

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

**Desarrollo de un Sistema Domótico
con Agentes Inteligentes y Robots
LEGO.**

Nicol Ailym González Allendes.

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

MARZO 2013

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

**Desarrollo de un Sistema Domótico
con Agentes Inteligentes y Robots
LEGO.**

Nicol Ailym González Allendes.

Profesor Guía: **Claudio Cubillos Figueroa.**

Profesor Correferente: **Wenceslao Palma Muñoz.**

Carrera: **Ingeniería de Ejecución en Informática**

MARZO 2013

*Dedicado a mi familia y amigos
por todo el apoyo en esta etapa.*

Índice

GLOSARIO DE TÉRMINOS	IV
LISTA DE ABREVIATURAS O SIGLAS.....	VII
LISTA DE FIGURA.....	VIII
LISTA DE TABLAS.....	XI
RESUMEN	XII
ABSTRACT	XIII
1 DESCRIPCIÓN DEL PROYECTO.....	1
<i>1.1 Introducción.....</i>	<i>1</i>
<i>1.2 Objetivos</i>	<i>2</i>
<i>1.2.1 Objetivo General.....</i>	<i>2</i>
<i>1.2.2 Objetivos Específicos</i>	<i>2</i>
<i>1.3 Plan de Trabajo Tentativo</i>	<i>2</i>
<i>1.3.1 Metodología</i>	<i>3</i>
<i>1.3.1.1 Agile UP.</i>	<i>3</i>
2 DOMÓTICA.....	4
<i>2.1 Componentes Básicos</i>	<i>4</i>
<i>2.1.1 Sensores.....</i>	<i>5</i>
<i>2.1.2 Actuadores.....</i>	<i>6</i>
<i>2.1.3 Interface</i>	<i>8</i>
<i>2.1.4 Infraestructura</i>	<i>9</i>
<i>2.1.5 Unidad de Control.....</i>	<i>10</i>
<i>2.2 Servicios a Gestionar</i>	<i>12</i>
<i>2.2.1 Gestión de Energía.....</i>	<i>12</i>
<i>2.2.2 Gestión de Confort</i>	<i>13</i>
<i>2.2.3 Gestión de la Seguridad.....</i>	<i>14</i>
<i>2.2.3.1 Tipos de sistemas de seguridad.....</i>	<i>15</i>
<i>2.2.4 Gestión de las Comunicaciones</i>	<i>15</i>
<i>2.2.5 Gestión del Entretenimiento.....</i>	<i>16</i>
<i>2.2.6 Gestión de Servicios para Discapacitados.....</i>	<i>16</i>
<i>2.3 Estándares.....</i>	<i>16</i>
<i>2.3.1 BatiBus</i>	<i>16</i>
<i>2.3.2 CEBus.....</i>	<i>17</i>
<i>2.3.3 EHS.....</i>	<i>17</i>
<i>2.3.4 EIB.....</i>	<i>17</i>
<i>2.3.5 Konnex.....</i>	<i>18</i>

2.3.6	<i>LonWorks</i>	18
2.3.7	<i>X-10</i>	19
2.4	<i>Trabajos Relevantes</i>	19
2.4.1	<i>Claudio Jara, “Arquitectura Domótica utilizando dispositivos X10 y comunicación mediante Web Services”</i>	19
2.4.2	<i>Juan P. Cofré, Gabriel Moraga, “Usabilidad y Accesibilidad Aplicada a Herramientas Domóticas”</i>	22
2.4.2.1	<i>Solución Propuesta</i>	22
2.4.2.1.1	<i>Diagramado</i>	23
3	ROBOTS LEGO MINDSTORMS NXT	24
3.1	<i>Características del Bloque Lego Mindstorms NXT</i>	24
3.2	<i>Micro-controlador</i>	24
3.3	<i>Sensores</i>	25
3.4	<i>Programación</i>	26
3.4.1	<i>ICommand</i>	26
3.4.2	<i>Lejos NXJ</i>	26
3.4.3	<i>RobotC</i>	28
4	AGENTES INTELIGENTES Y SISTEMAS MULTIAGENTE	29
4.1	<i>Agentes</i>	29
4.2	<i>Arquitectura de Agentes</i>	30
4.3	<i>Infraestructura Agente</i>	31
4.4	<i>Comunicación entre agentes</i>	32
4.4.1	<i>Niveles y Protocolos de Comunicación</i>	33
4.5	<i>Sistemas Multiagente</i>	33
4.6	<i>Metodologías de desarrollo de Sistemas Multiagente</i>	35
4.6.1	<i>BDI</i>	35
4.6.2	<i>Gaia</i>	36
4.6.3	<i>MASE</i>	37
4.6.4	<i>PASSI</i>	38
4.7	<i>Plataformas Multiagente</i>	40
4.7.1	<i>JADE</i>	40
4.7.2	<i>FIPA-OS</i>	41
5	PROPUESTA DE DESARROLLO	42
5.1	<i>Modelado del Sistema</i>	43
5.1.1	<i>Modelo de Requerimientos del Sistema</i>	43
5.1.1.1	<i>Modelo de Descripción del Dominio</i>	43
5.1.1.2	<i>Modelo de Identificación de Agentes</i>	44
5.1.1.3	<i>Modelo de Identificación de Roles</i>	45

