestión de qué es un agente, como hemos comentado, está aún siendo debatida, corriendo el riesgo de que cualquier programa sea denominado agente. Se pueden distinguir dos nociones extremas de agentes [3]:

Una noción débil de agente consiste en definir un agente como a una entidad que es capaz de intercambiar mensajes utilizando un lenguaje de comunicación de agentes. Esta definición es la más utilizada dentro de la ingeniería software basada en agentes, cuyo fin es conseguir la interoperabilidad entre aplicaciones a nivel semántico utilizando la emergente tecnología de agentes. Las características antes mencionadas son generalmente las que definen un agente con noción débil.

Una noción más fuerte o restrictiva de agente es la enunciada por Shoham [3] en su propuesta de programación orientada a agentes (AOP), donde un agente se define como una entidad cuyo estado es visto como un conjunto de componentes mentales, tales como creencias, capacidades, elecciones y acuerdos.

Las características que debe poseer un agente para tener una noción fuerte son:

- **Movilidad:** La capacidad de emigrar de una manera autónoma por la red (de un host a otro).
- **Veracidad:** Un agente no comunicará con conocimiento información falsa.
- **Benevolencia:** Un agente está dispuesto a ayudar a otros agentes si esto no entra en conflicto con sus propios objetivos.
- **Racionalidad:** Un agente actúa de forma racional, intentando cumplir sus objetivos si son viables.

## **2.2. Arquitecturas de Agentes**

Las arquitecturas de agentes describen la interconexión de los módulos software/hardware que permiten a un agente exhibir la conducta enunciada en las teorías de agentes. Frente a otras tecnologías con componentes fijos como la de objetos (atributos y métodos) o la de sistemas expertos (motor de inferencias, base de conocimiento y otros elementos opcionales), en los agentes nos encontramos con una gran variedad de arquitecturas.

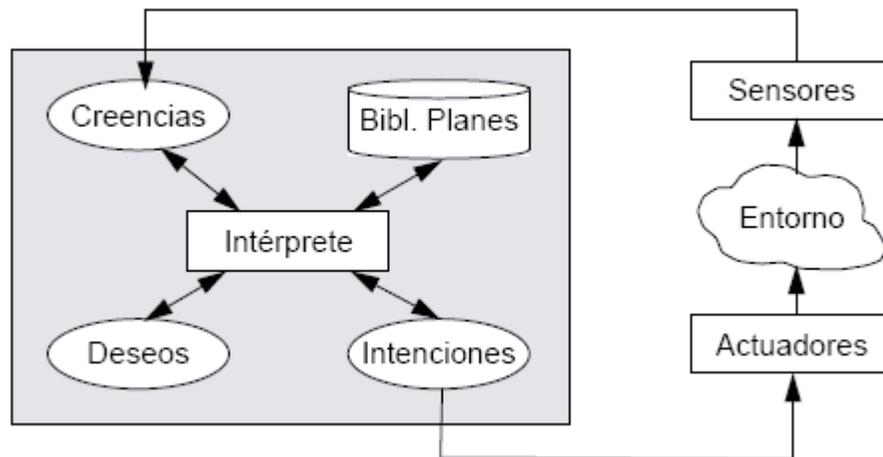
Podemos clasificar las arquitecturas atendiendo al tipo de procesamiento empleado en deliberativas, reactivas e híbridas.

### **2.2.1. Arquitecturas Deliberativas**

Las arquitecturas deliberativas siguen la corriente de la IA simbólica, que se basa en la hipótesis de los sistemas de símbolos-físicos enunciada por Newell y Simons, según la cual un sistema de símbolos físicos capaz de manipular estructuras simbólicas puede exhibir una

conducta inteligente. Para poder trabajar en el nivel de Conocimiento de Newell, nuestro problema será cómo describir los objetivos y medios de satisfacerlos, y cómo realizar la traducción del nivel de conocimiento al nivel simbólico.

Las arquitecturas de agentes deliberativos suelen basarse en la teoría clásica de planificación de inteligencia artificial: dado un estado inicial, un conjunto de operadores/planes y un estado objetivo, la deliberación del agente consiste en determinar qué pasos debe encadenar para lograr su objetivo, siguiendo un enfoque descendente (top-down). [7]



**Figura 2.1: Arquitectura Deliberativa [7]**

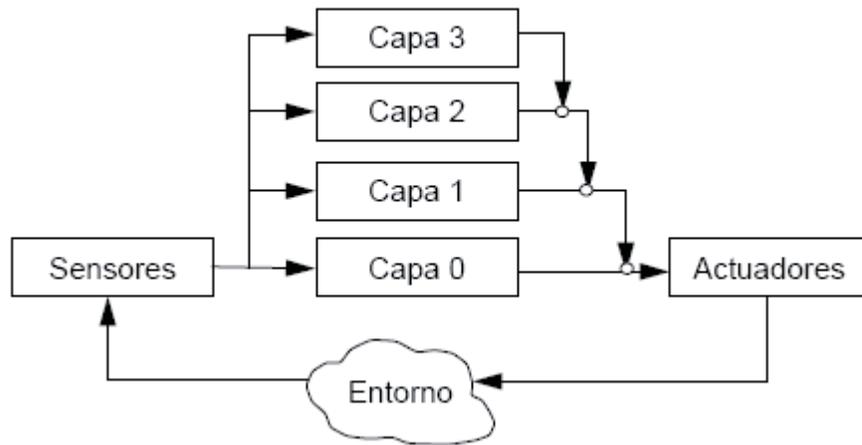
## 2.2.2. Arquitecturas reactivas

Las arquitecturas reactivas [20] cuestionan la viabilidad del paradigma simbólico y proponen una arquitectura que actúa siguiendo un enfoque conductista, con un modelo estímulo-respuesta. Las arquitecturas reactivas no tienen un modelo del mundo simbólico como elemento central de razonamiento y no utilizan razonamiento simbólico complejo, sino que siguen un procesamiento ascendente (bottom-up), para lo cual mantienen una serie de patrones que se activan bajo ciertas condiciones de los sensores y tienen un efecto directo en los actuadores. Esta discusión entre mantener una representación explícita del modelo o no, no es una discusión específica del campo de agente sino de la inteligencia artificial en general, de hecho las primeras arquitecturas de agentes reactivos se basan en los planificadores reactivos.

Las principales arquitecturas reactivas son [3]:

- Reglas situadas
- Arquitecturas de subsunción (subsumption) y autómatas de estado finito
- Tareas competitivas

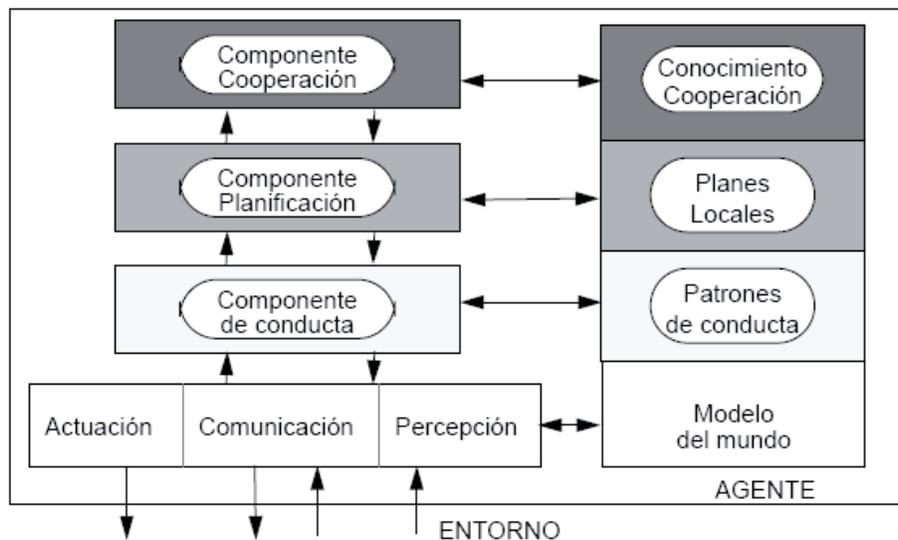
- Redes neuronales



**Figura 2.2: Arquitectura de subsunción [20]**

### 2.2.3. Arquitecturas Híbridas

Estas arquitecturas pretenden combinar aspectos deliberativos y reactivos. Los más claros exponentes de estas arquitecturas son PRS, Touring Machines e Interrap. Estas arquitecturas combinan módulos reactivos con módulos deliberativos. Los módulos reactivos se encargan de procesar los estímulos que no necesitan deliberación, mientras que los módulos deliberativos determinan qué acciones deben realizarse para satisfacer los objetivos locales y cooperativos de los agentes.[20]



**Figura 2.3: Arquitectura Híbrida (Interrap) [20]**

## 2.3. Sistemas Multiagente

Para lograr la resolución de problemas que un agente por sí sólo no puede resolver, se llega a la creación de los sistemas multiagente (SMA).

Una de las definiciones formales para estos sistemas fue formulada por Weiss [4], que dice:

*“Un sistema multi-agente es una red de problem-solvers que trabajan conjuntamente para encontrar respuestas a problemas que van más allá de las capacidades o conocimiento individuales de cada entidad.”*

Algunas de las características que posee un sistema SMA son:

- Cada agente es incapaz de resolver el problema por sí solo.
- El control del sistema es descentralizado.
- Los datos son dispersos.
- La comunicación es asíncrona.

Según [6], las motivaciones detrás del creciente interés en la investigación centrada en los SMA son:

- Resolver problemas demasiado grandes para ser resueltos por un solo agente centralizado, ya sea por sus limitados recursos o por el cuello de botella que se puede crear, pudiendo fallar en momentos críticos.
- Permitir la interconexión e interoperación entre los distintos sistemas legados. Todo esto para no caer en costos demasiado altos en actualizaciones de estos sistemas, por lo que se utilizan agente cooperativos para ser explotados por otras piezas de software, y evitar la reescritura de código.
- Proveer soluciones a problemas que podrían ser resueltos por una sociedad de agentes autónomos que interactúan entre sí.
- Proveer soluciones que utilicen eficientemente las fuentes de información que se encuentran en un ambiente distribuido.
- Proveer soluciones en que la experiencia se encuentra distribuida.
- Mejorar el rendimiento en las dimensiones de la eficiencia computacional, fiabilidad, extensibilidad, mantenibilidad, respuesta a errores, flexibilidad y reuso.

### **2.3.1. Dominios de Aplicación**

Los dominios de aplicación más prometedores deben poseer las siguientes características:

- **Distribución Inherente:** Los datos e información del problema a solucionar deben ser esencialmente distribuidos:
  - **Espacial:** Los datos aparecen en diversas localizaciones geográficas.
  - **Temporal:** La información genera en diversas horas.
  - **Semántico:** Datos clusterizados que requieren distintos lenguajes y ontologías para su acceso y uso.
  - **Funcional:** Datos clusterizados que requieren capacidades perceptivas, actuativas y cognoscitivas para su acceso y uso.
- **Complejidad Inherente**
  - Problema demasiado grande para ser solucionado de una manera tradicional (un solo sistema centralizado), debido a limitaciones de hardware y software en esa escala.
  - El enfoque centralizado no se prefiere debido a su alta dificultad, costo y tiempo consumido comparado con la baja flexibilidad y confiabilidad obtenidas.

## **2.4. Estándares Multiagente**

### **2.4.1. Arquitectura FIPA**

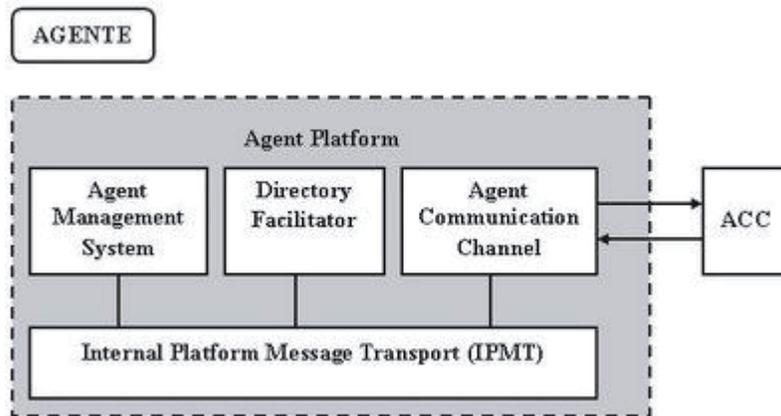
Como introducción, se debe señalar que FIPA [21] es un consorcio industrial fundado en 1996, formado por varias decenas de compañías de telecomunicaciones e informática. Este consorcio está dedicado a la estandarización de agentes, donde sus principales objetivos son:

- Acelerar el desarrollo de las tecnologías de agentes inteligentes mediante la producción de especificaciones acordadas intencionalmente.
- Producir estándares de software para agentes y multiagentes interactivos heterogéneos.
- Ser usados por la industria y organizaciones que desarrollan agentes.

Las especificaciones creadas por FIPA fueron evolucionando desde 1997 hasta el año 2000, en el que se llegó a la arquitectura abstracta (el núcleo de FIPA), la que define como dos agentes se pueden localizar y comunicar.

A la hora de definir la plataforma de agentes (Agent Platform) FIPA ha seguido una serie de principios, como definir solo el comportamiento externo (interfaz), dejando las

decisiones de diseño a los equipos de desarrollo. También, conseguir un sistema totalmente abierto de tal forma que los sistemas heterogéneos puedan interactuar a nivel de sociedad de agentes. Y por último, establece el modelo lógico referente a la creación, destrucción, registro, localización y comunicación de los agentes.



**Figura 2.4: Modelo de Arquitectura FIPA. [21]**

Los componentes de la plataforma son los siguientes:

- Sistema de Gestión de Agentes (AMS): Permite el registro de nuevos agente a la plataforma, proporcionando además un servicio de páginas blancas para los agentes, por lo que estos pueden conocer el nombre y dirección de cualquier otro agente.
- Facilitador de Directorio (DF): Es un servicio de páginas amarillas, por lo que los agentes pueden encontrar a otros pertenecientes a la plataforma según los servicios que estos ofrecen.
- Sistema de Transporte de Mensajes (STM): Permite gestionar la comunicación entre los agentes de distintas plataformas.

## 2.4.2. Infraestructura de Agentes

Para hacer posible la interacción de los agentes es necesaria la llamada infraestructura de agentes, que es quien debe proporcionar las regulaciones que los agentes siguen para comunicar y entenderse, permitiendo así compartir el conocimiento. Estas regulaciones son de 3 tipos:

- Ontologías: Entrega un significado común de los conceptos para que los agentes convengan y se entiendan.
- Lenguajes de Comunicación (ACL's): Permiten especificar las formas de comunicación (protocolos) entre agentes, logrando que los agentes puedan intercambiar mensajes que sean entendibles.

- Protocolos de Interacción (AIP's): Entregan convenciones para las interacciones entre agentes, permitiéndoles tener conversaciones, es decir, intercambios estructurados de mensajes.

### 2.4.3. Ontologías

El objetivo de la ontología es el estudio de todas las clases de entidades que conforman nuestro mundo, de la existencia en sí. Así, llamaremos una ontología a un catálogo de los tipos de cosas que asumimos que existen en un dominio de interés, desde la perspectiva de una persona que usa un lenguaje con el propósito de hablar acerca de ese dominio. Aunque el término procede de la Filosofía (ontología: parte de la filosofía que estudia el ente como tal), en los últimos tiempos ha sido reutilizado en el campo de la Inteligencia Artificial y en concreto en los Sistemas Multiagente. Así podemos definir la ontología de la siguiente forma [7]:

*“Una ontología define los términos y relaciones básicas que forman parte del vocabulario de una determinada área, así como las reglas para combinar términos y relaciones para definir extensiones del vocabulario”*

Las ontologías proveen una forma de describir los conceptos de una forma no ambigua, siendo un nivel apropiado de formalización semántica que muchos lenguajes no tienen.

A continuación se muestran los lenguajes mayormente utilizados para describir una ontología:

- 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, DARPA project (Defence Advanced Research Project Agency)
- OIL: Ontology Interchange Language, OntoKnowledge project (European IST)
- DAML+OIL: W3C.
- OWL: Web Ontology Language, W3C.

## 2.4.4. Lenguajes de Comunicación de Agentes

Uno de los aspectos más importantes a la hora de definir un sistema es la forma en que sus integrantes se comunicarán. Para esto, se necesita un lenguaje que sea entendido por todos, que permita saber cómo reaccionar frente a cada mensaje recibido.

Dentro del estudio de la comunicación se encuentran tres aspectos:

- **Sintaxis:** Define la estructura de los símbolos en la comunicación.
- **Semántica:** Denotación que entregan los símbolos.
- **Pragmática:** Cómo se interpretan estos símbolos.

Así, se puede definir el significado como la suma de la semántica y la pragmática, ya que al igual que en la comunicación humana (de la cual se basa la comunicación entre agentes), dos mensajes que compartan la misma semántica pueden tener significados distintos dependiendo del contexto en que sean realizados.

Hay tres aspectos importantes que se deben distinguir en la comunicación humana:

- **Locución:** Corresponde a la uterancia de lo comunicado (el sonido o lo escrito)
- **Ilocución:** El significado que el emisor intentaba entregar.
- **Perlocución:** Es el efecto causado por la locución.

En la comunicación humana, muchas veces se producen diferencias entre lo que el emisor quiso entregar, y lo que realmente entendió el receptor, muchas veces relacionado con el tono de voz o las circunstancias en que un mensaje fue realizado. Para evitar equivocaciones como estas, se ha implementado en la comunicación de agente los *performatives*, que define el tipo de acto comunicativo del mensaje, siendo este el único elemento obligatorio en un mensaje FIPA ACL. En la Tabla 2.1 se pueden observar los elementos que pueden estar presentes en un mensaje ACL.

**Tabla 2.1: Elementos de mensaje FIPA ACL.**

<b>Elemento</b>	<b>Categoría del Elemento</b>
performative	Tipo de acto comunicativo
sender	Participante en la comunicación
receiver	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

También existe un estándar llamado KQML (“Knowledge Query and Manipulation Language”), el cual es un lenguaje y protocolo para el intercambio de información y conocimiento, surgido de los trabajos para el desarrollo de técnicas y metodologías para la construcción de bases de conocimiento a gran escala que sean compartidas y reusables. [7]

Los elementos de un mensaje KQML son mostrados en la Tabla 2.2:

**Tabla 2.2: Parámetros reservados en un mensaje KQML**

<b>Palabra clave</b>	<b>Descripción</b>
:sender	emisor (actual) de la performativa
:receiver	receptor (actual) de la performativa
:from	el origen (emisor virtual) de la performativa de :content cuando se hace un forward
:to	el destinatario final (receptor virtual) de la performativa de :content cuando se hace un forward
:in-reply-to	etiqueta esperada en la respuesta al mensaje previo (la misma de :reply-with del mensaje previo)
:reply-with	etiqueta esperada en la respuesta al actual mensaje
:language	lenguaje de representación en el que está escrito :content
:ontology	nombre de la ontología asumida en :content
:content	información con la que la performativa expresa una actitud

## 2.4.5. Protocolos de Interacción

Los protocolos de interacción se definen como patrones de comunicación con una secuencia permitida de mensajes entre agentes, junto a las restricciones al contenido de estos. Estos protocolos corresponden a una estructura de alto nivel para establecer las relaciones entre los agentes.

Los protocolos definidos por FIPA se describen a continuación:

**Tabla 2.3: Protocolos de comunicación FIPA**

<b>Nombre protocolo</b>	<b>Descripción</b>
Request	A un agente se le pide realizar una acción
Request when	A un agente se le pide realizar una acción al producirse un acontecimiento
Query	A un agente se le pide un informe
Contract Net	Un agente toma el rol de administrador y desea que otros agentes realicen una tarea. El administrador recibe las propuestas u elige la mejor.
Brokering	Un agente actúa de intermediario entre 2 agentes, realizando las funciones de uno o reenviando las peticiones a este.
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	Similar al brokering, pero las respuestas en este caso van directamente al solicitante.
Propose	El agente iniciador propone la realización de una tarea a otros agentes, donde estos pueden aceptar o no.
Subscribe	Un agente pide ser notificado si cierta condición es verdadera.
Iterated Contract Net	Extensión de Contract Net, con múltiples iteraciones de licitación.

## 2.5. Metodologías de desarrollo

Para el desarrollo de sistemas multiagente, se necesitan elaborar estrategias de desarrollo que promuevan buenas prácticas que puedan ser adaptables a cada proyecto en particular. El objetivo es convertir el desarrollo de software en un proceso formal, con resultados predecibles, que permitan obtener un producto final de alta calidad, que satisfaga las necesidades y expectativas para lo que fue creado el proyecto.

El desarrollo de sistemas multiagentes no escapa a estos desafíos, por lo que también se han creado distintas metodologías de desarrollo. Algunas de ellas serán descritas a continuación.

### **2.5.1. BDI**

Las arquitecturas BDI [8] se inspiran en un modelo cognitivo del ser humano. Según esta teoría, los agentes utilizan un modelo del mundo, una representación de cómo se les muestra el entorno. El agente recibe estímulos a través de sensores ubicados en el mundo. Estos estímulos modifican el modelo del mundo que tiene el agente (representado por un conjunto de creencias). Para guiar sus acciones, el agente tiene Deseos. Un deseo es un estado que el agente quiere alcanzar a través de intenciones. Éstas son acciones especiales que pueden abortarse debido a cambios en el modelo del mundo.