5.1.1.4	Modelo Especificación de Tareas	47
5.1.2	<i>Modelo de Sociedad de Agentes</i>	48
5.1.2.1	Modelo de Descripción de la Ontología del Dominio	48
5.1.2.2	Modelo de Descripción de la Ontología de Comunicación	49
5.1.2.3	Modelo de Descripción de Roles	50
5.1.2.4	Modelo de Descripción de Protocolos	51
5.1.3	<i>Modelo de Implementación de Agentes</i>	52
5.1.3.1	Modelo de Definición de Estructura de Agentes	52
5.1.4	<i>Modelo de Despliegue</i>	55
5.1.4.1	Modelos de Configuración de Despliegue	55
5.2	<i>Esquema del Sistema</i>	56
5.3	<i>Caso de Estudio</i>	56
5.3.1	<i>Caso de Estudio: Confort</i>	57
5.3.2	<i>Caso de Estudio: Seguridad</i>	57
6	IMPLEMENTACIÓN DE LA PROPUESTA	59
6.1	<i>Creación de la Maqueta para presentar el caso de Estudio</i>	59
6.2	<i>Herramientas a utilizar</i>	60
6.2.1	<i>Eclipse</i>	60
6.2.1.1	LeJOS y Eclipse	61
6.2.1.2	Eclipse, LeJOS y Jade	62
6.3	<i>Comunicación entre PC y NXT-Project</i>	63
6.4	<i>Desarrollo del prototipo</i>	64
7	CONCLUSIONES	67
8	REFERENCIAS BIBLIOGRÁFICAS	68

Glosario de Términos

Actuador: son dispositivos que actúan sobre el medio exterior y afectan físicamente la vivienda. Los actuadores logran convertir una magnitud eléctrica en otro tipo, como puede ser mecánica o térmica, realizando un proceso inverso al que realizan los sensores. Algunos ejemplos de actuadores pueden ser por ejemplo el motor de las persianas, los contactos de un circuito de iluminación, entre otros.

Agente: un agente se entiende como Toda entidad que de forma autónoma perciba su entorno (real o simulado) mediante sensores y que actúe en el mismo mediante efectores.

Agentes Inteligentes: un agente inteligente se define como todo ente que es capaz de realizar acciones autónomas y flexibles en la búsqueda de objetivos para los cuales fue diseñado.

Algoritmos: secuencia de pasos, los cuales se ejecutan consecutivamente para conseguir algún tipo de resultado o tarea. Estos deben cumplir tres condiciones, 1.- debe ser preciso e indicar el orden de cada paso, 2.- debe estar definido. Si se sigue el algoritmo dos veces, se obtiene el mismo resultado, 3.- debe ser finito. El algoritmo debe terminar en un momento.

Bluetooth: Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.

Domótica: Conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda.

Driver: llamado normalmente controlador es un programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz para usarlo.

JAVA: Lenguaje de programación orientado a objetos. Con él se pueden desarrollar aplicaciones para distintos tipos de dispositivos y plataformas.

Lego Mindstorms NXT: kit de robótica desarrollado por la empresa Lego los cuales pueden ejecutar tareas a través de sus actuadores y percibir el ambiente a través de sus sensores.

LeJOS NXJ: Firmware y conjunto de utilidades y clases Java que conforman un entorno de programación Java completo, para escribir programas orientado a objetos para Lego Mindstorms NXT.

Metodologías: conjunto de procedimientos racionales utilizados para alcanzar objetivos que rigen en una investigación científica, una exposición doctrinal o tareas que requieran habilidades, conocimientos o cuidados específicos.