Para especificar el sistema de agentes, se emplean un conjunto de modelos que operan a dos niveles de abstracción: externo e interno. Primero, desde un punto de vista externo, un sistema se modela como una jerarquía de herencia de clases de agentes, de la que los agentes individuales son instancias. Las clases de agente se caracterizan por su propósito, sus responsabilidades, los servicios que ejecutan, la información acerca del mundo que necesitan y las interacciones externas. Segundo, desde un punto de vista interno, se emplean un conjunto de modelos (modelos internos) que permiten imponer una estructura sobre el estado de información y motivación de los agentes y las estructuras de control que determinan su comportamiento (creencias, objetivos y planes en este caso).

En esta metodología, la integración con el ciclo de vida de software es reducida. Los autores proponen una serie concreta de pasos para generar los modelos. Estos pasos se repiten haciendo que los modelos, que capturan los resultados del análisis, sean progresivamente elaborados, revisados y refinados. Además, el refinamiento de los modelos internos conlleva la realimentación de los modelos externos. [8]

### **2.5.2. MaSE**

MaSE (Multi-agent Systems Software Engineering) [8] parte del paradigma orientado a objetos y asume que un agente es sólo una especialización de un objeto. La especialización consiste en que los agentes se coordinan unos con otros vía conversaciones y actúan proactivamente para alcanzar metas individuales y del sistema.

En MaSE los agentes son sólo una abstracción conveniente, que puede o no poseer inteligencia. En este sentido, los componentes inteligentes y no inteligentes se gestionan igualmente dentro del mismo almacén.

El proceso de desarrollo en MaSE es un conjunto de pasos, la mayoría de los cuales se ejecutan dentro de la herramienta que soporta MaSE, AgentTool.

El análisis en MaSE consta de tres pasos: capturar los objetivos, capturar los casos de uso y refinar roles. Como productos de estas etapas se esperan: diagramas de objetivos, que representan los requisitos funcionales del sistema; diagramas de roles, que identifica roles, tareas asociadas a roles y comunicaciones entre roles y entre tareas; y casos de uso, mostrados no como diagrama sino como una enumeración de los casos de uso considerados con la posibilidad de usar diagramas de secuencia para detallarlos.

El diseño consta de cuatro pasos: crear clases de agentes, construir conversaciones, ensamblar clases de agentes y diseño del sistema. Como productos de estas etapas, MaSE espera: diagramas de clases de agentes, que enumeran los agentes del sistema, roles jugados e identifican conversaciones entre los mismos; descomposición del sistema (agente) en subsistemas (componentes del agente) e interconexión de los mismos (definición de la arquitectura del agente mediante componentes); diagramas UML de despliegue para indicar cuántos agentes habrá en el sistema y de qué tipo.

Las comunicaciones entre diferentes elementos (componente-componente, agente-agente, rol-rol, tarea-tarea) se refieren al envío de estímulos desde una máquina de estados a otra. En el caso de las tareas, a estas máquinas las denominan en MaSE diagramas de tareas concurrentes.[8]

### **2.5.3. GAIA**

GAIA [8] pretende ayudar al analista a ir sistemáticamente desde unos requisitos iniciales a un diseño que, según los autores, esté lo suficientemente detallado como para ser implementado directamente.

En GAIA se entiende que el objetivo del análisis es conseguir comprender el sistema y su estructura sin referenciar ningún aspecto de implementación. Esto se consigue a través de la idea de organización. Una organización en GAIA es una colección de roles, los cuales mantienen ciertas relaciones con otros y toman parte en patrones institucionalizados de interacción con otros roles. Los roles agrupan cuatro aspectos: responsabilidades del agente, los recursos que se le permite utilizar, las tareas asociadas e interacciones.

GAIA propone trabajar inicialmente con un análisis a alto nivel. En este análisis se usan dos modelos, el modelo de roles para identificar los roles clave en el sistema junto con sus propiedades definitorias y el modelo de interacciones que define las interacciones mediante una referencia a un modelo institucionalizado de intercambio de mensajes, como el FIPA-Request. Tras esta etapa, se entraría en lo que GAIA considera diseño a alto nivel. El objetivo de este diseño es generar tres modelos: el modelo de agentes que define los tipos de agente que existen, cuántas instancias de cada tipo y qué papeles juega cada agente, el modelo de servicios que identifica los servicios (funciones del agente) asociados a cada rol, y un Modelo de conocidos, que define los enlaces de comunicaciones que existen entre los agentes.

A partir de aquí, los autores de GAIA proponen aplicar técnicas clásicas de diseño orientado a objetos. Sin embargo, GAIA declara que queda fuera de su ámbito. Esta

metodología sólo busca especificar cómo una sociedad de agentes colabora para alcanzar los objetivos del sistema, y qué se requiere de cada uno para lograr esto último.[8]

## 2.5.4. PASSI

PASSI (Process for Agent Societies Specification and Implementation) [9] es una metodología paso-a-paso y de requerimientos-a-código para el diseño y desarrollo de sociedades multiagente. La metodología integra modelos de diseño y conceptos tanto de enfoques de la ingeniería de software orientada a objetos como de la inteligencia artificial.

El proceso de diseño se compone de 5 modelos (ver Figura 2.5): El modelo de requerimientos del sistema (System Requirements Model); el modelo de la sociedad de agentes (Agent Society Model), que es un modelo de los agentes involucrados en la solución en términos de sus roles, interacciones sociales, dependencias, y ontología; el modelo de implementación de agentes (Agent Implementation Model) es un modelo de la arquitectura de solución en términos de clases y métodos (a 2 niveles de abstracción: individual y multi agente); el modelo del código (Code Model) es un modelo de la solución a nivel de un lenguaje específico y el modelo de despliegue (Deployment Model) es un modelo de la distribución de los elementos del sistema (agentes) a través de las unidades de procesamiento de hardware, y sus movimientos a través de las diferentes plataformas disponibles.[9]

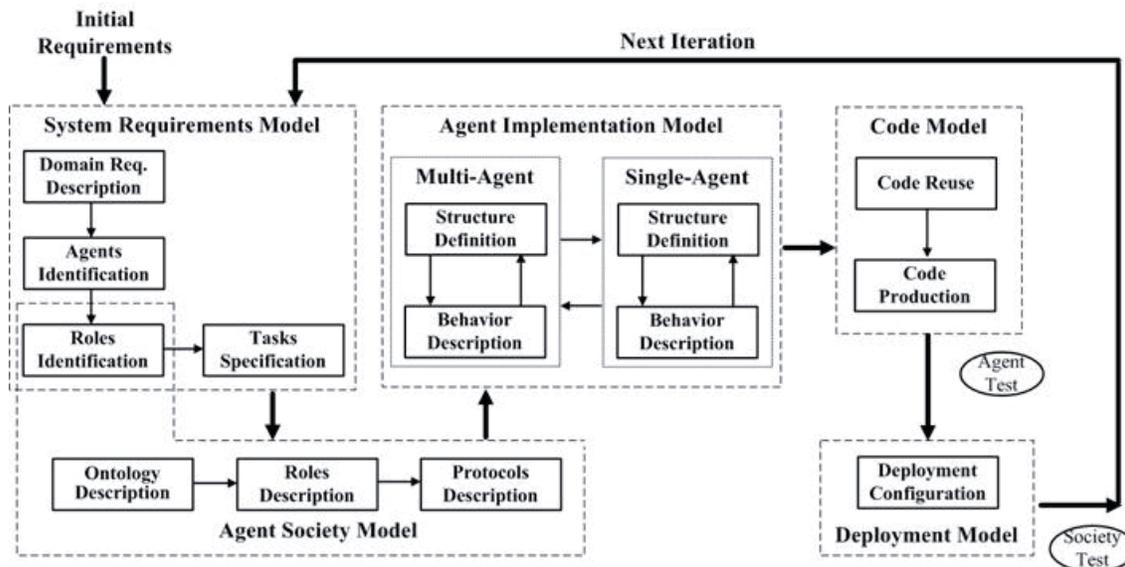


Figura 2.5: Modelos y fases de PASSI [9]

Los modelos y fases de PASSI son:

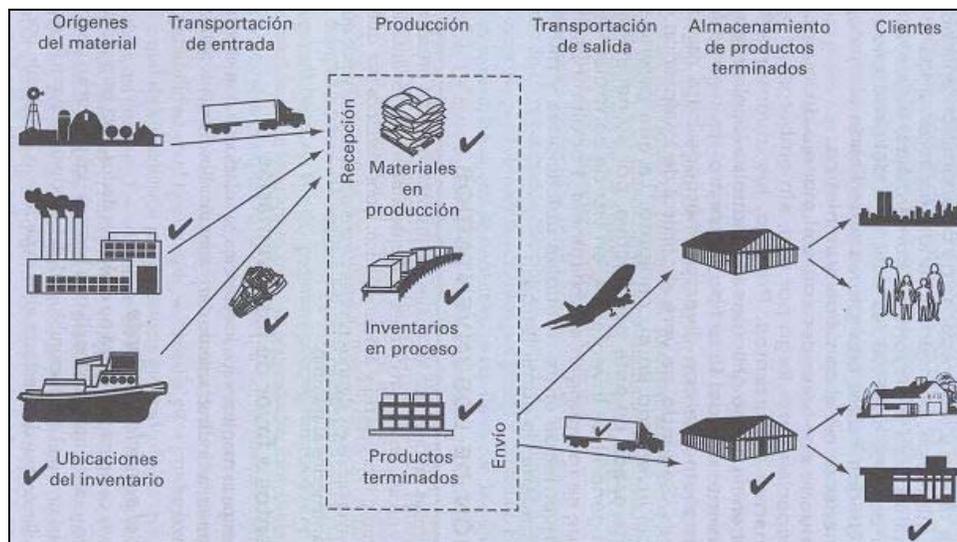
- Modelo de Requerimientos del Sistema: Un modelo antropomórfico del sistema en términos de agencia y propósito. Desarrollar este modelo involucra cuatro pasos:
  - Descripción del Dominio (D.D.): Una descripción funcional del sistema usando diagramas de casos de uso, obteniéndose una descripción de los requerimientos.

- Identificación de Agentes (A.Id.): separación de las responsabilidades en agentes, representados por paquetes UML estereotipados.
  - Identificación de Roles (R.Id.): Utiliza diagramas de secuencia para explorar las responsabilidades de cada agente a través de escenarios específicos para cada rol.
  - Especificación de Tareas (T.Sp.): Especificación a través de diagramas de actividad de los comportamientos de cada agente.
- Modelo de Sociedad de Agentes: Un modelo sobre las interacciones sociales y dependencias entre los agentes que participan en la solución. El desarrollo de este modelo involucra tres pasos que se agregan al modelo anterior.
    - Descripción de la Ontología (O.D.): Utiliza diagramas de clases y restricciones OCL para describir el conocimiento atribuido a los agentes y la pragmática de sus interacciones.
    - Descripción de Roles (R.D.): Utiliza diagramas de clases para mostrar los distintos roles que puede tener un agente, las tareas involucradas en cada rol, las capacidades de comunicación y las dependencias inter-agentes.
    - Descripción del Protocolo (P.D.): Utiliza diagramas de secuencia para especificar la gramática de cada protocolo de comunicación en términos de los performativos de la teoría del habla.
- Modelo de Implementación de Agentes: Un modelo de la arquitectura de solución en términos de las clases y métodos. El desarrollo de este modelo incluye los siguientes pasos:
    - Definición de la estructura de los agentes (A.S.D.): Utiliza diagramas de clases para describir la estructura de las clases de agentes.
    - Descripción del Comportamiento de los Agentes (A.B.D.): Utiliza diagramas de actividad o diagramas de estado para describir el comportamiento de los agentes individualmente.
- Modelo de Código: Un modelo de la solución a nivel de código, que tiene los siguientes pasos:
    - Reutilización de Código (C.R.): Una librería de clases y diagramas de actividad con el código reusable asociado.
    - Completitud de Código (C.C.): Código fuente del objetivo del sistema.
- Modelo de Despliegue: Un modelo con la distribución de las partes del sistema a través de las unidades de procesamiento del hardware, y su migración entre las unidades de procesamiento. Involucra sólo un paso:
    - Configuración de Despliegue (D.C.): Utiliza diagramas de despliegue para describir la ubicación de los agentes en las unidades de proceso disponibles y todas las restricciones de migración y movilidad. [5]

## 3. Teoría de Inventarios

### 3.1. Definición

Los inventarios [11] son acumulaciones de materias primas, provisiones, componentes, trabajo en proceso y productos terminados que aparecen en numerosos puntos a lo largo del canal de producción y de logística de una empresa. Como se muestra en la Figura 3.1.



**Figura 3.1: Los inventarios se sitúan en cada nivel (o escalón) de la cadena de suministro[11]**

Los inventarios se hayan con frecuencia en lugares como almacenes, patios, pisos de las tiendas, equipos de transporte y en los estantes de las tiendas minoristas. Tener estos inventarios disponibles puede costar, al año, entre 20 y 40% de su valor. Por lo tanto, administrar cuidadosamente los niveles de inventario tiene un buen sentido económico. Aunque se ha avanzado mucho para reducir los inventarios mediante diferentes sistemas, como el sistema justo a tiempo (just-in time), la comprensión del tiempo, la respuesta rápida y las prácticas de colaboración aplicadas en toda la cadena de suministro, la inversión anual en inventarios realizada por fabricantes, minoristas y mayoristas comerciales, representa un importante costo dentro de la economía de un país [11].

### 3.2. Evaluación de los inventarios

Hay numerosas razones por las cuales los inventarios están presentes en un canal de suministros: aun así, en años recientes, el mantenimiento de inventarios ha sido totalmente criticado como innecesario y antieconómico. Se entregan algunas razones de por qué una empresa pudiera querer inventarios en algunos niveles de sus operaciones, y por qué esa misma empresa querría mantenerlos al mínimo.

### **3.2.1. Argumentos a favor de los inventarios**

Las razones para mantener los inventarios se relacionan con el servicio al cliente o para costear economías indirectamente derivadas de ellos. Consideremos brevemente algunas de estas razones.[11]

- Mejorar servicio al cliente: Los sistemas de operación no están diseñados para responder, de manera instantánea, a los requerimientos que los clientes hacen de los productos o servicios. Los inventarios suministran un nivel de disponibilidad del producto o servicio que, cuando se localiza cerca del cliente, puede satisfacer altas expectativas del cliente por la disponibilidad del producto.
- Reducir costos: Aunque mantener inventarios tiene un costo asociado, su uso puede reducir indirectamente los costos de operación de otras actividades de la cadena de suministro, que podrían más que compensar el costo de manejo de inventarios. Dicho esto, se puede ahorrar en economías de producción, transporte, y comprar una mayor cantidad en tiempo que cierto producto se encuentra en un precio más bajo que en otras épocas.

### **3.2.2. Argumentos en contra de los inventarios**

- Primero, los inventarios son considerados como pérdidas. Absorben el capital que podría estar disponible para mejor uso de otra manera, como mejorar la productividad o la competitividad. Además, no contribuyen con ningún valor directo a los productores de la empresa, aunque almacenan valor.
- Segundo, pueden enmascarar problemas de calidad. Cuando ocurren problemas de calidad, reducir los inventarios existentes para proteger la inversión de capital es, a menudo, la consideración principal.
- Por último, el uso de inventarios promueve una actitud aislada del canal de suministros como un todo. Con los inventarios, a menudo, se aísla una etapa del canal de otra.

## **3.3. Tipos de Inventarios**

Los inventarios pueden clasificarse en cinco formas.

1. Pueden clasificarse en ductos. Estos son los inventarios en tránsito entre los niveles del canal de suministros. Cuando el movimiento es lento o sobre grandes distancias, o ha de tener lugar entre muchos niveles, la cantidad de inventario en ductos puede exceder al que se mantiene en los puntos de almacenamiento.

2. Se pueden mantener existencias para especulación, pero todavía son base del inventario que debe manejarse. Las materias primas, como cobre, oro y plata se compran tanto para especular con el precio como para satisfacer los requerimientos de la operación.
3. Las existencias pueden ser de naturaleza regular o cíclica. Estos son los inventarios necesarios para satisfacer la demanda promedio durante el tiempo de entre reaprovisionamientos sucesivos.
4. El inventario puede crearse como protección contra la variabilidad en la demanda de existencias y el tiempo total de reaprovisionamiento. Esta medida extra de inventario, o existencias de seguridad, es adicional a las existencias regulares que se necesitan para satisfacer la demanda promedio y las condiciones del tiempo total promedio.
5. Cuando se mantiene durante un tiempo, parte del inventario se deteriora, llega a caducar, se pierde o es robado. Dicho inventario se refiere como existencias obsoletas, stock muerto o perdido. Cuando los productos son de alto valor, perecederos o pueden ser robados fácilmente, deben tomarse precauciones especiales para minimizar la cantidad de dicho stock.

### **3.4. Objetivos del Inventario**

El manejo del inventario implica equilibrar la disponibilidad del producto (o servicio al cliente), por una parte, con los costos de suministrar un nivel determinado de disponibilidad del producto, por la otra.

#### **3.4.1. Disponibilidad del producto**

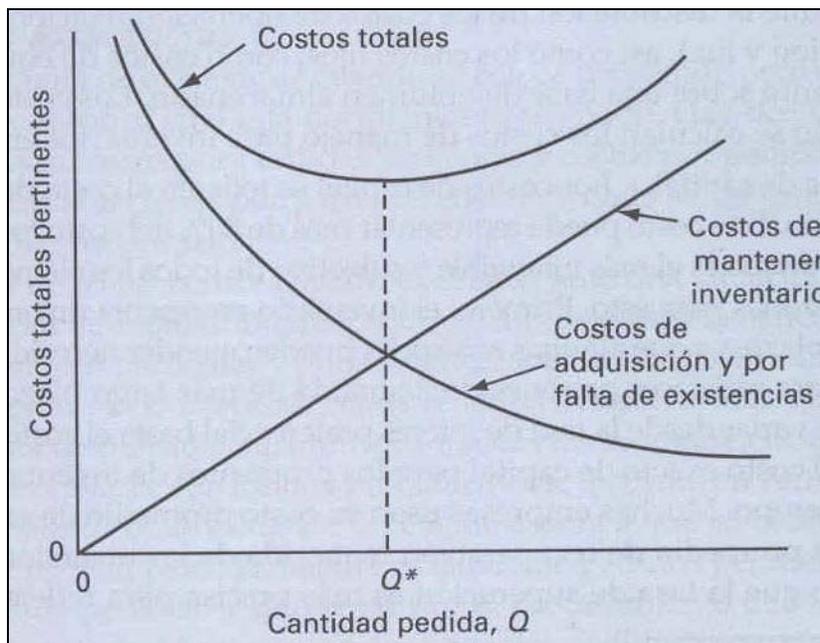
El principal objetivo del manejo de inventarios es asegurar que el producto esté disponible en el momento y en las cantidades deseadas. Normalmente, esto se basa en la probabilidad de la capacidad de cumplimiento a partir del stock actual. A esta probabilidad, llamada nivel de servicio [11], puede definirse para un artículo como

$$\text{Nivel de servicio} = 1 - \frac{\text{Número de unidades agotadas anualmente}}{\text{Demanda anual total}}$$

El nivel de servicio se expresa como un valor entre 0 y 1. Dado que un nivel de servicio objetivo está típicamente especificado, la tarea es controlar el número esperado de unidades agotadas. Para el cálculo del nivel de servicio de más de un producto, se debe multiplicar el nivel de cada uno de estos.

### 3.4.2. Costos pertinentes

Para determinar la política de inventarios son importantes tres clases generales de costos: costos de adquisición, costos de manejo y costos por falta de existencias. Estos costos están en conflicto, o en equilibrio entre sí. Para determinar la cantidad de pedidos de un artículo por reabastecer en un inventario, en la Figura 3.2 se muestran estos efectos relevantes en equilibrio.[11]



**Figura 3.2: Equilibrio de los costos pertinentes del inventario con la cantidad de pedido[11]**

1. Costos del artículo: Se refieren al precio de compra de algún artículo que la empresa adquiera o ésta produzca. Para bienes comprados, el precio total incluye el precio de lista, costos de transporte y envío, impuestos, y aranceles. En caso de artículos manufacturados, incluyen el costo de materias primas, mano de obra y gastos de distribución. Pueden ser constantes, o se pueden ofrecer con un descuento que depende del volumen del pedido.
2. Costos de colocación del pedido: Son los ocasionados por el transporte de un pedido de artículos. Abarcan actividades de compra, preparación de especificaciones y documentos, órdenes de compra, seguimiento a los proveedores e inspección de pedidos cuando llegan.
3. Los costos de mantenimiento son los gastos en que se incurre al mantener inventarios, por Ej. alquiler, electricidad, impuestos, pérdidas, obsolescencia, primas de seguros y costos de mano de obra.
4. Los costos de agotamiento (escasez) de existencias se causan cuando la empresa no puede satisfacer por completo el pedido de un cliente. La compañía pierde el margen

de aportación de esa venta y puede perderlo en ventas futuras. En algunas ocasiones debe pagarse una sanción.

### **3.5. Tipos de Demanda**

Las principales clasificaciones para la demanda son:

- Continua o Discreta: La unidad de medida de la demanda puede variar según el entorno y la presentación del artículo concreto (unidades, centenas, litros, kilogramos, etc.)
- Determinística o Probabilística: Hay casos en que la demanda futura se supone perfectamente conocida; otras veces se supone que los valores de la demanda son aleatorios.
- Dependiente o Independiente: La demanda de componentes dependerá de la demanda de productos finales, mientras que la de estos últimos se considerará independiente.
- Homogénea o Heterogénea: La demanda es homogénea si su valor es constante en el tiempo.
- Diferida o Perdida: Si no se satisface la demanda (ruptura de inventario), a veces será posible diferir la entrega.

### **3.6. Políticas de control de inventarios**

Una política de control de inventarios es un procedimiento llevado a cabo para auxiliar a los responsables a responder a las siguientes preguntas:

- ¿Cuánto se debe ordenar?
- ¿Cuándo se debe realizar un pedido?

Las respuestas a estas preguntas se pueden responder mediante distintos modelos de control de inventarios, donde los más importantes serán vistos a continuación.

#### **3.6.1. Control de inventarios por incrementos (push)**

Este método es apropiado cuando las cantidades de producción o de compra exceden los requerimientos a corto plazo de los inventarios a los que tienen que enviarse dichas cantidades. Este tipo de control es un método razonable para el control de inventarios donde la producción o la compra es la fuerza dominante en el momento de determinar las cantidades de reaprovisionamiento.

### 3.6.2. Control básico de inventarios por demanda (pull)

Este método otorga bajos niveles de inventario en los puntos de abastecimiento, debido a su respuesta a las condiciones particulares de la demanda y de costo de cada punto de abastecimiento. Bajo este tipo de control, se pueden realizar varias modificaciones, dependiendo del número de pedidos (único o repetitivo) y el conocimiento de la demanda (determinada o incierta), entre otros factores.

#### Pedido único

Esta variación puede ocurrir generalmente por la temporalidad en la que se necesitan los productos, ya sea por el perecimiento de estos (frutas y verduras de una determinada época), o por el momento en que se venden (ropa de moda), por lo que no se necesita realizar pedidos continuos. Deseamos determinar el tamaño que debe tener este pedido.

Para hallar el tamaño más económico de pedido ( $Q^*$ ) [11], debemos hallar el punto donde la ganancia marginal de la siguiente unidad vendida es igual a la pérdida marginal de no vender la siguiente unidad. La ganancia marginal por unidad vendida es

$$\text{Ganancia} = \text{Precio por unidad} - \text{Costo por unidad}$$

La pérdida incurrida por unidad por no venderla es

$$\text{Pérdida} = \text{Costo por unidad} - \text{Valor de desecho por unidad}$$

Considerando la probabilidad de un número dado de unidades que se venden, las ganancias y las pérdidas esperadas se equilibran en este punto. Es decir,

$$CP_n (\text{pérdida}) = (1 - CP_n) (\text{Ganancia})$$

Donde  $CP_n$  representa la frecuencia acumulada de vender al menos  $n$  unidades del producto. Resolviendo la expresión anterior para  $CP_n$  tenemos:

$$CP_n = \frac{\text{Ganancia}}{\text{Ganancia} + \text{Pérdida}}$$

Esto significa que deberíamos continuar incrementando la cantidad de pedido hasta que la probabilidad acumulada de vender unidades adicionales iguale la relación de ganancia / (ganancia + pérdida).

#### Pedidos repetitivos

Los pedidos de reaprovisionamiento de inventarios se repiten en el tiempo y pueden suministrarse completos de manera instantánea, o bien, los artículos de los pedidos pueden suministrarse en el tiempo. Se ilustran ambos casos.

## Reabastecimiento instantáneo

Cuando la demanda es continua y la tasa es esencialmente constante, el control de los inventarios se realiza especificando, 1) la cantidad que se usará para reaprovisionar el inventario según una base periódica, y 2) la frecuencia de reaprovisionamiento del inventario. La fórmula básica de la cantidad económica de pedido (CEP, o EOQ) sirve como base para muchas políticas de inventario del método de demanda (pull) usadas actualmente en la práctica.

La fórmula básica CEP se desarrolla a partir de una ecuación de costo total que involucra el costo de adquisición y el costo de manejo de inventario. Se expresa como:

Costo total = costo de adquisición + costo de manejo

$$TC = \frac{D}{Q}S + \frac{ICQ}{2}$$

Donde

TC = Costo pertinente total y anual del inventario

Q = Tamaño del pedido para reaprovisionar el inventario, en unidades

D = Demanda anual de artículos, que ocurre a una tasa cierta y constante en el tiempo, en unidades/año

S = Costo de adquisición, \$/período

C = Valor del artículo manejado en inventario, en \$/unidad

I = Costo de manejo como porcentaje del valor del artículo, %/año

El término  $D/Q$  representa en número de veces al año que se coloca un pedido de reaprovisionamiento en su fuente de suministro. El término  $Q/2$  es la cantidad promedio del inventario disponible.

Como Q varía de tamaño, un costo sube cuando el otro baja. Puede demostrarse matemáticamente que existe una cantidad óptima de pedido ( $Q^*$ ) cuando los dos costos están en equilibrio y resulta el costo total mínimo. La fórmula para esta CEP es

$$Q^* = \sqrt{\frac{2DS}{IC}}$$

El tiempo óptimo entre los pedidos es, por lo tanto,

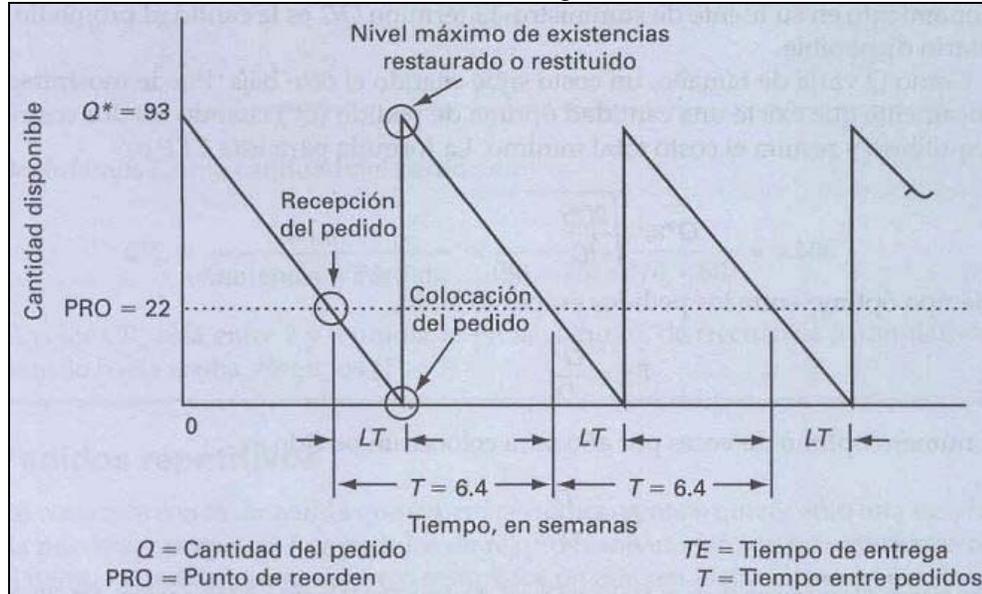
$$T^* = \frac{Q^*}{D}$$

Y el número óptimo de veces por año para colocar un pedido es

$$N = \frac{D}{Q^*}$$

### Tiempo de entrega de reaprovisionamiento

Usando esta fórmula como parte de un procedimiento de control básico de inventarios, vemos que surge un patrón como forma de diente de sierra de reducción y reaprovisionamiento de inventarios, como se ve en la Figura 3.3.



**Figura 3.3: Modelo de control básico de inventarios de demanda (pull) para una parte de reaprovisionamiento.[11]**

Ahora se puede introducir la idea del *punto de reorden*, que es la cantidad a la que se debe dejar caer el inventario antes de colocar un nuevo pedido. El punto de reorden (PRO) es  $PRO = d * TE$

Donde

PRO = cantidad de punto de reorden, en unidades

d = tasa de demanda, en unidades de tiempo

TE = tiempo de entrega promedio, en unidades de tiempo

La tasa de demanda (d) y el tiempo de entrega promedio (TE) deben expresarse en la misma dimensión de tiempo.

### 3.6.3. Control avanzado de inventarios por demanda (pull)

El control avanzado de inventarios por demanda significa que se reconoce que la demanda y el tiempo de entrega no se pueden conocer con seguridad. Por lo tanto, se tiene que planear para una situación en la que no haya suficientes existencias disponibles para surtir las solicitudes de los clientes. Además de las existencias regulares que se mantiene para satisfacer la demanda promedio y el tiempo de entrega promedio, se añade una cantidad de incremento al inventario. La cantidad de estas existencias de seguridad, o amortiguador, foja el nivel de

disponibilidad de existencias suministradas a los clientes al controlar la probabilidad de que ocurra falta de existencias.

Dos métodos de control de inventarios forman la base para la mayor parte de las filosofías de manejo de tipo demanda (pull) con patrones de demanda perpetua. Estas son: 1) el método de punto de reorden, y 2) el método de revisión periódica. Los sistemas prácticos de control pueden basarse en cualquiera de estos métodos o en una combinación de ellos.

## **Método del punto de reorden con demanda incierta**

### **Hallar Q\* y PRO**

El control de inventarios por punto de reorden supone que la demanda es perpetua y actúa continuamente en el inventario para reducir su nivel. El nivel efectivo de inventario en un momento determinado del tiempo es la cantidad disponible más la cantidad de pedido, menos cualquier obligación contra el inventario, como pedidos pendientes. Existe la probabilidad de que la demanda exceda la cantidad que queda en el inventario. Esto se controla elevando o descendiendo el punto de reorden y ajustando Q\*.

En este caso, la demanda se conoce sólo al grado de una distribución de probabilidad normal. Esta demanda durante la distribución de tiempo de entrega (DDLT) tiene un punto medio de  $X'$  y una desviación estándar de  $s'_d$ . Los valores de  $X'$  y de  $s'_d$  por lo regular no se conocen de forma discreta, pero pueden estimarse fácilmente sumando la distribución de la demanda de un período único sobre la duración del tiempo de entrega. Por ejemplo, supongamos que la demanda semanal para un artículo se distribuye normalmente con una media de  $d=100$  unidades y una desviación estándar  $s'_d = 10$  unidades. El tiempo de entrega es de 3 semanas. Se desea acumular la distribución semanal de la demanda en una distribución de demanda DDLT de tres semanas. El punto medio de la distribución DDLT simplemente es la tasa de demanda  $d$  por el tiempo de entrega  $TE$ , o  $X' = d * TE = 100 * 3 = 300$ . La variación de la distribución DDLT se halla sumando las variaciones de las distribuciones de demanda semanales. Es decir  $s'^2_d = TE(s'_d)^2$ . La desviación estándar es la raíz cuadrada de  $s'^2_d$ , que es  $s'_d = s_d \sqrt{TE} = 10\sqrt{3} = 17.3$

Hallar Q\* y PRO es más bien matemáticamente complejo, sin embargo, puede hallarse una aproximación satisfactoria si determinamos primero Q\*, según la fórmula básica CEP. Luego hallamos

$$PRO = d * TE + z*(s'_d)$$

El término  $z$  es el número de desviaciones estándar desde la media de la distribución DDLT, para darnos la probabilidad deseada de tener existencias durante el tiempo de entrega. El valor de  $z$  se halla en una tabla de distribución normal.

### **Nivel promedio de inventario**

El nivel promedio de inventario para un artículo es el total de las existencias regulares más las existencias de seguridad. Es decir,

$$\begin{aligned} \text{Inventario promedio} &= \text{existencias regulares} + \text{existencias de seguridad} \\ \text{AIL} &= Q / 2 + z^*(s'_d) \end{aligned}$$

### Costo pertinente total

El costo pertinente total es útil para comparar políticas alternativas de inventarios.

Este se define como

Costo total = costo del pedido + costo de manejo, existencias regulares + costo de manejo, existencias de seguridad + costo por falta de existencias

$$TC = \frac{D}{Q}S + IC\left(\frac{Q}{2} + z s'_d\right) + \frac{D}{Q}k s'_d E_{(z)}$$

Donde k es el costo por unidad por falta de existencias.

Para explicar el costo por falta de existencias se debe decir que el término  $s'_d E_{(z)}$  representa el número esperado de unidades agotadas durante un ciclo de pedido.  $E_{(z)}$  se llama la unidad normal de pérdida integral cuyos valores están tabulados como función de la desviación normal z. Segundo, el término  $D/Q$  es el número de ciclos de pedido por período, normalmente un año. De aquí que el número de ciclos del pedido, dé el número total esperado de unidades agotadas para todo el período. Luego, multiplicándolo por el costo de falta de existencias da el costo total del período.

### Nivel de servicio

El nivel de servicio, o tasa de disponibilidad del artículo, se define como:

$$SL = 1 - \frac{(D/Q)(s'_d * E_{(z)})}{D} = 1 - \frac{s'_d * (E_{(z)})}{Q}$$

### 3.6.4. Método del punto de reorden con costos conocidos de falta de existencias

Cuando se conocen los costos de falta de existencias no es necesario asignar un nivel de servicio al cliente. Puede calcularse el equilibrio óptimo entre el servicio y el costo. Un procedimiento de cálculo iterativo se perfila así:

1. Aproximar la cantidad de pedido a partir de la fórmula básica CEP, es decir,

$$Q^* = \sqrt{\frac{2DS}{IC}}$$

2. Calcular la probabilidad de tener existencias durante el tiempo de entrega si se permite tener pedidos pendientes

$$P = 1 - \frac{QIC}{Dk}$$

O si en un período de falta de existencias se pierde la venta

$$P = 1 - \frac{QIC}{Dk + QIC}$$

Hallar  $s'_d$ . Buscar el valor de  $z$  que corresponde a  $P$  en la tabla de distribución normal. Hallar  $E(z)$  de la tabla integral de pérdida normal por unidad.

- Determinar una  $Q$  revisada a partir de la fórmula CEP modificada, que es

$$Q = \sqrt{\frac{2D[S + ks'E(z)]}{IC}}$$

- Repetir los pasos 2 y 3 hasta que no haya cambios en  $P$  o  $Q$ . Continuar.
- Calcular PRO y otras estadísticas como se desee.

### 3.6.5. Método del punto de reorden con tiempos de demanda y entrega inciertos

Las incertidumbres en el tiempo de entrega pueden extender el realismo del modelo de punto de reorden. Lo que se desea hacer es hallar la desviación estándar ( $s'_d$ ) de la distribución DDLT, basada en la incertidumbre tanto de la demanda como el tiempo de entrega. Sumar la variación de la demanda a la variación de tiempo de entrega da una fórmula revisada para  $s'_d$ , que es

$$s'_d = \sqrt{TS * s_d^2 + d^2 S_{TE}^2}$$

Donde  $S_{TE}$  es la desviación estándar del tiempo de entrega.

### 3.6.6. Modelo de revisión periódica con demanda incierta

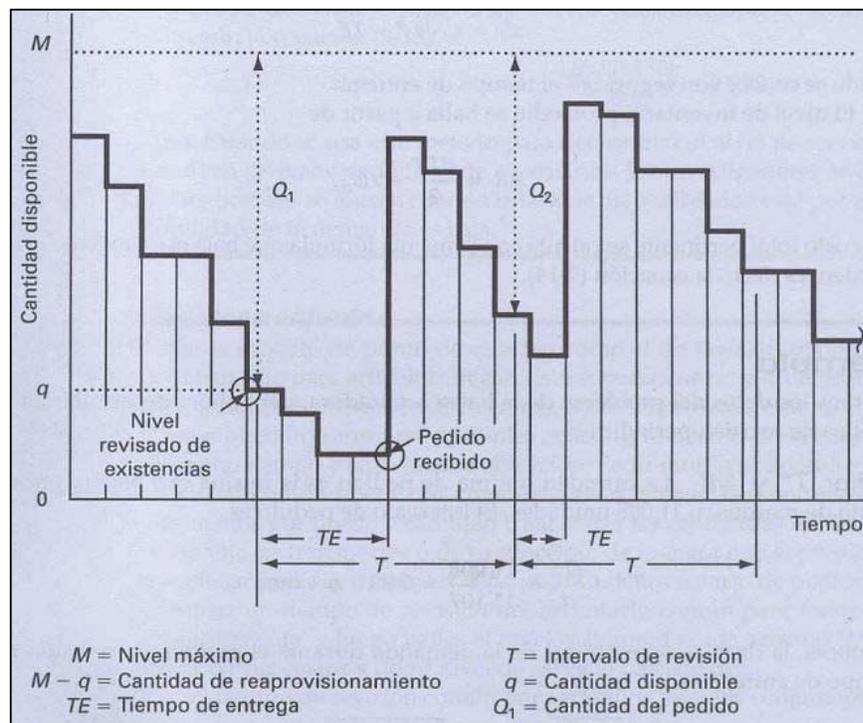
Una alternativa para el método de control del punto de reorden es el de revisión periódica. Aunque el método del punto de reorden ofrece un control preciso sobre cada artículo del inventario, y por lo tanto el costo total pertinente más bajo, tiene algunas desventajas económicas. A parte de monitoreo constante, al ser pedidos productos en distintos momentos. El control de revisión periódica da por resultado un poco más de inventario, pero los costos añadidos de su manejo pueden estar más que compensados por costos administrativos reducidos, precios más bajos o costos de adquisición más bajos. Las razones por las que se puede preferir son:

- Se usa sistema manual de contabilidad de inventarios, y es conveniente revisarlo en una agenda definida.
- Pueden ordenarse numerosos artículos de manera conjunta desde los mismos orígenes del vendedor.
- Tener ahorros significativos de transporte al solicitar diferentes artículos al mismo tiempo.

## Control de artículo único

El modelo de revisión periódica es muy parecido al modelo de punto de reorden bajo condiciones de demanda incierta. Sin embargo, una diferencia importante en el modelo de revisión periódica es que tiene que protegerse frente a las fluctuaciones de la demanda durante el intervalo de pedido y tiempo de entrega, punto que sólo son importantes las fluctuaciones de la demanda durante el tiempo de entrega en el momento de calcular las existencias de seguridad usando el método de punto de reorden.

El control de revisión periódica opera como se muestra en la Figura 3.4. Eso es, el nivel de inventario para un artículo es auditado a intervalos predeterminados ( $T$ ). La cantidad por colocar en un pedido es la diferencia entre una cantidad máxima ( $M$ ) y la cantidad disponible en el momento de la revisión. Por lo tanto, el inventario se controla estableciendo  $T^*$  y  $M^*$ .



**Figura 3.4: Control de inventario de revisión periódica con incertidumbre para un artículo.[11]**

Una buena aproximación para el intervalo de revisión óptima comienza con el modelo de control básico de inventario. Esto es

$$Q^* = \sqrt{\frac{2DS}{IC}}$$

Y el intervalo de revisión es

$$T^* = \frac{\text{Cantidad de pedido}}{\text{Demanda anual}} = \frac{Q^*}{D}$$

Después, se construye la distribución de la demanda en intervalos de un pedido más el tiempo de entrega [DD(T\* + TE)]. El punto donde la probabilidad de que se produzca falta de existencias durante el período de protección (1 - P) es igual al área bajo la curva de la distribución normal, es el punto de nivel máximo (M\*). Este punto puede calcularse como

$$M^* = d(T^* + TE) + z(s'_d)$$

Donde d(T\* + TE) es el punto medio de la distribución DD(T\* + TE), d es la tasa promedio de demanda diaria, y s'\_d es la desviación estándar de la distribución. Esta desviación se calcula ahora como

$$s'_d = s_d \sqrt{T^* + TE}$$

Donde se conoce con seguridad el tiempo de entrega.

El nivel de inventario promedio se halla a partir de

$$AIL = \frac{dT^*}{2} + z(s'_d)$$

## Pedidos conjuntos

Tanto el modelo de punto de reorden como el de revisión periódica descritos hasta ahora han sido para artículos únicos. Esto supone que cada artículo en el inventario se controla independientemente de los otros. En muchos casos, ésta no es la mejor práctica dado que pueden comprarse múltiples artículos al mismo proveedor o pueden producirse al mismo tiempo y en la misma ubicación. Pedir múltiples artículos al mismo tiempo y en el mismo pedido puede dar como resultado ganancias económicas, como calificar para descuentos por precio y cantidad o satisfacer las cantidades mínimas del vendedor, de la compañía de transportes o de producción, de manera que la política de inventario debería reflejar pedidos conjuntos. Una política de inventario de pedido conjunto implica determinar un tiempo de revisión de inventario común para todos los artículos pedidos conjuntamente, y luego hallar el nivel máximo de cada artículo (M\*) según se impone a partir de sus costos y de su nivel de servicio particulares.[11]

El tiempo de revisión común para artículos pedidos conjuntamente es

$$T^* = \sqrt{\frac{2(O + \sum_i S_i)}{I \sum_i C_i D_i}}$$

Donde O es el costo común de procurar un pedido y el subíndice i se refiere a un artículo en particular. El nivel máximo para cada artículo es

$$Mi^* = d_i(T^* + TE) + z_i(s'_d)_i$$

El costo pertinente total es

Costo total = Costo de pedido + costo de manejo de las existencias regulares + costo de manejo de las existencias de seguridad + stock de falta de existencias

$$TC = \frac{O + \sum_i S_i}{T} + \frac{TI \sum_i C_i D_i}{2} + I \sum_i C_i z_i (s'_d)_i + \frac{1}{T} \sum_i k_i (s'_d)_i (E_{(z)})_i$$

### 3.7. Control multinivel

Manejar los inventarios a través de todo el canal de distribución se vuelve un tema importante, antes que la administración en puntos de almacenamiento individuales e independientes.[11]

El cálculo de un buen control de inventario multinivel se hace usando un sistema de control de existencias base. Las bases de este sistema sirven para planear el nivel de almacenamiento en la posición del inventario más el inventario de todos los niveles inferiores de cualquier nivel en el canal de suministros. Es decir, planear el nivel de inventario para un escalón en particular no se determina a partir de la información de la demanda sólo del nivel anterior, sino más bien de la demanda del cliente final. Hay menos variabilidad de la demanda para un nivel en particular cuando la demanda final puede usarse en un proceso de planeación de inventario en un nivel superior. La característica de la demanda por todo un canal de suministros es que cuanto más se aleja el nivel de los clientes finales, mayor es la desproporción de la demanda. La planeación de inventarios basada sólo en pedidos desde el nivel anterior provoca más existencias de seguridad que si se planeara usando la demanda del cliente final.

Los minoristas atienden a los clientes finales a partir de sus inventarios y el almacén reaprovisiona las existencias de los minoristas. En un sistema base de control de existencias, los niveles de inventario de los minoristas se controlan usando un método apropiado, como el control de punto de reorden. La información de la demanda para un minorista se deriva de los clientes finales en el territorio del minorista. La posición del inventario para un minorista es la cantidad disponible más la cantidad en pedido desde el almacén.

Para subir un nivel a partir de los minoristas (el nivel del almacén), la demanda para propósitos de planeación se deriva de agregar la demanda de los clientes finales de todos los minoristas. La posición del inventario para el nivel del almacén, pero no el almacén sí mismo, es la suma del inventario de los minoristas más el inventario en el almacén y el inventario en tránsito (en pedido) hacia el almacén y desde este. El punto y las cantidades de reorden se determinan por la posición del nivel del inventario, y no por la del almacén en sí mismo. El nivel promedio de abastecimiento en el almacén se halla restando los niveles de inventario promedio de los minoristas del nivel del inventario, asumiendo que el inventario en tránsito es insignificante.

El procedimiento del sistema base de existencias para la planeación de inventarios puede continuarse para niveles adicionales dentro de la cadena de suministros. Se debe recordar planear niveles de inventario para cualquier nivel basado en la demanda final de un artículo y no en los pedidos desde el nivel anterior.

### **3.8. Sistemas multiagente en la cadena de suministro**

El enfoque de agentes tiene ventajas sobre el uso de otros enfoques convencionales en la gestión de la cadena de suministro, donde uno de sus principales desafíos es el manejo de inventarios, como son: el incremento del grado de reacción de la empresa a los requerimientos del mercado, la optimización total de la cadena de suministro, la optimización de la cadena en la asignación efectiva de recursos y la gestión dinámica de materiales e inventario, y el incremento de la efectividad del intercambio y realimentación de la información.

A continuación se muestran varios proyectos relacionados con la utilización de agentes en la cadena de suministro [12].

- DragonChain de Kimbrough, Wu y Zhong de la universidad de Pensilvania (Estados Unidos) trata de automatizar la gestión de la cadena de suministro con el objetivo de reducir el efecto látigo. El sistema actúa siguiendo el modelo del Juego de la cerveza desarrollado por el Grupo de Estudios Dinámicos del MIT y utiliza agentes que a través de algoritmos genéticos tratan de determinar la mejor política de reaprovisionamiento.
- Agent Building Shell de la Universidad de Toronto (Canada) es una biblioteca de software de clases que soportan las herramientas de implementación de agentes. La arquitectura de dichos agentes está realizada utilizando cuatro capas: capa de gestión de conocimiento, capa de ontología, de cooperación y de resolución de conflictos, y la capa de comunicación y coordinación
- MetaMorph II es una mejora de un proyecto previo denominado MetaMorph. En él los agentes forman una federación basada en mediadores que tienen dos papeles: permitir que los agentes se encuentren entre sí y coordinarlos. Estos proyectos han sido desarrollados en la universidad de Calgary (Canadá)
- NetMan (NETworked MANufacturing) Propone una estructura en red para poder conseguir sistemas de producción ágiles en un entorno dinámico. Cada empresa se divide en unidades de negocio independientes que colaboran entre sí. Cada unidad se coordina junto con las correspondientes unidades de clientes y proveedores de la empresa a la que pertenecen, basándose sus relaciones en una serie de contratos y convenciones predefinidos.
- BPMAT es una librería desarrollada por IBM que trata de modelizar las actividades de una empresa. Esta se complementa con la ACL, librería que permite modelizar los flujos entre empresas. Estas herramientas se apoyan en los trabajos del equipo de Swaminathan (Swaminathan et al., 1998) de la Universidad de Carnegie Melon (Estados Unidos) que ha buscado elementos comunes a toda cadena de suministro comparando tres cadenas de suministro tipo.
- MASCOT (Multi-Agent Supply Chain coOrdination Tool) también desarrollado en la universidad de Carnegie Melon (Estados Unidos). Es una arquitectura basada en

agentes para la planificación y coordinación que busca la mejora de la agilidad de la cadena de suministro. El sistema coordina la producción a través de las instalaciones, tanto internas como externas, de la empresa, además de evaluar decisiones sobre la concepción de nuevos productos y de sus componentes.

- DaSCh ha sido desarrollado por Parunak y su equipo (Parunak and VanderBok, 1988) en ERIM (Michigan, Estados Unidos). Este proyecto explora las técnicas de modelización de los proveedores, incluyendo no solo los de primer nivel, sino los de los niveles siguientes. Buscan modelizar mediante agentes los flujos de materiales y de información teniendo en cuenta que estos no son perfectos.
- MASC está desarrollado en la Universidad d'Aix Marseille 3 (Francia) y estudia las formas de coordinación de los componentes de la cadena logística. Esta coordinación se realiza en respuesta a una oferta a los posibles participantes, que responden en función de su capacidad y de la carga de producción que ya posea. Los actores que ganen la oferta forman la cadena de suministro que llevará el producto deseado hasta el consumidor.
- OCEAN (Organization and Control Emergence with an Agent Network). Se trata de un sistema descentralizado y de arquitectura abierta, cuyo objetivo es mostrar que la cooperación a nivel global puede aparecer a partir de la competencia entre empresas a nivel individual. Este trabajo se ha realizado en el INSA de Lyon (Francia) y en la Universidad de Montpellier 2 (Francia)
- Charro, Santos y Reis [14] exploran una simulación multiagente para analizar la cadena de suministro, modelando el juego de la cerveza ("The Beer Game"), que es una simulación de una cadena de 4 niveles. Para esto se han utilizado las librerías REPAST de Java.
- En el trabajo de Chen [13], se desarrolla un sistema multiagente para simular una cadena de suministro, utilizando Redes Neuronales Artificiales (ANN) para calcular la política óptima de inventarios, basándose también en el juego de la cerveza. Se examina el rendimiento de las políticas óptimas bajando los costos y aumentando la eficiencia del sistema.

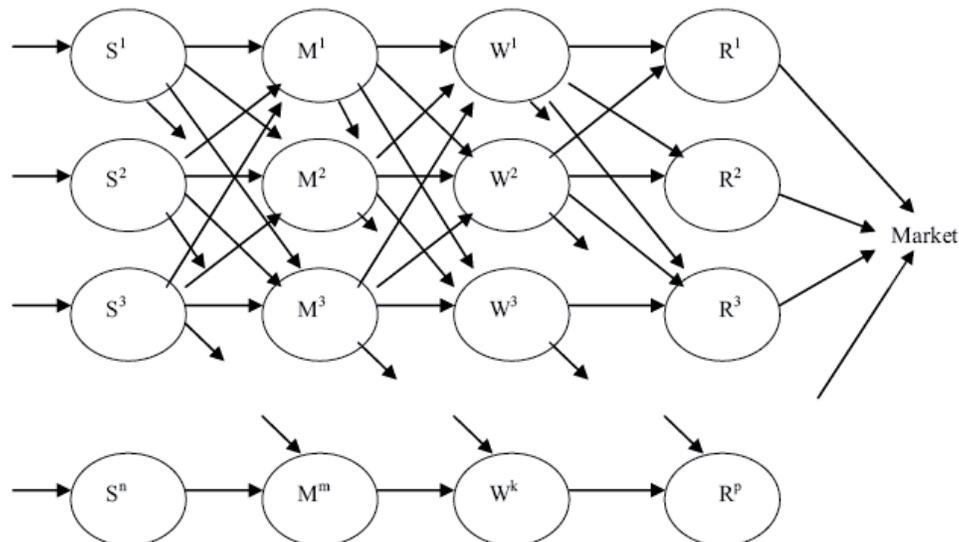
Además, se ha encontrado en la literatura otros trabajos relacionados a sistemas multiagente y cadenas de suministro, donde algunos son un marco teórico sobre el tema, y otros son trabajos que abarcan técnicas específicas. A continuación una breve reseña de estos trabajos.

### **3.8.1. R. Sarker - A Multi-agent Simulation Study for Supply Chain Operation**

Este paper presenta una simulación multiagente para una cadena de suministros de una fábrica. El modelo es capaz de manejar complejas redes con muchos niveles, donde cada uno

de los niveles cuenta con muchos procesos e interacciones entre estos. Se discute la arquitectura multiagente y se muestran los resultados de la simulación para analizar los aspectos operacionales. [15]

El trabajo considera una cadena de suministro con 4 niveles: minorista, mayorista, fabricante y proveedor. Los minoristas venden los productos directamente al mercado. Se supone que los proveedores tienen materiales suficientes para ofrecer a los fabricantes. Las actividades de los fabricantes se pueden dividir en dos tipos: (i) ordenar y recibir las materias primas, y el mantenimiento del stock y (ii) los pedidos de fabricación, recepción de los productos finales, y mantenimiento del stock. Los mayoristas reciben los productos de los fabricantes, ofrecen a los minoristas, y mantienen las existencias de los productos cuando sea necesario. Esta investigación se considera una cadena de suministro con varias entidades, en cada nivel, tales como múltiples proveedores, fabricantes, mayoristas y minoristas. La estructura de la red para este problema se muestra en la Figura 3.5. Además, se consideran varios productos de flujo a través de la red.



**Figura 3.5: Cadena de suministro, desde el proveedor (S) hasta el minorista (R).[15]**

En cualquier cadena de suministro, el minorista, mayorista, fabricante, proveedor o cualquier otra entidad puede ser tratado como un agente inteligente. Cada agente tiene sus propios modelos o algoritmos para tomar sus propias decisiones, tiene una serie de parámetros o indicadores para expresar su situación, tiene uno o más proveedores y tiene uno o más clientes. Por ejemplo, un minorista (un agente) determina su demanda en el mercado, calcula la cantidad de pedido propio, realiza pedidos, recibe productos procedentes de los distribuidores, actualiza su estado, calcula costos y vende en el mercado. Las características anteriores son comunes para todas las entidades de la cadena de suministro. Para el agente proveedor (en el lado izquierdo de la cadena), asumimos que tiene un stock infinito y no necesita de nadie para satisfacer la demanda. El agente minorista vende mercancías en la demanda y no acepta ninguna orden directa del mercado, sino que cumple con la demanda del mercado de su inventario. Se asume en el paper que el tiempo de espera es de cero para el minorista. Eso significa que, una vez que la oferta se recibe la mercancía está lista para la venta al mercado.

El agente minorista utiliza un simple método de promedio móvil de previsión de la demanda del mercado.

El agente utiliza el método del lote económico para determinar su cantidad de reabastecimiento y sigue una política de inventario continuo con los siguientes parámetros:

- Un nivel de inventario máximo de  $(QR + s)$ , donde QR es la cantidad de pedidos y s es el stock de seguridad.
- Resuelve la cantidad QR una vez que la acción alcanza  $(LR * D + s)$  de la cantidad, donde LR es el lead time y D es la tasa de demanda.
- La recepción de una orden se realiza después de LR períodos.

El nivel de inventarios y la escasez se calculan como sigue:

- El nivel de inventario es  $It = It-1 + \text{órdenes recibidas} - Dt$ , para  $It \geq 0$ , donde It y Dt son el nivel de inventario y la demanda en el período t
- Escasez =  $St = |It|$ , cuando  $It < 0$

El minorista selecciona su proveedor preferido basado en la fiabilidad del suministro, la preferencia de negocios, la proximidad geográfica o cualquier otro indicador.

El agente calcula su costo basado en las siguientes ecuaciones:

- Costo de inventario
- Costo de ordenamiento
- Costo de escasez
- Costo de transporte

El código de la simulación se ha desarrollado en Java. El tiempo de simulación (o actualización del reloj) se considera como un día. Después de cada paso de tiempo, la simulación va de derecha a izquierda de la cadena. Para cada agente, los pasos del algoritmo de simulación son muy similares y se examinan brevemente a continuación.

- En  $t = 0$ , se inicializan todos los parámetros e insumos necesarios
  - Pronosticar la demanda
- En cualquier momento  $t > 0$ , para cada minorista, a continuación, para cada mayorista, a continuación, para cada fabricante, a continuación, para cada proveedor:
  - Actualización del Pronóstico
  - Actualizar el nivel de inventario si hay algún cambio debido a la entrega de artículos vendidos o recibidos

- Si el stock está por debajo de la punto de reorden, determinar la cantidad de pedido y un pedido
- Cuando se realiza el pedido, se sigue las reglas de selección de proveedores
- Si hay escasez, se actualiza la cantidad de esta.
- En  $t = T$  (fin de la simulación), generar una salida

Después de ejecutar la simulación para un número determinado de pasos de tiempo, cuyos resultados detallados se almacenan en forma de tabla. De los resultados detallados, se generan resúmenes para el análisis.

El modelo de simulación requiere de gran cantidad de datos para la demanda del mercado. La entrada manual de datos sería una tarea muy tediosa, por lo que se leen los datos de un archivo de entrada. Se ha desarrollado un archivo HTML con un applet de Java para que los investigadores interesados puedan experimentar con el modelo en Internet. Cabe señalar que este experimento se lleva a cabo durante un determinado conjunto de datos de la demanda y un único producto. Sin embargo, la versión completa del sistema debía permitir cualquier tamaño de la demanda y manejar cualquier cantidad de productos.

### **3.8.2. Sara Saberi - Multi Agent System For Negotiation In Supply Chain Management**

En este artículo se ha desarrollado un sistema multi agente para simular una cadena de suministro. Cada entidad se modela como un agente y la coordinación para el control de inventarios minimiza el costo total de la cadena debido al intercambio de información y el conocimiento utilizando el mecanismo de negociación. El resultado mostró una reducción razonable del costo total y el efecto látigo. [19]

Se han tomado las siguientes premisas para el desarrollo de este trabajo:

- Las empresas están obligadas a cumplir con los pedidos de los clientes, aunque puede ser difícil de hacerlo.
- Las empresas tienen que responder a los pedidos de forma rápida y eficiente en el limitado tiempo disponible para cumplir con los requisitos del cliente.

Los pedidos inesperados, sin embargo, provocan en la mayoría de los casos retrasos en la entrega y disminuye la eficiencia en todos los miembros. Para coordinar las diferentes entidades de la cadena de suministro y resolver estos problemas, las decisiones de negociación han sido identificadas como cruciales para el éxito global de fabricación.

Las técnicas de negociación se utilizan para superar los conflictos, y para llegar a un acuerdo entre los agentes, en lugar de aceptar una solución predeterminada. De hecho, la negociación es el núcleo de muchas interacciones de los agentes ya que a menudo es inevitable entre los diferentes participantes en el proyecto, con sus tareas particulares y el conocimiento del dominio, interactuar para lograr sus objetivos individuales y lograr a su vez los objetivos del grupo. La importancia de la negociación en el MAS es probable que aumente

debido al crecimiento de la comunicación estandarizada rápida y poco costosa de infraestructuras, que permite diseñar agentes para interactuar en un entorno abierto, en tiempo real, realizando transacciones de forma segura.

El modelo propuesto en este paper se basa en el juego de la cerveza de cuatro niveles, pero asume que son entidades sin límite en escalón de clientes y una única entidad en los otros. Los tres niveles - proveedor, distribuidor y minorista - tienen sistemas de inventario diferentes. La cadena de suministro multi-escalón basada en agentes es simulada con los sistemas multiagente, siendo estos agentes capaces de resolver el problema adecuándose a la oferta, la demanda y la asignación de recursos en forma dinámica en tiempo real, mediante el reconocimiento de las oportunidades, tendencias y potencialidades, llevando a cabo las negociaciones y la coordinación. Los agentes se implementan como objetos JAVA utilizando el JADE toolkit que satisface el comportamiento descrito anteriormente. Esto significa que cada agente procesa por separado con su propio control interno y un buzón de correo. Los agentes pueden comunicarse de forma asíncrona utilizando todos los demás buzones. Los buzones se implementan como bases de datos con registros que representan los mensajes entrantes. Además de eso, todas las variables que representan el estado de un agente que también se almacenan en una base de datos modelo de estado. El modelo de simulación que se ha desarrollado, es lo suficientemente flexible como para añadir nuevos agentes, o para editar las propiedades de los agentes existentes para examinar los resultados de la subasta para diferentes escenarios de operaciones.

En este modelo, dos tipos de agentes se utilizan para responder a diversos tipos de servicios para la cadena de suministro. Uno de ellos es un agente de control y el otro es un agente de previsión de la demanda. Estos agentes tienen la capacidad de coordinación para especificar las características tanto estáticas como dinámicas de diversas entidades de la cadena de suministro. Los clientes generan demandas al azar y estocásticas al azar y sólo el minorista ve la demanda de los clientes y les envía las mercancías. De la misma manera se emplea entre el distribuidor y el minorista y entre el proveedor y el distribuidor. El tiempo de entrega entre escalones es determinista y el proveedor tiene infinitos bienes. Otras variables, como los costos de mantención y orden tienen un valor fijo. Estos supuestos son importantes y necesarios para la previsión del agente que se encarga de calcular la cantidad de orden y tiempo para el detallista y el distribuidor basado en la cantidad económica de pedido (EOQ).

Las órdenes del cliente que fueron insatisfechas se transforman como pérdida en el minorista. Para el distribuidor, frente a una falta de inventario, los pedidos de los minoristas serán cambiados a pedido pendiente. En este sistema, en caso de falta de inventario en el minorista para satisfacer las necesidades de los clientes, el agente minorista abre un mecanismo de negociación que ayuda a resolver este problema. Al iniciar el mecanismo de negociación el minorista obtiene los nuevos precios de los clientes y con los pedidos en curso y estos nuevos precios trata de identificar las necesidades de los clientes que se han de cumplir, mediante el método de la mochila. De esta manera se puede alcanzar el máximo beneficio para el detallista. En este sistema sólo hay una base de datos que es compartida por todos los agentes, de esta manera el agente "Comportamiento" utiliza esta información para la predicción basada en nuevos datos de otros agentes y luego envía la información para la actualización para el minorista y distribuidor.

### 3.8.3. Manuel D. Rossetti – An Object-Oriented Framework For Simulating Multi-Echelon Inventory Systems

En este trabajo se discute el diseño y la utilización de un framework orientado a objetos para la simulación de sistemas de inventario multi-escalón. Se presenta un resumen para describir la forma en que el framework puede ser utilizado a través de su aplicación en dos ejemplos. Además, se describe el diseño mediante la descripción de los elementos conceptuales más importantes en el modelo orientado a objetos. El marco está basado en la librería Java Simulation Library (JSL) y permite el modelado y ejecución de modelos de simulación. Los resultados y discusión indican la flexibilidad y la potencia del modelado con el framework. Además se muestra un resumen de la investigación futura del modelado de las cadenas de suministro complejas.[16]

Se presenta la capa de inventario a través de dos ejemplos (1) un simple modelo continuo (R, Q) y (2) un sistema de inventario multi-escalón y multi-elemento.

El paquete creado para tratar el inventario actualmente consta de 18 clases y 8 interfaces que representan diferentes elementos de modelado de inventario. La intención del paper es brindar el suficiente detalle para poder hacer el modelado conceptual con el paquete de inventario.

A fin de ilustrar el diseño, se da una breve descripción de las seis clases importantes:

- ItemType
- Demand
- DemandGenerator
- BackLogPolicyAbstract
- Inventory
- InventoryPolicyAbstract.

ItemType representa o describe los elementos o productos en el sistema de inventario. La clase Demand representa una solicitud para el inventario y proporciona el estado de la petición. Un ejemplo de registro de la demanda contiene el tipo de elemento asociado a la solicitud, el cliente de la solicitud, el proveedor de los artículos del inventario, el remitente de los artículos, el importe de la solicitud, los requisitos del cliente y otros asociados con la solicitud, como si está permitido pedidos pendientes o llenado parcial.

La clase DemandGenerator crea instancias de la demanda y actúa como un cliente que genera la demanda de los detalles de los requisitos del cliente.

La clase Inventory representa unidades de artículos que pueden ser solicitados y realiza un seguimiento de la cantidad de inventario en mano, atrasados, perdidos, etc. Una instancia de la clase de inventario representa el estado del inventario en un momento dado.

La clase Inventory proporciona métodos para solicitar el inventario y para cubrir las demandas para este. Cada clase de inventario se asocia con una política de inventario.

El InventoryPolicyAbstract es una clase base abstracta que permite la encapsulación de reglas para controlar el inventario asociado. Es una regla, política o estrategia que rige el comportamiento de reordenamiento del inventario. La política de inventario determina el momento de ordenar y cuánto ordenar. Una serie de políticas de inventario diferentes se han implementado en el paquete de inventario. Asociada a una clase de inventario es una política de atraso.

La clase abstracta de atraso, BackLogPolicyAbstract, representa las diversas normas o comportamientos que pueden ser utilizados a las demandas de acumulación de inventario y para cubrir los retrasos asociados con el inventario.

Las clases descritas anteriormente permiten la simulación de un Modelo de Inventario de un solo nivel. Cuando se produce una demanda, la cantidad demandada se cuantifica, y a continuación el sistema verifica la disponibilidad de stock. Si la cantidad disponible es suficiente para llenar orden, la demanda se satisface y la cantidad disponible se decrementa. Por otro lado, si la cantidad disponible no es suficiente para llenar la orden, toda la orden queda pendiente de entrega. Los pedidos pendientes se acumulan en una cola, entregándose bajo la metodología FIFO después de llegada la orden de reposición. La cantidad de inventario disponible se comprueba cada vez que se realiza un pedido, y cuando se satisface una entrega pendiente. Cuando la cantidad del inventario cae bajo el punto de reorden, se crea una orden de reposición. La orden de reposición tomará un cierto tiempo en llegar. Una vez llegado el pedido de reposición, la cantidad disponible incrementa.

Además, se incorporan nuevas clases para poder modelar un Sistema de Inventario Multi-nivel, las cuales son:

- InventoryIfc
- InventoryCustomerIfc
- InventoryProviderIfc
- InventoryShipperIfc
- InventoryHoldingPoint
- LeadTimeReplenishmentProvider
- NoDelayShipper

- TimeBasedItemTypeShipper

La clase InventoryCustomerIfc declara todas las operaciones relativas a un cliente y la clase InventoryProviderIfc describe el comportamiento de un proveedor de inventario. Se le puede consultar a un proveedor de inventario si dispone del inventario que cumple con los requisitos de un cliente por medio del método providesInventory() y luego el getInventory() puede utilizarse para conseguir el acceso al inventario en la localización de un proveedor en particular.

A fin de facilitar el traslado de los pedidos de reposición, existe la clase abstracta InventoryShipperAbstract que se puede utilizar para modelar este comportamiento. Por ejemplo, dos envíos típicos son: NoDelayShipper donde la reposición se realiza al instante y TimeBasedItemTypeShipper para causar un retraso en la reposición (Lead Time) en función del tipo de artículo que es enviado. La clase InventoryHoldingPoint, encapsula todos los costos de mantener material almacenado en un lugar determinado.

La clase LeadTimeReplenishmentProvider es bastante similar a la InventoryHoldingPoint salvo que en este caso hace referencia a un inventario que a la misma vez es su propio proveedor. La clase LeadTimeReplenishmentProvider contiene a su vez al objeto TimeBasedItemTypeShipper, que indica el tiempo de retraso en la reposición a su cliente de acuerdo con lo especificado en la distribución. Esto permite que al proveedor primario del sistema multinivel que se modela (que tiene una cantidad infinita de elementos) se le pueda solicitar material con un tiempo de retraso.

## 4. Solución propuesta

### 4.1. Descripción de la propuesta

La solución propuesta consiste en diseñar un sistema inteligente que simula una cadena de suministro, y en la cual se intenta resolver el problema del control de inventarios. Como variante de este se ha seleccionado una cadena de 3 niveles (Almacén central, Centro de Distribución y Retailer) con la particularidad que estos inventarios serán multi-ítem, trabajando en la minimización de los costos al mismo tiempo de proveer un alto nivel de servicio al cliente.

Como requerimiento para llevar a cabo esta propuesta, se necesita de una tecnología inteligente para poder adaptar las demandas de los clientes conforme estos van variando, tomando decisiones por facultad propia de forma eficiente. Además, se necesita que el sistema sea distribuido para que cada almacén dentro del inventario sea independiente en sus acciones, pero que a la vez pueda comunicarse con los demás.

Para lograr el objetivo se trabajó en el diseño de un sistema multiagente, tecnología que cumple con todos los requerimientos anteriormente dados, mediante la cual cada entidad participante del problema de inventario será representada por un agente. Estos agentes a su vez, se envían mensajes para coordinar el manejo de los productos mediante un lenguaje de comunicación común.

Observando propuestas anteriores, como las investigaciones señaladas en el capítulo anterior, se diseñó el sistema multiagente siguiendo los estándares FIPA, y utilizando como metodología de desarrollo PASSI. Para modelar el sistema se utilizó la herramienta Rational Rose, disponible en la Escuela de Ingeniería Informática, agregando el PASSI Toolkit para la adaptación de UML al enfoque de agentes. Para codificar el sistema se utilizó la tecnología JADE, derivada de JAVA, la cual sirve como plataforma para la creación y comunicación de los agentes. También se tomó como base la metodología RUP, realizando un desarrollo iterativo, con variaciones durante el transcurso del proyecto.

Los agentes identificados son:

- **Supplier Agent:** Es el agente encargado de proveer artículos hacia los niveles inferiores, lo que normalmente sería el fabricante del producto o un almacén central. Es el nivel más alto dentro del modelo del inventario. Tiene como característica principal el supuesto de que posee una cantidad infinita de materiales para ser distribuido al resto de los niveles que realicen pedidos.
- **Inventory Agent:** Es el agente encargado de representar un inventario dentro del sistema. En la formulación inicial del sistema se había determinado trabajar con 2 tipos de agentes inventario (Centro de Distribución y Retailer), pero se ha determinado que ambos cumplen la misma función, y utilizan los mismos datos de entrada y salida para

comunicarse con los demás agentes, diferenciándose solamente por la facultad de ofrecer retraso en la entrega del pedido al cliente por falta de existencias o no. Mientras el retailer no ofrece la opción de entregar los productos con retardo a los clientes finales (ya que es parte del supuesto que estos elegirán otro lugar donde comprar), el centro de distribución sí lo debe hacer, para garantizar la existencia de productos de los inventarios inferiores en el futuro. Estos inventarios, por un lado, avisarán al agente monitoreador los movimientos y pasos de mensajes que realicen, y por el otro, imprimirán en un archivo texto los pedidos realizados y recibidos a lo largo de la simulación.

- **Inventory Creator Agent:** Este agente será encargado de realizar 2 grandes funciones. La primera será la de crear y lanzar los agentes que compondrán el sistema a la plataforma, creando según los datos de un archivo los inventarios en los distintos niveles, junto a los ítems que estos tendrán, la cantidad y la política de los productos con faltas de existencias. La segunda función será la de generador de la demanda, ya que según datos de un archivo, este agente hará el papel de cliente final realizando pedidos a los distintos inventarios retailer.
- **Dealer Agent:** Es el agente encargado de realizar el pedido de ítems desde un inventario de nivel inferior (Centro de Distribución o Retailer) hacia su inventario proveedor. Al realizar un pedido, si este ha sido aceptado, debe esperar un tiempo (lead time o tiempo de reaprovisionamiento) para poder entregar el pedido al inventario que lo realizó. Este lead time es variable para cada nivel del inventario.
- **Monitor Agent:** Este agente es el encargado de monitorear el trabajo de los distintos agentes del inventario, ya que es quien recibe la confirmación de los agentes de que han realizado todas las tareas en el día. Además es quien avisa a los agentes Dealer que se ha pasado al siguiente día, para que estos lo descuenten del lead time. Los agentes inventario le enviarán además los datos de estos como el costo total, lead time o nivel de servicio. A partir de esto, se mostrarán los resultados de las simulaciones mediante una interfaz y en archivos de texto.

Algunas funcionalidades que debe entregar el sistema son:

- **Comunicación entre los agentes:** El sistema debe manejar elementos críticos de la comunicación entre los agentes como es la concurrencia de datos, vale decir, que controle cuando 2 o más agentes (o varios comportamientos dentro de 1 solo) intente acceder a la base de datos (implementada en tablas hash) con la información de los productos y la cantidad de estos, para que no se produzca inconsistencia en la información.
- **Control del inventario:** Esto quiere decir que el sistema debe ser capaz de calcular, de acuerdo a la demanda, el momento en el que debe un almacén pedir uno o más artículos al nivel superior del inventario, y por supuesto también indicar la cantidad de estos.

- **Monitoreo de la simulación:** Debe mostrarse mediante una interfaz los acontecimientos que sucedan en la simulación del inventario. Eventos como el pedido de un cliente, entrega de un producto, petición a un nivel superior de artículos, aceptación o rechazo del pedido, y los valores que sean calculados mediante las políticas, como son los costos totales, cantidad óptima a pedir, nivel de servicio, etc.

## **4.2. Justificación**

Este proyecto se justifica ya que debido al mejoramiento de las tecnologías inteligentes y distribuidas es posible realizar sistemas que resuelvan eficazmente el problema de control de inventarios, logrando dinamicidad, proactividad en la toma de decisiones de parte de cada una de las entidades que participan en la cadena, y un mejor nivel de comunicación entre todos los niveles. Además, será un punto de partida para futuras investigaciones en el área, entregando un prototipo funcional que sirva tanto como para mejoramiento, como para la integración de este sistema a otros sistemas de mayor envergadura. Los protocolos de FIPA y la metodología PASSI están teniendo un gran auge como estándares en el diseño de sistemas multiagente, pero aún es un mundo nuevo en el que hay mucho por crear.

## **4.3. Alcances y limitaciones**

- Para plasmar la arquitectura, distribución y dinamicidad de los inventarios, un sistema multiagentes parece ser una de las soluciones más apropiadas.
- Se comenzó a diseñar e implementar el sistema partiendo por la creación de los agentes y su integración a la plataforma, siguiendo por la comunicación entre ellos, y terminando con la realización efectiva de los roles de cada agente.
- Se decidió implementar la política de revisión periódica con demanda incierta, que es la más aconsejable para el trato de inventarios multi-ítem, ya que de esta forma pueden revisarse al mismo tiempo los niveles de inventario de todos los artículos, realizando pedidos conjuntos, lo que sería lo más acercado a la realidad.
- El sistema simulará la demanda durante el tiempo que se especifique en los archivos (aunque las pruebas se han realizado con simulaciones de 1000 días), donde se realizarán saltos de 1 día en el que el agente monitoreador deberá recibir la confirmación de todos los agentes involucrados de que han finalizado sus tareas para avanzar al siguiente día. La revisión de los pedidos y políticas irá desde el retailer hacia los inventarios superiores.
- Los datos que se utilizarán para probar el sistema y realizar la simulación serán ficticios, generándose mediante la aplicación MS Excel, a partir de una media y desviación estándar dadas.
- La información de los ítems que tendrá cada almacén en el inventario será almacenada en una tabla hash que tendrá internamente cada uno de estos.

## **4.4. Resultados esperados**

Una vez que el sistema tenga una versión estable, esta debe ser capaz de:

- Lograr propuestas efectivas para el control de inventarios, como la cantidad de pedido en una orden, y el intervalo de tiempo para cada servicio. La idea es lograr el mejor servicio al cliente posible.
- Ser una base que sirva para la investigación futura de sistemas multiagente en el área de logística, ya sea para estudios dentro de esta universidad o para otras personas.

## **4.5. Especificaciones de diseño**

En esta sección se presenta el diseño del sistema, siguiendo una metodología de desarrollo iterativa. Debido a razones de espacio, se presentarán en este capítulo solamente los diagramas más importantes. Los diagramas restantes se encuentran en el anexo al final este documento.

### **4.5.1. Modelo de Requerimientos del sistema**

#### **Descripción del Dominio**

En este modelo se entrega una visión general del sistema, mostrando las funcionalidades principales basándose en casos de uso.

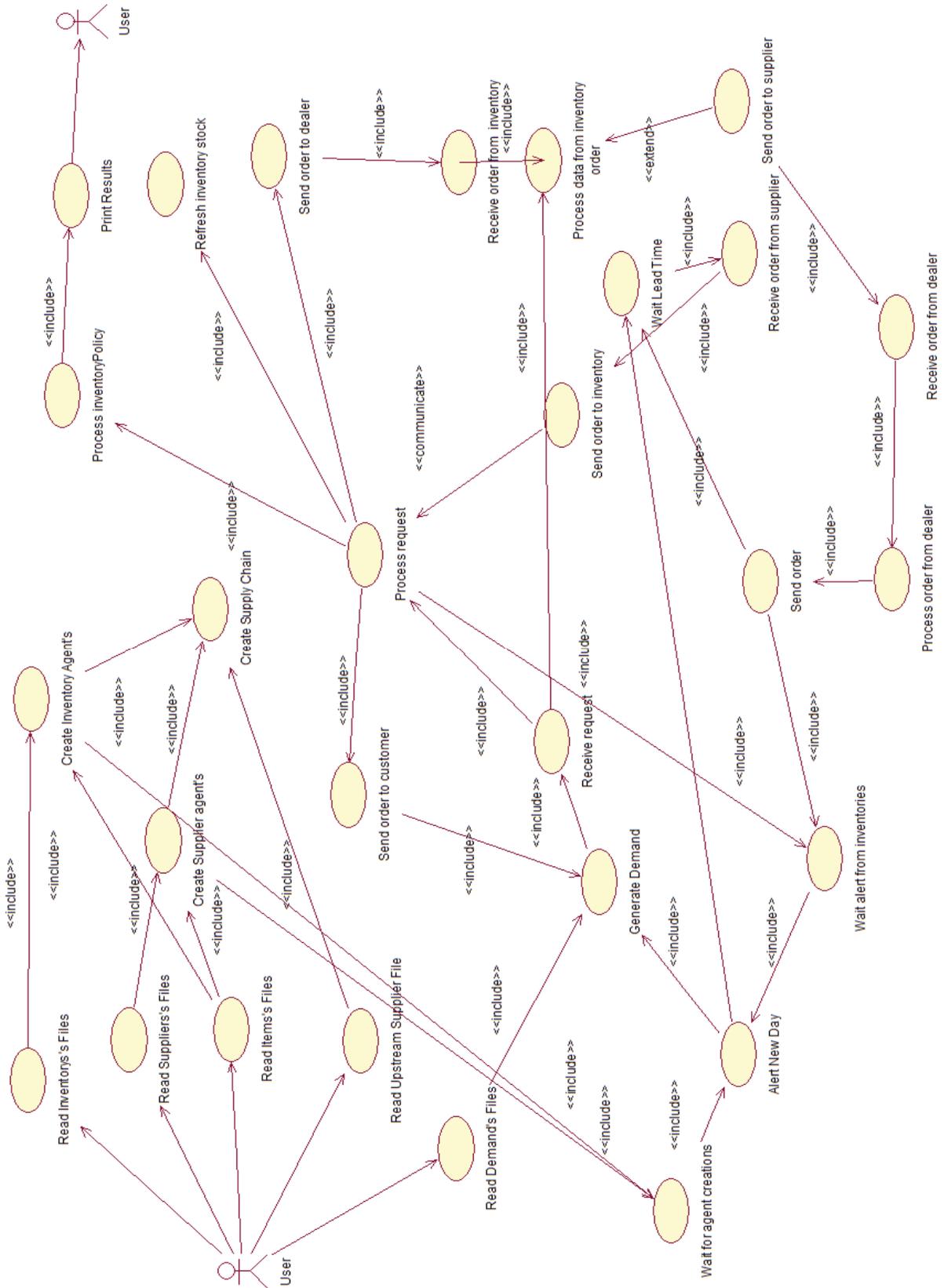


Figura 4.1: Descripción del Dominio

## **Identificación de agentes**

A partir del modelo obtenido en el punto anterior, se obtienen los agentes que interactuarán en el sistema, identificándose sus funcionalidades y como se comunicarán. A diferencia del modelo anterior, en el que se utilizan las relaciones <<include>> o <<extend>> en todos los casos, se agrega una nueva relación, para definir la comunicación entre 2 casos de usos pertenecientes a distintos agentes. Esta interacción se llamará <<communicate>>.

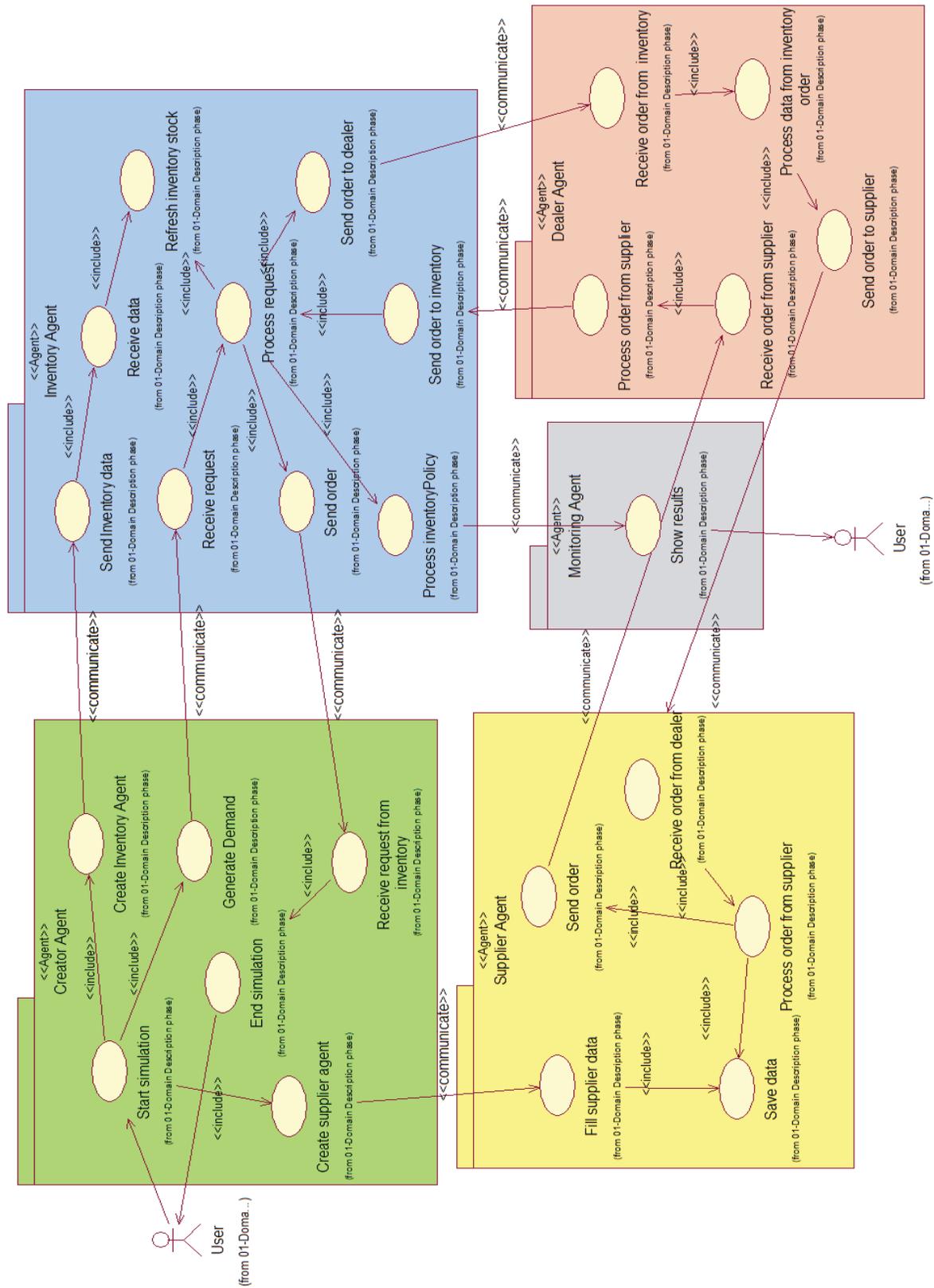


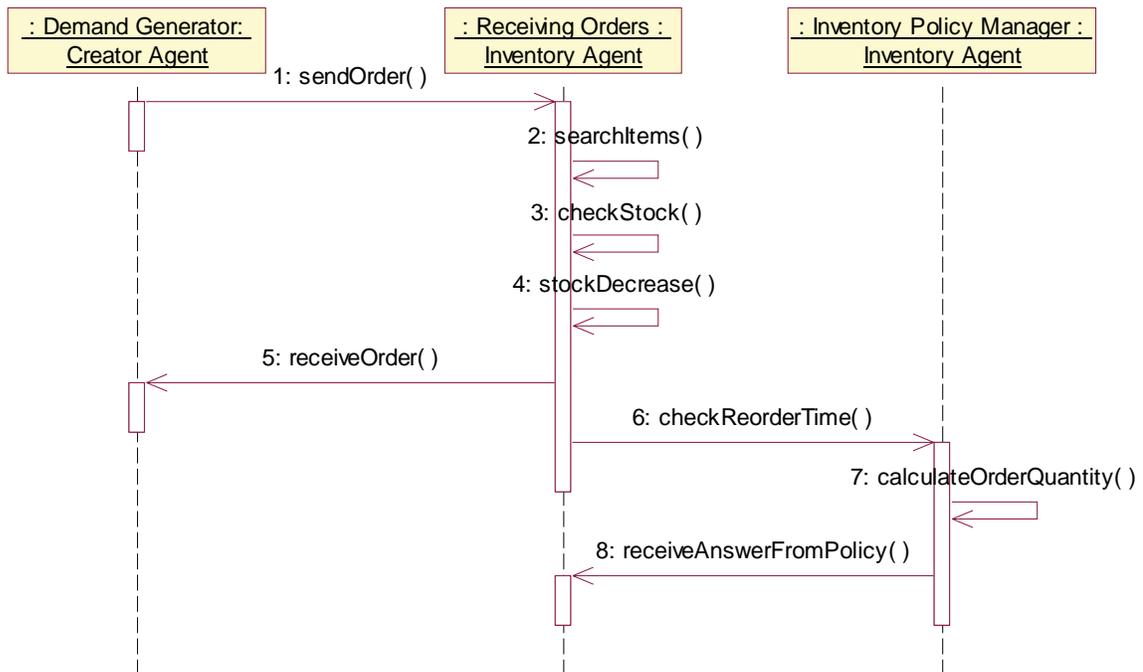
Figura 4.2: Identificación de Agentes

## Identificación de Roles

En este modelo se utilizan diagramas de secuencia para explorar las responsabilidades de cada agente a través de escenarios específicos para cada rol.

En ese informe se han creado 2 escenarios: El primero corresponde a un pedido por parte del generador de demanda, donde el retailer tiene el stock necesario para satisfacer este pedido. Este es el caso ideal en el trabajo del sistema.

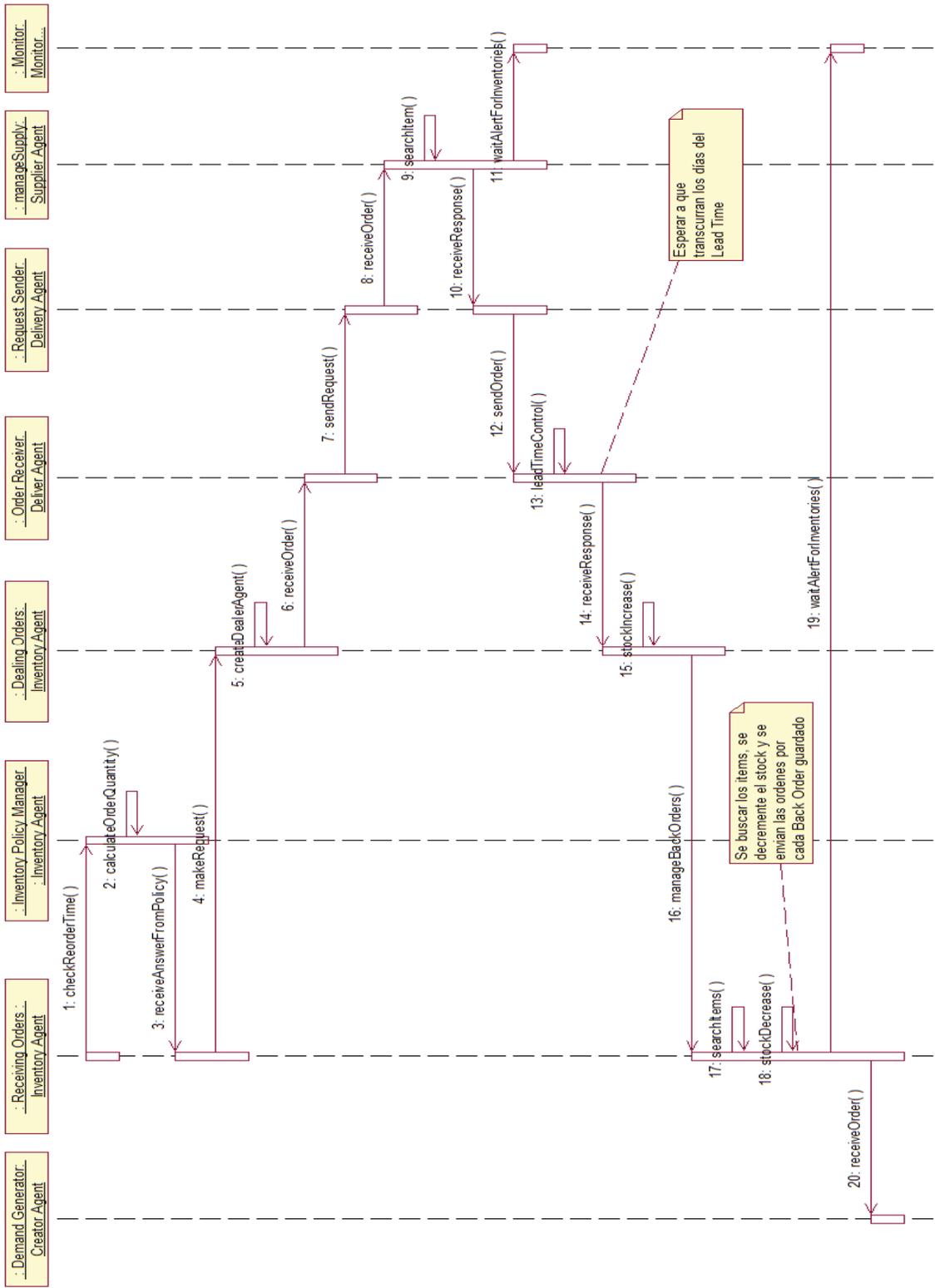
### *Escenario: Realizar pedido al retailer con stock suficiente*



**Figura 4.3: Realizar pedido al retailer con stock suficiente**

El segundo escenario muestra cuando un inventario realiza un pedido al supplier debido a que se lo exige la política que utiliza. En este caso el supplier comprueba si tiene el ítem que se le pide, y lo entrega. El inventario luego deberá revisar las órdenes que tiene pendientes para entregarlas.

**Escenario: Realizar pedido al supplier**



**Figura 4.4: Realizar pedido al supplier**

## Especificación de tareas

Este modelo muestra la especificación a través de diagramas de actividad de los comportamientos de cada agente, donde cada nodo representa una actividad que el agente debe hacer. A continuación se muestran las tareas de los 3 principales agentes: Inventory, Dealer y Supplier.

### Inventory Agent

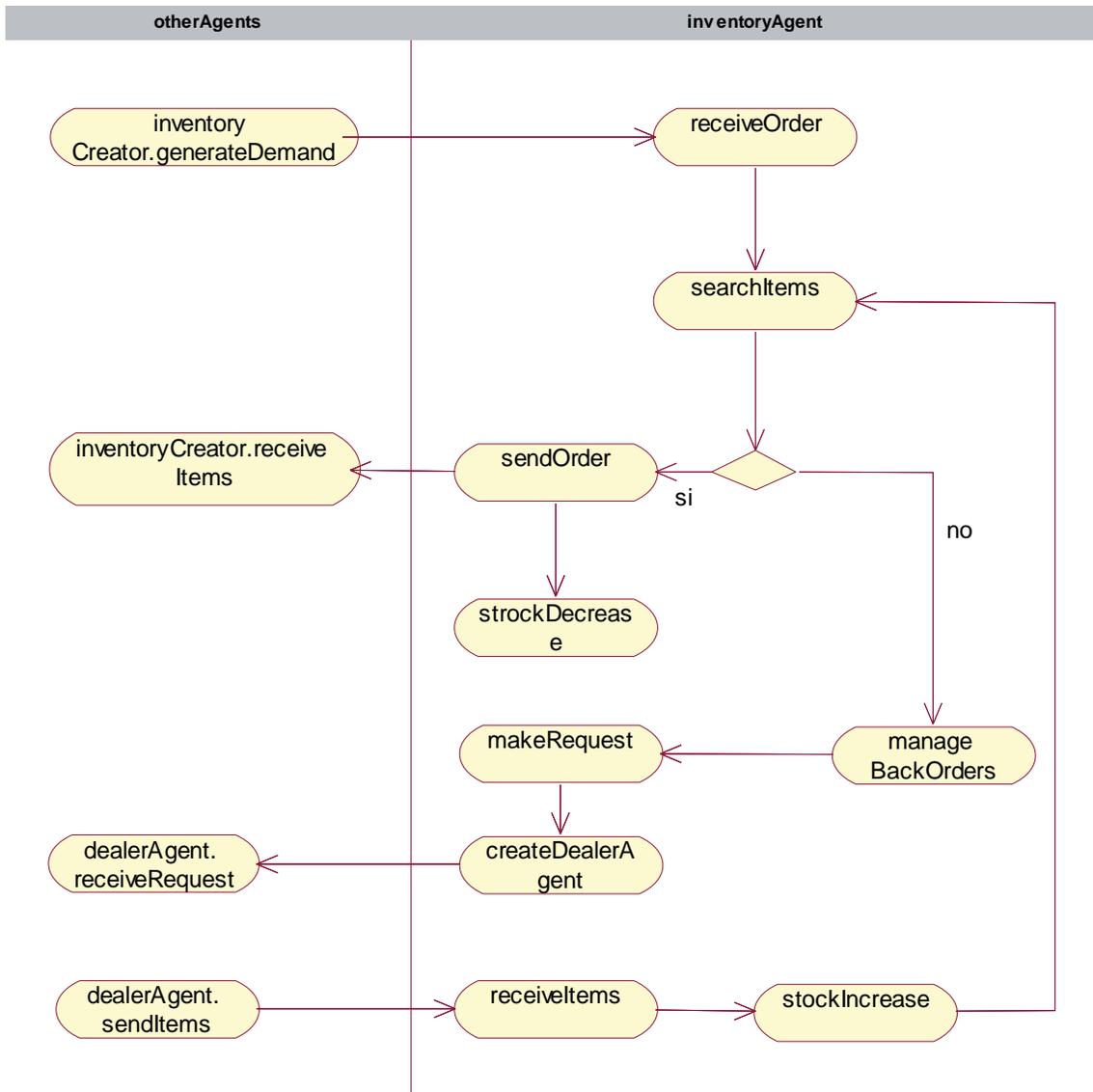


Figura 4.5: Inventory Agent

## Dealer Agent

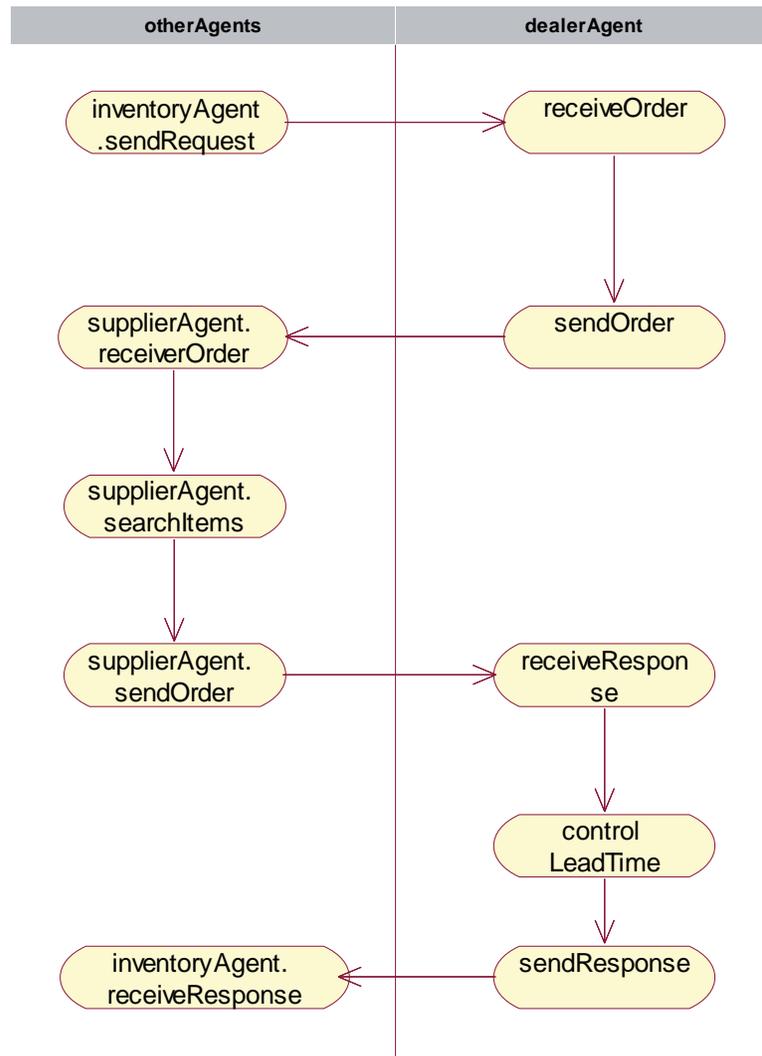


Figura 4.6: Dealer Agent

## Supplier Agent

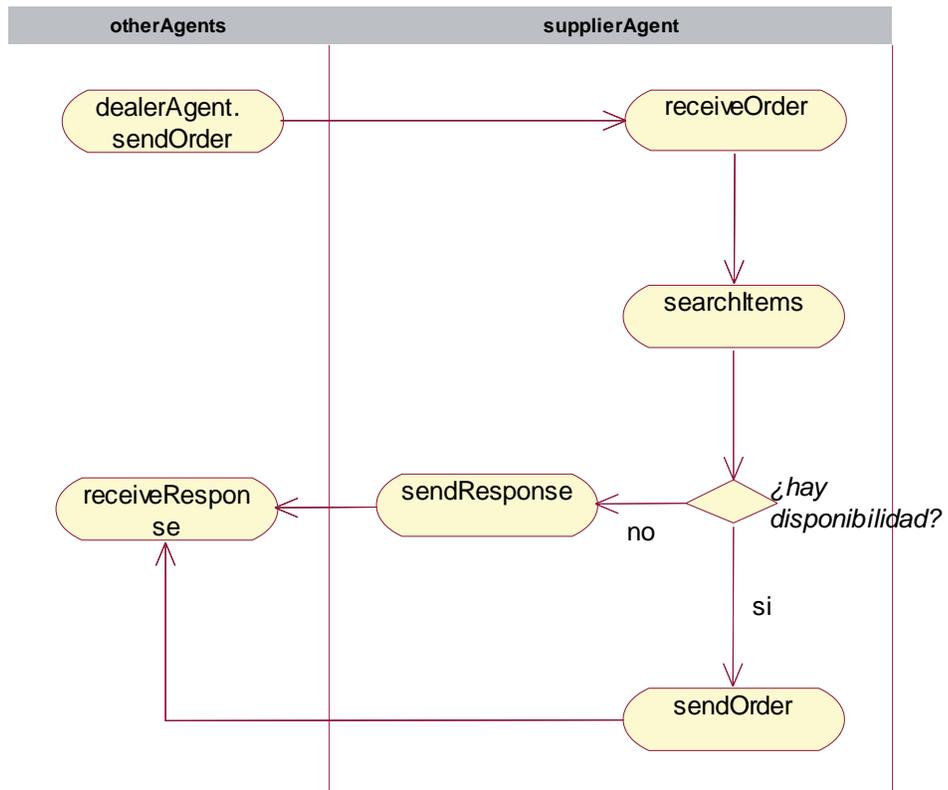


Figura 4.7: Supplier Agent

## 4.5.2. Modelo de Sociedad de Agentes

### Descripción de la ontología de dominio

Este diagrama muestra conocimiento atribuido a los agentes individuales y la pragmática de sus interacciones

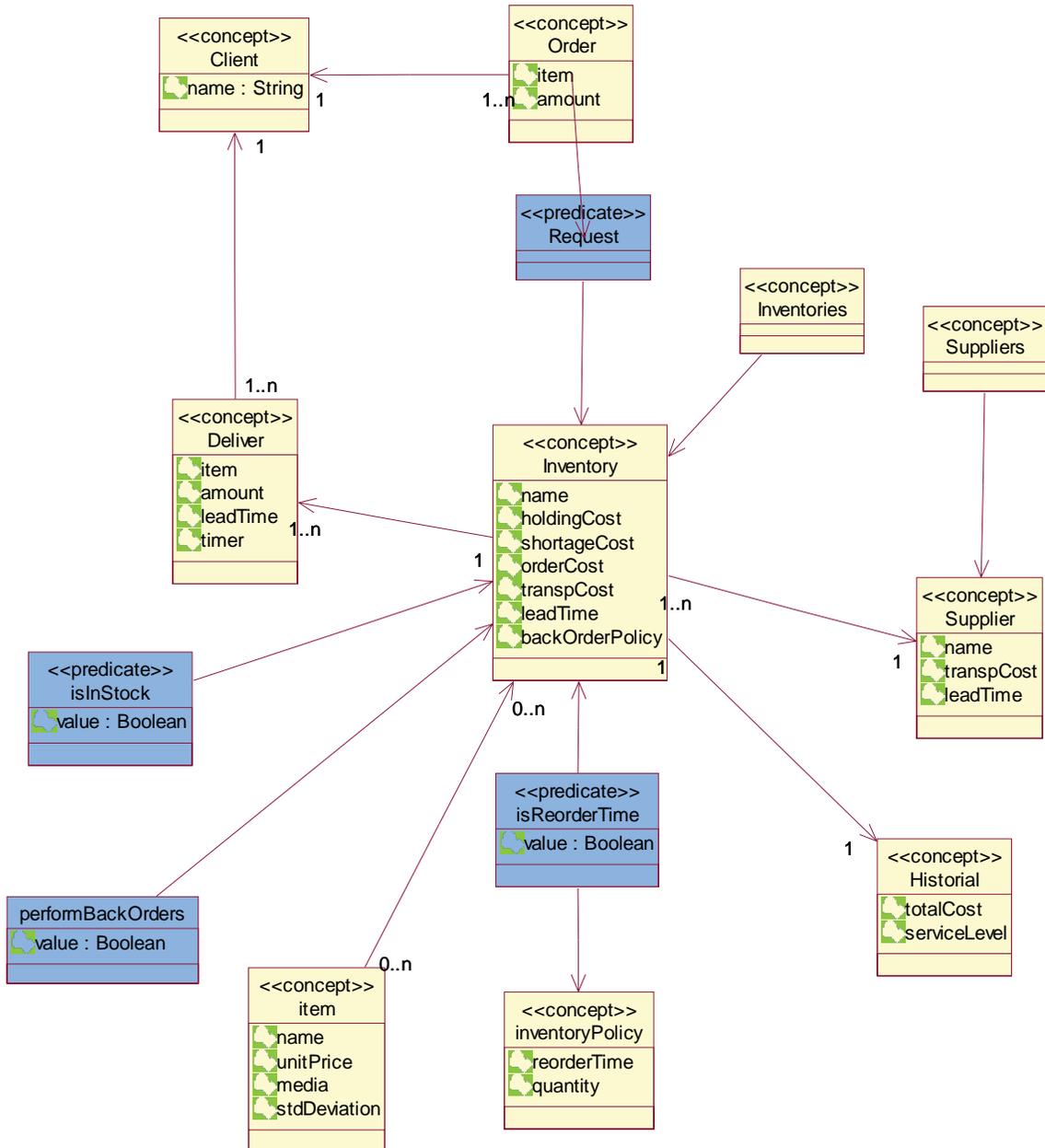


Figura 4.8: Descripción de la ontología de dominio

## Descripción de Roles (R.D.)

Utiliza diagramas de clases para mostrar los distintos roles que puede tener un agente, las tareas involucradas en cada rol, las capacidades de comunicación y las dependencias inter-agentes.

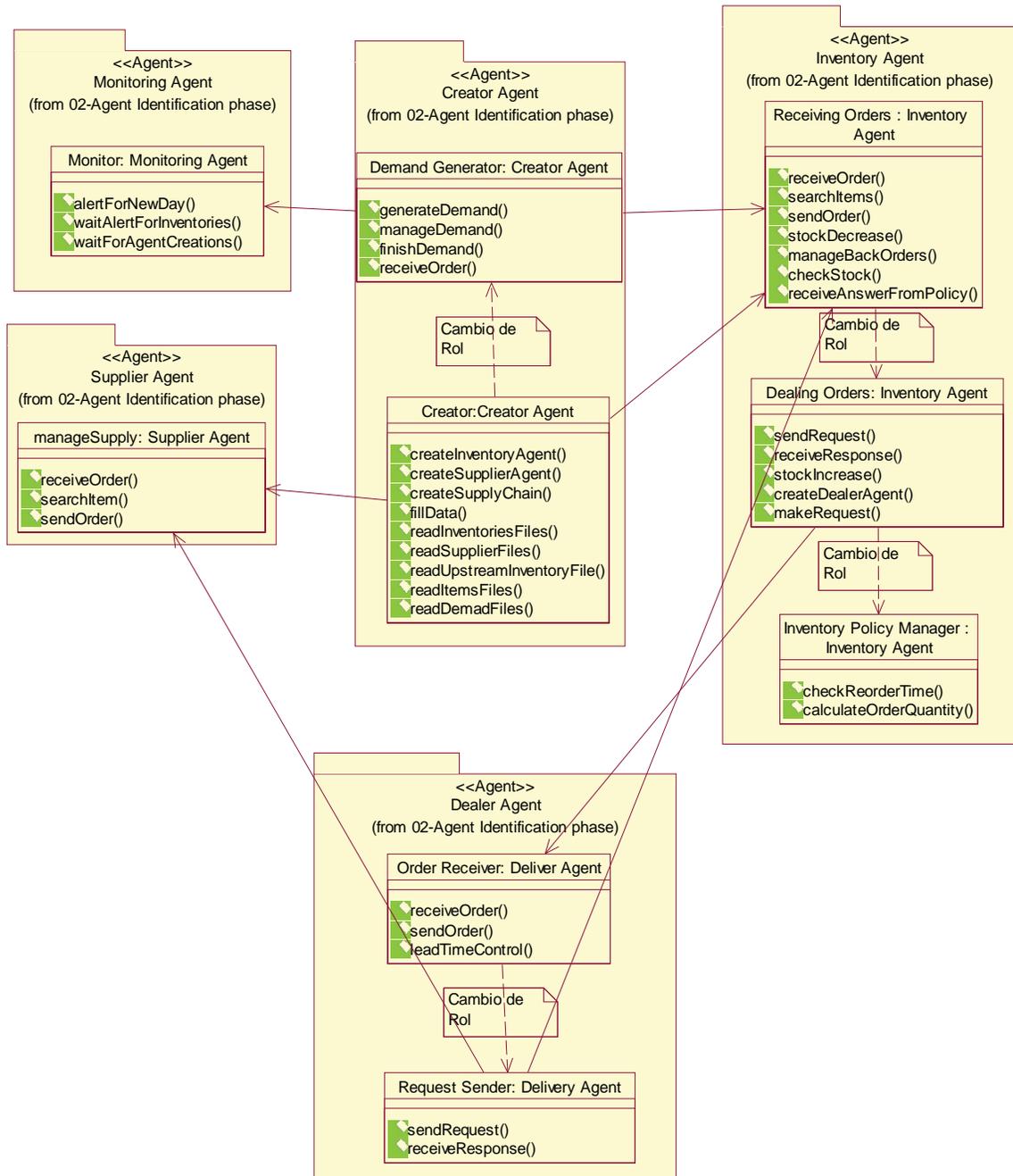


Figura 4.9: Descripción de Roles

### **4.5.3. Modelo de Implementación de Agentes**

#### **Datos Necesarios al interior de los agentes**

##### **Inventory Creator Agent**

Este agente leerá desde archivos de texto las demandas de los clientes a los que simulará para que hagan los pedidos a los retailers durante el período que se estime.

##### **Inventory Agent**

Debe tener la información de los ítems que contendrá, además de la cantidad de estos. También debe tener implementada su política de inventario para calcular la cantidad de pedido en cada revisión periódica. Además debe guardar un vector con los pedidos que no ha podido cumplir (en el caso de ser un centro de distribución), así como el ID de su Inventario superior.

##### **Dealer Agent**

Debe tener la información del pedido que se ha realizado, así como el ID del inventario al que se le realizará el pedido. Como último dato, se debe tener el tiempo necesario para el reaprovisionamiento (en días), para saber cuánto tiempo del reloj debe esperar para entregar el pedido.

##### **Supplier Agent**

Este agente contendrá la lista de ítems que podrá entregar, implementado en un vector.

##### **Monitor Agent**

Este agente contendrá el número de agentes inventarios y suppliers, para saber en qué momento debe avanzar al siguiente día.

## Descripción de la Estructura de Agentes (A.S.D.)

### Definición de la Arquitectura Multiagente

El diagrama de clases contiene clases y actores. Cada clase simboliza un agente del sistema. Los compartimientos de los atributos se pueden utilizar para representar el conocimiento del agente (refiriendo las entidades definidas en la descripción de la ontología de dominio), mientras que los compartimientos de las operaciones se utilizan para especificar las tareas del agente. Las relaciones indican el flujo de la información intercambiada (las comunicaciones).

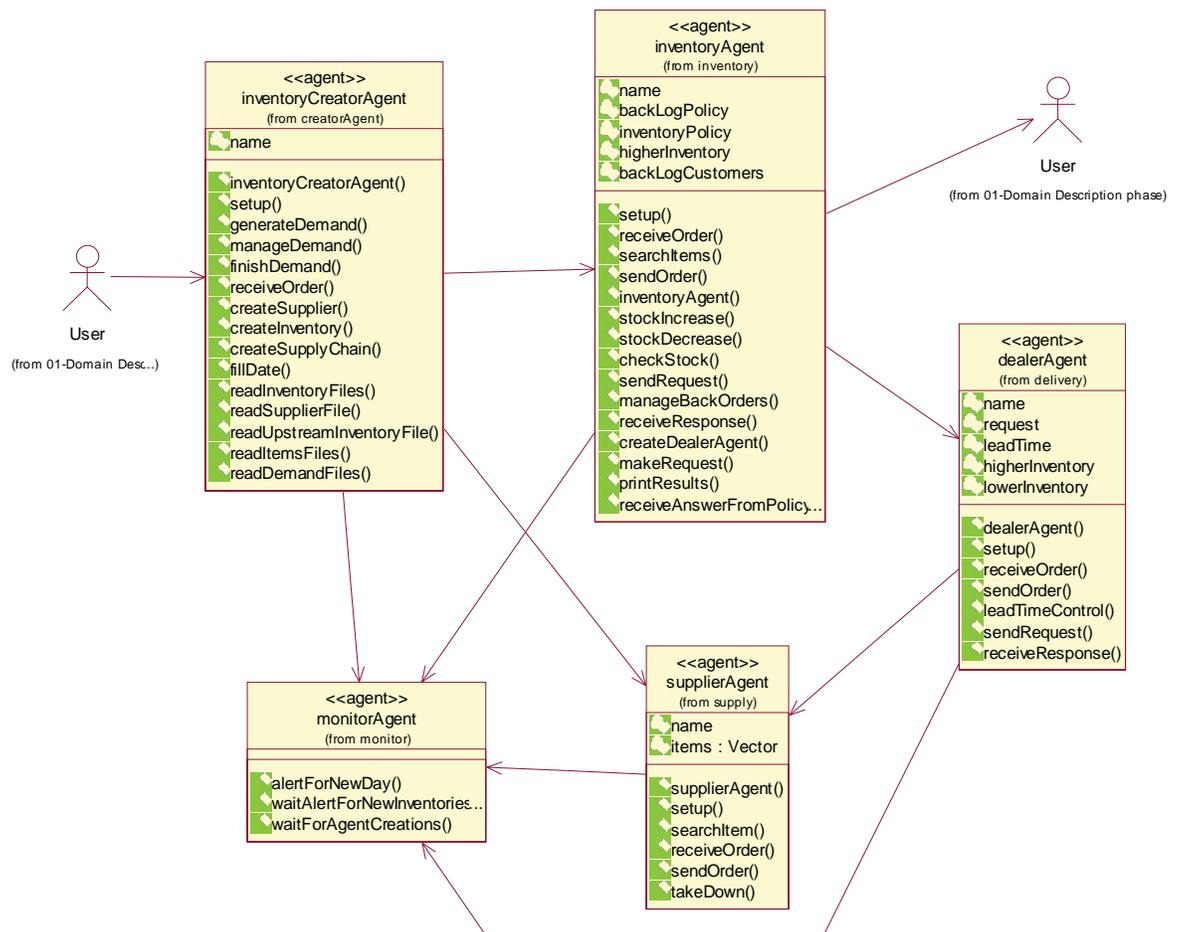


Figura 4.10: Definición de la Arquitectura Multiagente

## Definición de la Estructura del Agente Individual

Utiliza diagramas de clases para describir la estructura de las clases de agentes.

### Inventory Agent

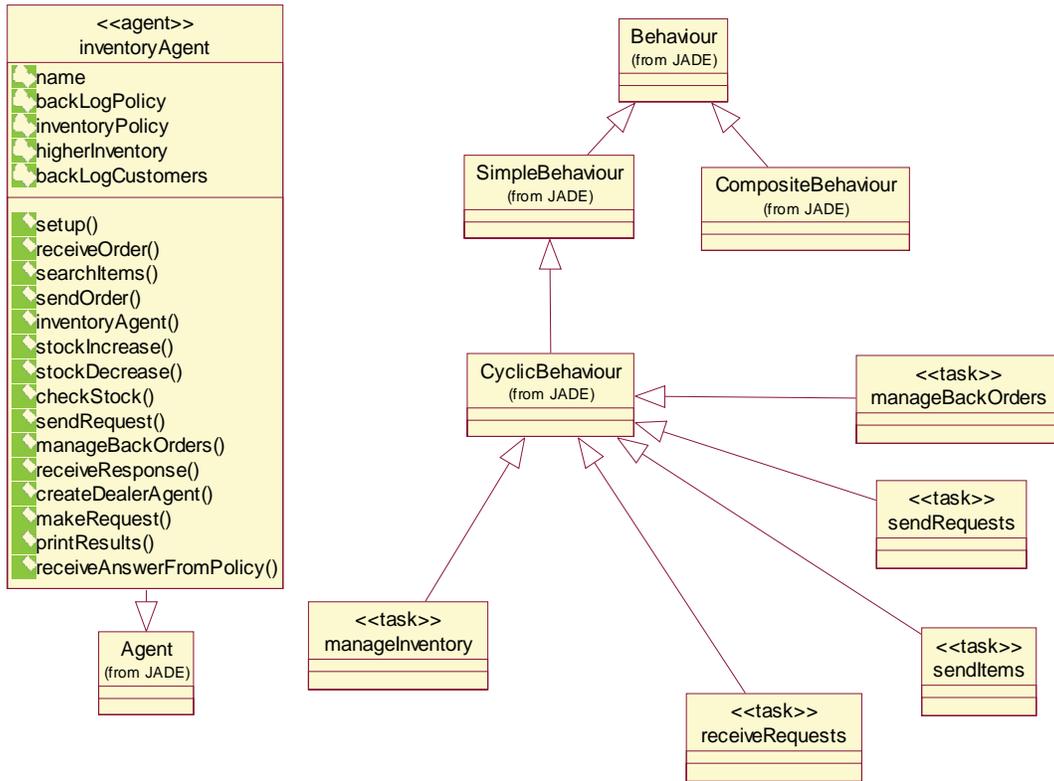
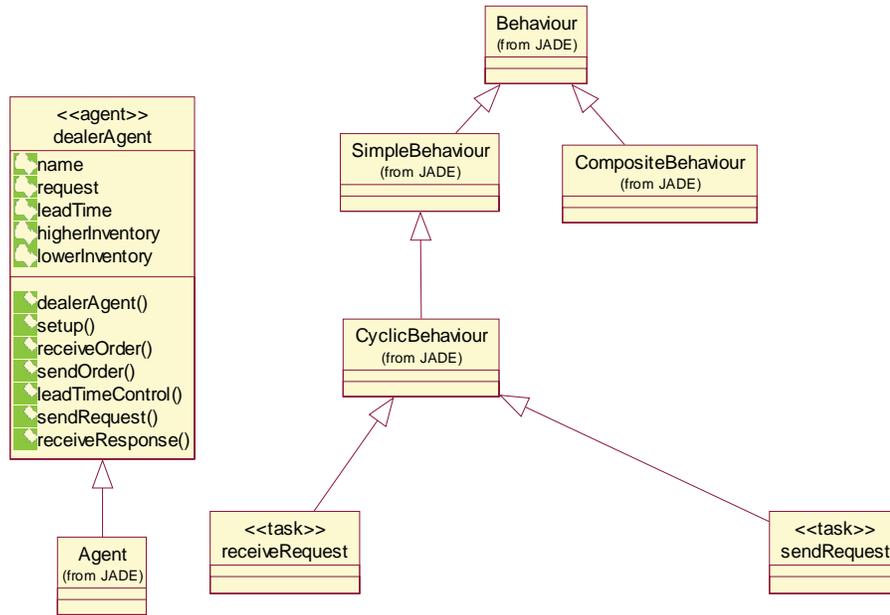


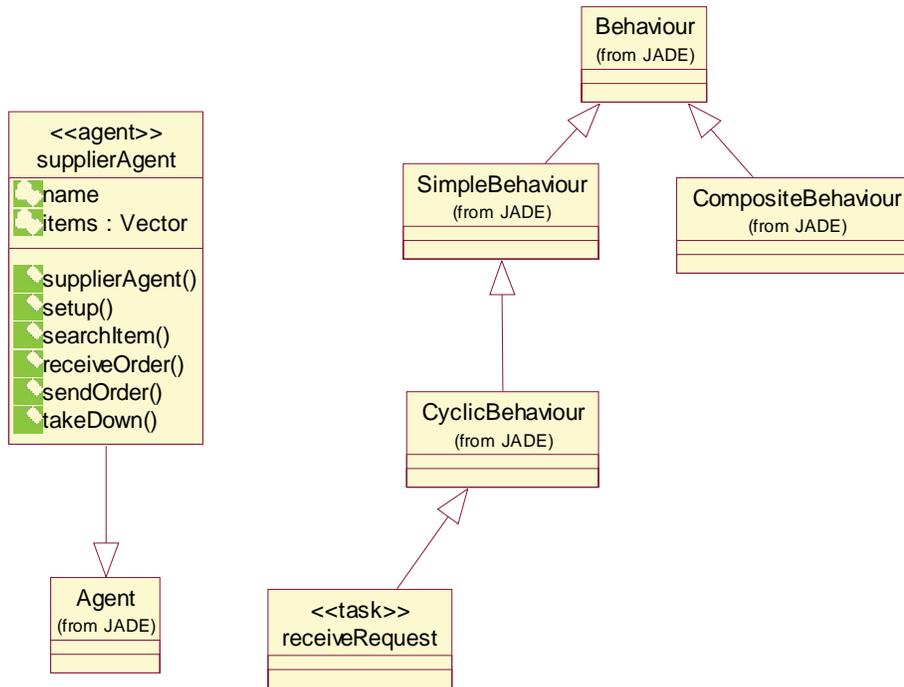
Figura 4.11: Inventory Agent

### Dealer Agent



**Figura 4.12: Dealer Agent**

### Supplier Agent



**Figura 4.13: Supplier Agent**

## MonitorAgent

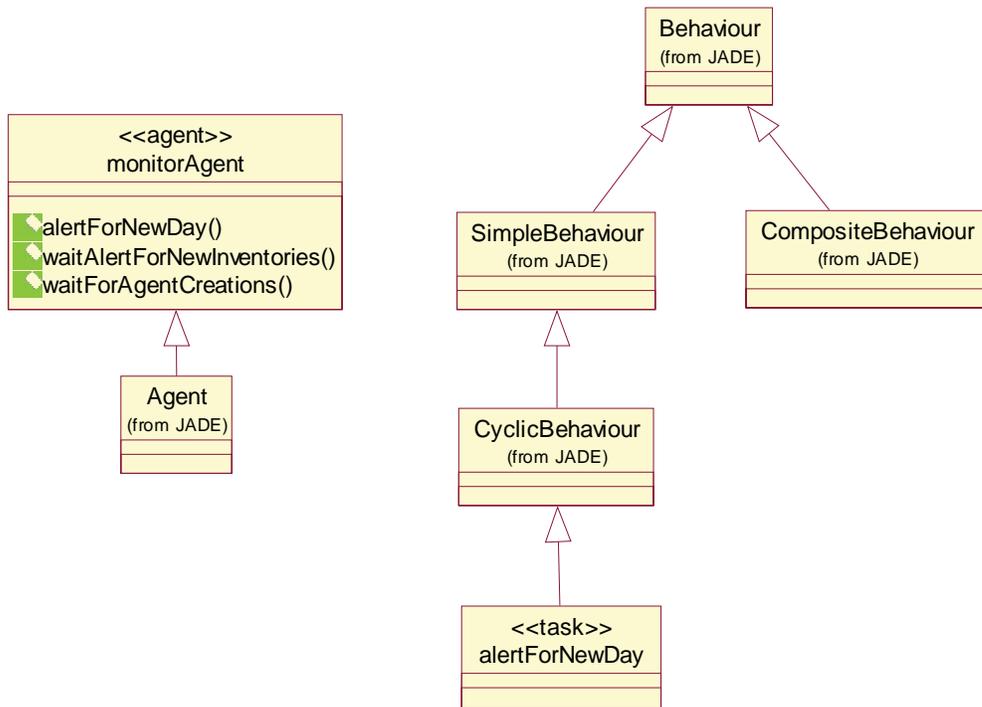


Figura 4.14: Monitor Agent

## Descripción del Comportamiento Multiagente

Este diagrama de actividad puede ser utilizado para mostrar el flujo de acontecimientos entre y dentro las clases agente y sus clases internas (que representan sus tareas).

El siguiente diagrama representa el momento en que el inventario debe realizar un pedido al supplier, de acuerdo a su política de inventario.

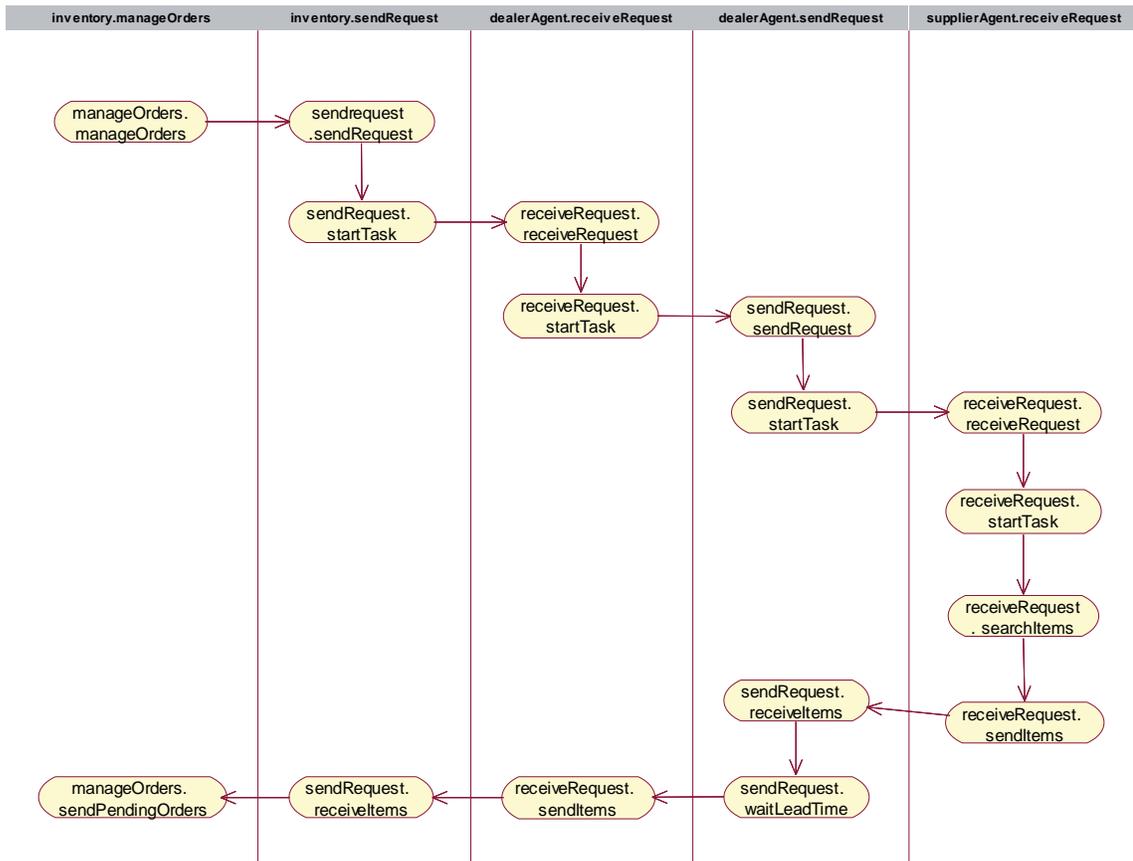


Figura 4.15: Descripción del Comportamiento Multiagente

## 5. Implementación y resultados de la arquitectura propuesta

En este capítulo se presenta en detalle la implementación del sistema, los archivos de entrada, la interfaz gráfica, los datos de prueba y los resultados generados a partir de las simulaciones.

### 5.1. Archivos de entrada y salida

En la simulación se utilizan 5 tipos de archivos, 4 de entrada y uno de salida.

#### 1º Archivo: Demand.txt

Corresponde a la demanda que se le solicitará a los retailers, indicando los pedidos que se realizarán durante un período determinado. Cada archivo corresponderá a la demanda de 1 retailer. Este archivo tendrá el siguiente formato:



Figura 5.1: Archivo Demand.txt

#### 2º Archivo: Inventories.txt

Corresponde a la lista de todos los inventarios que estarán en el sistema, además de los datos de estos, los cuales son el nombre, política, si acepta backOrders, su proveedor, costo de almacenamiento, de falta de existencias, de pedido, de transporte, el lead time y el inventario inicial.

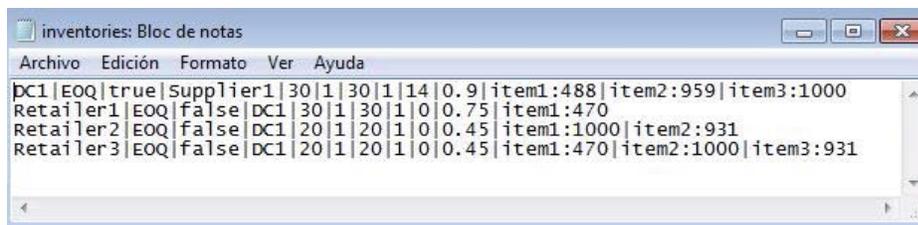
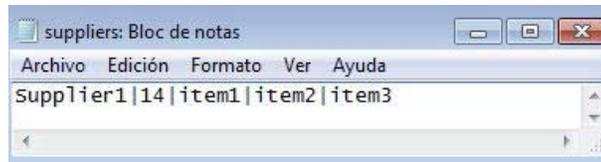


Figura 5.2: Archivo Inventories.txt

#### 3º Archivo: Suppliers.txt

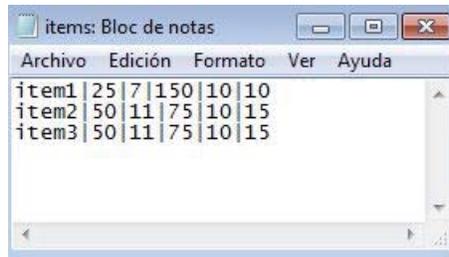
Corresponde a la lista de todos los suppliers o proveedores que estarán en el sistema, además del tiempo de entrega y los ítems que este contendrá.



**Figura 5.3: Archivo Suppliers.txt**

#### 4º Archivo: Items.txt

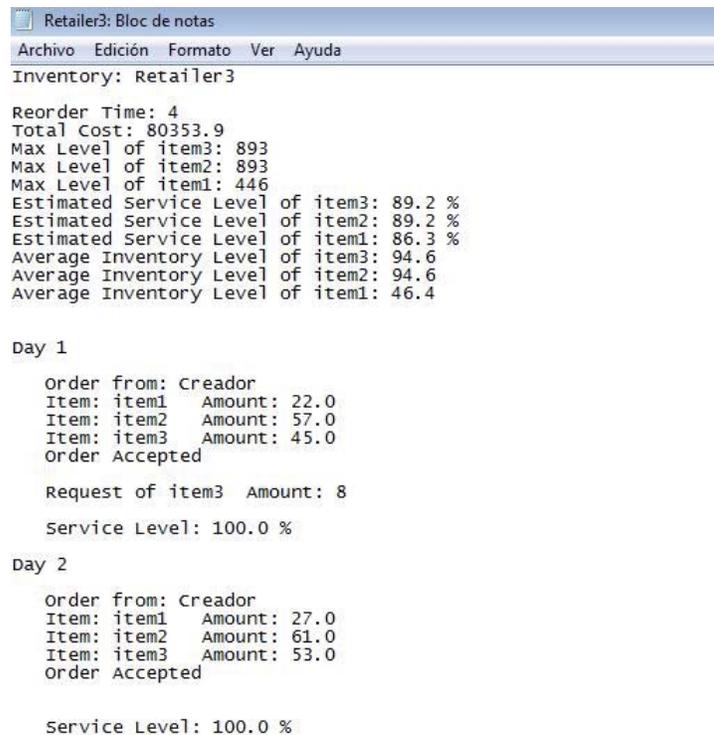
Corresponde a la información de los ítems con los que trabajará el sistema, teniendo los datos de la media de la demanda, la desviación estándar, el costo de mantenimiento, costo de adquisición y costo por falta de existencias.



**Figura 5.4: Archivo Items.txt**

#### 5º Archivo: Results.txt

Corresponde al archivo de salida que entregará los resultados de la simulación, como los costos totales, el nivel de servicio al cliente, y los movimientos de materiales que hubo cada día. Todos los inventarios tendrán un archivo de resultados.



**Figura 5.5: Archivo Results.txt**

## 5.2. Implementación del sistema

El sistema lee los 4 archivos de entrada, disponibles en la carpeta principal del sistema, para generar los agentes inventarios, ya sean Retailers, Centros de Distribución (creados ambos desde el InventoryAgent) o Proveedor (SupplierAgent).

Luego, existe también una carpeta llamada “demand”, que contiene los archivos de demanda de cada uno de los retailers del sistema, desde donde el Inventory Creator Agent sacará la información para hacer los pedidos. Luego, cada vez que un inventario necesite hacer un pedido a un inventario superior, este crea un Dealer Agent, que realiza el pedido al inventario, y guarda la información del tiempo restante (lead time) para entregar el pedido.

Una de las simplificaciones del sistema ha sido el hecho de tener una demanda media y desviación estándar constante para un ítem, sin importar el Retailer que lo esté manejando, para facilitar la construcción del sistema y las políticas de inventario. Otra simplificación corresponde a la aceptación de pedidos conjuntos, o de varios ítems a la vez, donde se ha decidido entregar sólo si se puede entregar completamente el pedido. Si sólo se tiene una parte del pedido en stock, se rechaza y se guarda como back order si el inventario tiene habilitada esta opción.

Además, se ha implementado una interfaz gráfica básica, que contiene información de cada inventario, como su lead time, su inventario superior, los datos obtenidos a través de la política de inventario, su nivel de servicio y los movimientos de ítems asociados a cada inventario, como los pedidos y las entregas de materiales. Una muestra de esta interfaz se ve en la Figura 5.6.

Sistema para problemas de inventario

Retailer3 DC1 Retailer2 Retailer1 Supplier 1

Inventory: Retailer 3

Lead Time: 0  
Supplier: DC1

Reorder Time: 4  
Total Cost: 80353.9  
Max Level of item3: 893  
Max Level of item2: 893  
Max Level of item1: 446

Estimated Service Level of item3: 89.2 %  
Estimated Service Level of item2: 89.2 %  
Estimated Service Level of item1: 86.3 %

Average Inventory Level of item3: 94.6  
Average Inventory Level of item2: 94.6  
Average Inventory Level of item1: 46.4

Service Level: 95.6 %

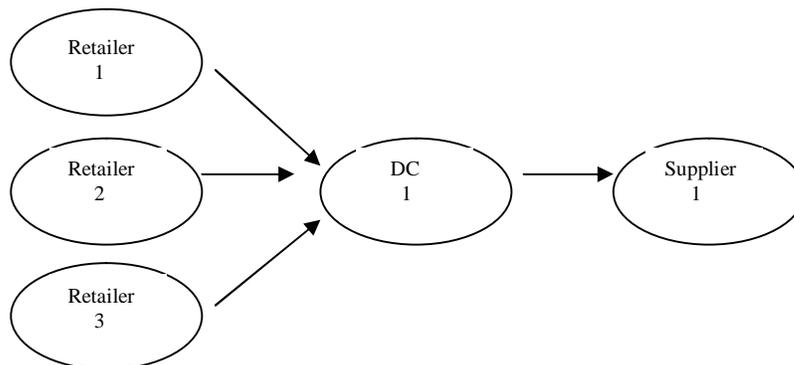
Day	Type	Customer	Item	Quantity	Accepted	Service Level
1	Order	Creador	item1	22.0		
	Order	Creador	item2	57.0		
	Order	Creador	item3	45.0		
	Inform	Creador			Pedido aceptado	
	Request	DC1	item3	8		100.0 %
2	Order	Creador	item1	27.0		
	Order	Creador	item2	61.0		
	Order	Creador	item3	53.0		
	Inform	Creador			Pedido aceptado	
						100.0 %
3	Order	Creador	item1	23.0		
	Order	Creador	item2	34.0		
	Order	Creador	item3	47.0		
	Inform	Creador			Pedido aceptado	
						100.0 %

Figura 5.6: Interfaz gráfica del sistema: Pestaña del Retailer 1

### 5.3. Resultados

La primera prueba que se realizó en el sistema fue la reproducción del ejemplo sobre pedidos conjuntos para el modelo de revisión periódica con demanda incierta que aparece en el libro de Ronald Ballou: “Logística: Administración de la cadena de suministro” [11]. A partir de este ejemplo se ha armado una cadena de suministro que consiste en 3 niveles, donde en el nivel más bajo (Retailer) se encuentran 3 inventarios, en el segundo nivel (Centro de Distribución) se encuentra un inventario, y en el nivel más alto (Proveedor) hay un inventario más. Cada Retailer contiene distintos ítems para ofrecer al cliente final: Retailer1 contiene al ítem1, Retailer 2 a los ítems 1,2 y Retailer 3 a los ítems 1, 2 y 3.

El profesor Guillermo Cabrera, de la Pontificia Universidad Católica de Valparaíso, decía que era muy difícil encontrar datos reales de este tipo, por lo que lo mejor era utilizar una demanda creada ficticia, y para eso se realizó el contacto con Vanessa Vivanco, Ingeniera Civil Informática de la PUCV, para conseguir un archivo de demanda aleatoria que ella poseía. Estos archivos, que crean una demanda ficticia para cada ítem según la media y desviación estándar, fueron utilizados para simular la demanda durante 1000 días. El resultado de la generación de demanda queda registrado en los archivos Retailer.txt de la carpeta demand, la cual es utilizada por el agente InventoryCreatorAgent.



**Figura 5.7: Cadena de Suministro de Prueba**

Los datos de los ítems (nombre, media, desviación estándar, costo de mantenimiento, costo de adquisición y costo por falta de existencias), se muestran en la Figura 5.8.

Ítem	Media	Desviación Estándar	Costo de Mantenimiento	Costo de Adquisición	Costo por falta de existencias
item1	25	7	150	10	10
item2	50	11	75	10	15
item3	50	11	75	10	15

**Figura 5.8: Ítems de la simulación de prueba.**

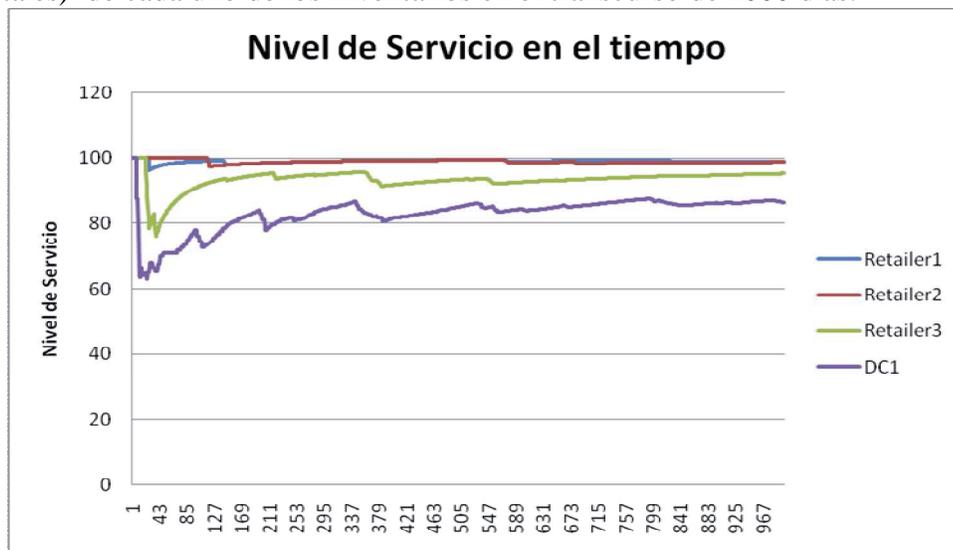
Y los datos de los Retailer se muestran en la Figura 5.9.

Inventory	EOQ	true/false	Supplier	30/1	30/1	14	0.9	item1	item2	item3
DC1	EOQ	true	Supplier1	30	1	30	1	14	0.9	item1:488 item2:959 item3:1000
Retailer1	EOQ	false	DC1	30	1	30	1	0	0.75	item1:470
Retailer2	EOQ	false	DC1	20	1	20	1	0	0.45	item1:1000 item2:931
Retailer3	EOQ	false	DC1	20	1	20	1	0	0.45	item1:470 item2:1000 item3:931

**Figura 5.9: Inventarios de la simulación de prueba**

Luego de que el sistema simulara mil días se han obtenido los siguientes resultados:

1. En el siguiente gráfico se muestra el Nivel de Servicio (pedidos aceptados / pedidos totales) de cada uno de los Inventarios en el transcurso de 1000 días:



**Figura 5.10: Nivel de servicio de la simulación de prueba.**

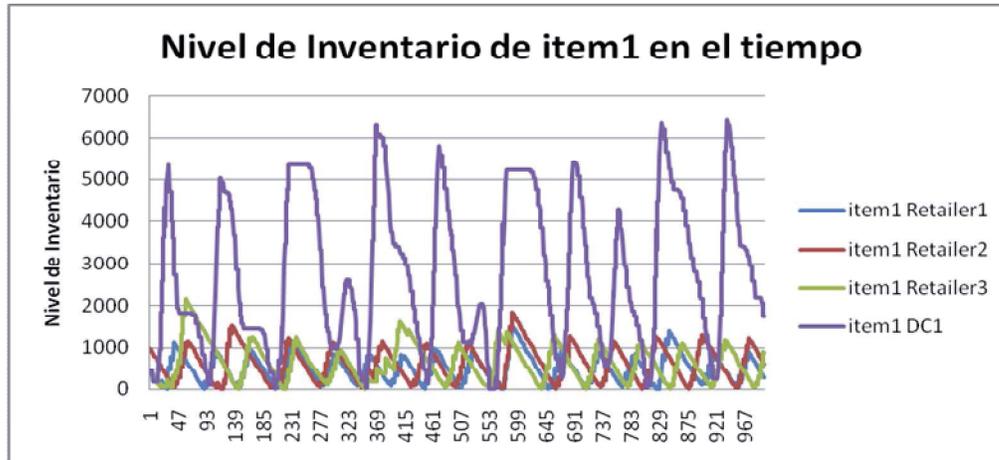
De aquí se puede ver que Retailer 1 es quien tiene un mayor nivel de Servicio (99%), Retailer 2 tiene un 98% y Retailer 3 un 95%. Esto se puede explicar debido a que Retailer 1 sólo recibe pedido de un solo ítem, por lo que las posibilidades de cumplir con el pedido son mucho mayores que los demás Retailers, que al poseer más variedad en los pedidos, tienen mayores posibilidades de no tener algunos de los productos, y por lo tanto, de rechazar el pedido. Todos estos inventarios cumplen e incluso mejoran el Nivel de Servicio estimado por la Política de Inventario, que había calculado un 96, 90 y 89% respectivamente.

Distinto es el caso del inventario mayorista (DC1) al cual se le había estimado un Nivel de Servicio sobre el 95%, pero que sin embargo tuvo un 86%, aunque de cualquier manera no impidió que los inventarios que interactúan directamente con el cliente tuvieran un alto nivel.

Se puede apreciar también, que durante los primeros 365 días, el nivel de servicio de los inventarios ha sido irregular y bastante bajo, dado que los primeros días los inventarios no tenían cantidad suficiente para satisfacer los pedidos, y debió esperar el lead time (en este caso 14 días entre el Supplier y DC1) para recibir los ítems necesarios.

A pesar de esto, el sistema ha logrado funcionar confiablemente en cuanto a nivel de servicio se trata.

2. En el siguiente gráfico se muestra el Nivel de los Inventarios para el ítem 1 en el transcurso de 1000 días:



**Figura 5.11: Nivel de inventario del ítem 1 de la simulación de prueba.**

En el gráfico se puede ver que el nivel de inventario varía considerablemente dependiendo del escalón en que se encuentra el inventario, ya que mientras más alto se encuentra en el escalafón, más ítems maneja. En este caso, el inventario DC1, que es el proveedor de los Retailers, tiene un nivel de inventario promedio considerablemente más alto que sus clientes.

Entre los Retailers, el inventario que sólo maneja pedidos de item1 (Retailer1) tiene un nivel de Inventario Promedio un poco menos a los demás (496.7 unidades contra 613 y 633.8 de los otros Retailers), que se puede explicar debido a que su Nivel de Servicio es mayor, lo que implica que se desabastece menos oportunidades, por lo que realiza menos pedidos en el tiempo, a diferencia de los otros Retailers que pueden tener un Nivel de Servicio un poco más bajo, por lo que en ocasiones pueden realizar pedidos más seguidos antes que les llegue el encargo realizado al inventario superior, por lo que en un momento, cuando lleguen todos esos pedidos, el inventario aumenta considerablemente.

De esta misma forma, el inventario DC1, además de manejar un mayor número de ítems ya que su demanda es mayor, también tiene un Nivel de Servicio menor, como se mostro en el apartado interior, por lo que también en un momento le llega una gran cantidad de pedidos al mismo tiempo.

## 6. Aspectos Finales

### 6.1. Conclusiones

En este trabajo se ha entregado el análisis, el diseño y la implementación del sistema de control de inventario, proyecto desarrollado para optar al título de Ingeniero Civil Informático.

Se ha logrado diseñar un sistema que cumple con uno de los objetivos principales de un sistema de inventario, que es el de entregar altos niveles de servicio al cliente, ya que las pruebas realizadas entregan resultados sobre el 95% para todos los inventarios que interactúan con el cliente, luego de ciclos de 1000 días de duración.

Aún así, existen problemas para que este sistema pueda ser una herramienta fiable para construir una verdadera cadena de suministro, ya que la política inventario utilizada (revisión periódica con demanda incierta multi-ítem), al ser una de las más clásica y elementales, no maneja de buena forma la cadena multinivel, produciéndose la distorsión de la demanda llamada efecto látigo, donde los niveles superiores de inventario reciben información de la demanda que no corresponde a la realidad del mercado, por lo que los centros de distribución en muchos momentos contaban con demasiados niveles de ítems que incrementan los gastos para la empresa.

A partir de esto se puede decir que se han completados los objetivos para este proyecto, sin embargo el sistema necesita una política más inteligente, que permita un mayor aprendizaje de la demanda, y así planificar de mejor medida la cantidad de pedido y los momentos en que se deben realizar. Hay muchas investigaciones que han buscado otras fórmulas para definir las políticas de inventario, como algunas basadas en heurísticas como tabu search y simulated annealing, redes neuronales y algoritmos genéticos, pero que sin embargo arrancaban de los límites de este proyecto.

Para el caso de la arquitectura, el modelo propuesto irá evolucionando en un proceso incremental de desarrollo, intentando agregar mejoras tanto funcionales como estéticas.

A pesar de haber buscado insistentemente datos de benchmark disponibles para probar un sistema de las características al que se está realizando, no se han podido encontrar datos que puedan comparar este sistema con otros, aunque se continuará con la búsqueda de estos. Hasta el momento, lo más cercano a estos datos han sido los solicitados vía e-mail a Sean P. Willems, autor de [18], pero que sólo contienen la distribución de los inventarios y las características de estos, mas no la demanda asociada. Otros estudios simplemente han creado su propia demanda de forma aleatoria, como en [19], la cual no ha sido publicada, lo cual dificulta comparar resultados.

La idea inicial del desarrollo de este sistema era encontrar una propuesta de diseño de un sistema multiagente para inventario ya realizada, y desde esta base comenzar a planificar mejoras y diseñar este sistema en PASSI, pero fue bastante difícil encontrar trabajos que cumplan con los requerimientos, ya que a pesar de que se han realizado proyectos, la

documentación encontrada no entrega demasiada información sobre los modelos, además de que normalmente sólo se nombran las políticas de inventario o las heurísticas utilizadas para el control de estos, sin encontrar algoritmos que puedan funcionar como base, lo que hizo reconsiderar el propósito del sistema, empezando a modelar desde cero. Existen algunas librerías en lenguaje JAVA para el desarrollo de cadenas de suministro en general, e inventarios en particular, desde donde sacaron ideas sobre la forma en que los agentes interactúan y la arquitectura de estos, sin embargo, no se reutilizó código.

Se eligió la arquitectura de agentes para resolver este problema por las razones descritas en el capítulo anterior, pero también porque PASSI, que se basa en la utilización de UML, tiene una forma bastante intuitiva para trabajar, a diferencia de otras metodologías como Gaia o MaSE, donde se hubiese gastado demasiado tiempo el sólo comprenderlas. Además, haber cursado el curso de sistemas multiagente entregó las bases para el desarrollo de un sistema con estas características

Dado que la comunicación entre los agentes fue implementada en una primera fase, luego la generación de demanda en una segunda fase, y la luego la creación de la política de inventario fue implementada en la tercera fase del proyecto, el riesgo fue controlado, corrigiendo los problemas suscitados en cada fase como problemas aislados, lo que los hizo mucho más manejables que si se hubiesen tratado todos juntos como podría haber sido si se hubiese utilizado una Metodología de Desarrollo más rígida, como puede ser la Metodología de Cascada.

## **6.2. Trabajo Futuro**

Los trabajos pendientes se orientan y basan principalmente en la mejora al sistema implementado, donde se debe ver principalmente la implementación de una política que trate más eficazmente la cantidad y el período de los pedidos, además del tratamiento de los back orders.

Además, se debe trabajar en la interfaz gráfica del sistema, que ha sido realizada de forma bastante primitiva, que muestra de una forma más amigable los hechos que acontecen en la simulación.

Por último, y como mejora más avanzada, se debe incorporar al manejo de inventarios el problema asociado al transporte de los pedidos, incorporando datos de rutas para encontrar el mejor camino a realizar.

## 7. Referencias

- [1] S. Russel and P. Norvig. “Artificial Intelligence: A Modern Approach”. Prentice Hall, 2nd edition, 2003.
- [2] Wooldridge, M. en “Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence”. Ed. G. Weiss, MIT Press, Massachusetts, USA. 1999.
- [3] M. Wooldridge and N. R. Jennings. “Intelligent agents: Theory and practice”. The Knowledge Engineering Review, 10(2):115–152, 1995.
- [4] Y. Shoham. “Agent-oriented programming”. Artificial Intelligence, 60(1):51–92, Mar. 1993.
- [5] Weiss, G. “Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence”. MIT Press, Massachusetts, USA. 1999.
- [6] Katia P. Sycara, “Multiagent Systems”. AI Magazine Volume 19 Number 2, 1998.
- [7] R. Rizo, F. Llorens, M. Pujol, “Arquitecturas y comunicación entre agentes”. Universidad de Alicante.
- [8] J. Gómez Sanz, “Metodologías para el desarrollo de sistemas multi-agente”. Revista Iberoamericana de Inteligencia Artificial, 2003.
- [9] M. Cossentino, L. Sabatucci, “Modeling Notation Source PASSI”. Foundation for Intelligent Physical Agents. Abril 2003.
- [10] The PASSI Design Process.
- [11] R. H. Ballou, “Logística: Administración de la cadena de suministro”. Prentice Hall. 5ª Edición, 2004.
- [12] T. Moyaux, “Supply Chain Management and Multiagent Systems: An Overview”. Universit Laval, Dpt. d’Informatique et de Gnie Logiciel, Quebec, Canada.
- [13] H. Chen, “Using Artificial Neural Network in Multi-Agent Supply Chain Systems”. Department of Industrial Engineering Chung Yuan Christian University. Taiwan.
- [14] Charro, Santos y Reis, “Applying Multi-Agent simulation to supply chains”. Department of Information Systems, ISCTE.
- [15] S. Saberi, “Multi agent system for negotiation in supply chain management”. The 6th International Conference on Manufacturing Research (ICMR08), Brunel University, UK, September 2008

- [16] M. Rosetti, "An object-oriented framework for simulating multi-echelon inventory systems". Winter Simulation Conference, 2006.
- [17] S. Yung, "Applying Multi-Agent Technology To Supply Chain Management". Journal of Electronic Commerce Research, VOL. 1, NO. 4, 2000.
- [18] S. Willems, "Real-World Multiechelon Supply Chains Used for Inventory Optimization". Manufacturing & Service Operations Management, Vol. 10, No. 1, Winter 2008. Data Set disponible en <http://supplychain.bu.edu/>
- [19] R. Sarker, "A Multi-agent Simulation Study for Supply Chain Operation". Computational Intelligence for Modelling, Control and Automation, Volume 1, Issue , 28-30 Nov. 2005.
- [20] C. Iglesias, "Fundamentos de los Agentes Inteligentes". Universidad Politécnica de Madrid.
- [21] D. Ferreiro, "Estándar FIPA - Foundation for Intelligent Physical Agents", Universidad de Vigo, 2002.
- [22] "Selecting a Development Approach", Centers for Medicare & Medicaid Services, Feb. 2005.