Metodología Ágil: constituyen un nuevo enfoque en el desarrollo de software, debido a la simplicidad de sus reglas y prácticas, orientada a equipos de desarrollo pequeños, flexible ante cambios y con una ideología de colaboración

Ontología: Existen numerosas definiciones de ontologías, entre las que cabe destacar: "Una ontología es un vocabulario acerca de un dominio: términos, relaciones, reglas de combinación para extender el vocabulario". Neches, 1991. "Una ontología es la especificación de una conceptualización". Gruber, 1993. "Una ontología es una especificación formal de una conceptualización compartida". Borst, 1997. "Una ontología es una base de datos que describe los conceptos generales o sobre un dominio, algunas de sus propiedades y cómo los conceptos se relacionan unos con otros". Weingand, 1997. Una ontología necesariamente incluirá un vocabulario de términos y una especificación de su significado que impone estructura al dominio y restringe las posibles interpretaciones. Uschold-Jasper.

Plugin: complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

Protocolos: En informática, un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red por medio de intercambio de mensajes. Ésta es una regla o estándar que controla o permite la comunicación en su forma más simple, puede ser definido como las reglas que dominan la sintaxis, semántica y sincronización de la comunicación.

Rational Unified Process (RUP): proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

Robot: su origen viene de la palabra checa robota. Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas.

Sensor: es básicamente un dispositivo que detecta una determinada acción (temperatura, presión, etc.), y la transmite adecuadamente.

Sistemas Multi-Agente: se puede entender como una red de problem-solvers que trabajan conjuntamente para encontrar respuestas a problemas que van más allá de las capacidades o conocimientos individuales de cada entidad.

Sistemas Multi-Robot: Sistema compuesto por dos o más robot, los cuales trabajan en conjunto para alcanzar un objetivo en común.

Universal Serial Bus: estándar industrial desarrollado en los años '90 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos y dispositivos electrónicos.

Lista de Abreviaturas o Siglas

- MAS : Multi-Agent Systems
- RUP : Rational Unified Process
- Ágil UP: Ágil Unified Process
- PASSI : Process for Agent Societies Specification and Implementation
- RAE : Real Academia de la lengua Española
- IA : Inteligencia Artificial
- ACL : Agent Communication Language
- AIP : Protocolos de Interacción
- FIPA : Foundation for Intelligent Physical Agents
- MaSE : Multi agent systems Software Engineering
- UML : Unified Modeling Language
- JADE : Java DEvelopment Framework
- FIPA-OS: FIPA Open Source
- USB : universal serial bus
- SDK : Software Developer Kit
- HDK : Hardware Developer Kit
- BDK : Bluetooth Developer Kit
- JDK : Java Development Kit

Lista de Figura

Figura 1 Ejemplo de Dispositivos de Sistemas de Domótica [CasaDomo, 1999]	4
Figura 2 Sensores [Birtingenio, 2011].....	5
Figura 3 Actuadores: Relé [Radioelectronica, 2011].....	7
Figura 4 Actuadores: Contactores [Moeller, 2011]	7
Figura 5 Actuadores: Reguladores o Dimmers [Shoptronica, 2011].....	7
Figura 6 Actuadores: Electroválvulas [Hogartintorero, 2011], [Directindustry, 2011]	8
Figura 7 Actuador: Motores Eléctricos [Mahersolhogar, 2011].....	8
Figura 8 Actuador: Resistencia Eléctrica [Electromoda, 2011][Solac, 2011].....	8
Figura 9 Interfaz [[Dte.Upct, 2011].....	8
Figura 11 Topología: Bus [Elab. Propia]	9
Figura 11 Topología: Centralizada [Elab. Propia].....	10
Figura 12 Topología: Mixta [Elab. Propia]	10
Figura 13 Modelo de la Arquitectura [C. Jara, 2009].....	20
Figura 14 Diagrama de Clase [C. Jara, 2009].....	21
Figura 15 Diagrama de Clases [J.P.Cofré, 2010].	23
Figura 17 Prototipo Interfaz [J.P.Cofré, 2010].....	23
Figura 17 Brick RCX y NXT [LegoImag, 2001]	25
Figura 18 Sensor de Luz [LegoImag, 2001].....	25
Figura 19 Sensor de Ultrasonido [LegoImag, 2001]	25
Figura 20 Sensor de Tacto [LegoImag, 2001].....	26
Figura 21 Sensor de Sonido [LegoImag, 2001].....	26
Figura 22 Agente [Erbol, 2011].....	30
Figura 23 Arquitectura Deliberativa [Iglesias, 1998].....	30
Figura 24 Arquitectura Reactivas [Iglesias, 1998]	31
Figura 25 Arquitectura Híbridos [Iglesias, 1998].....	31
Figura 26 Esquema del ciclo mental interno de un agente BDI [Dccia, 2011]	36
Figura 27 Organización de la Metodología Gaia [Upv, 2011]	37
Figura 28 Fases de MaSE [Ana Mas. 2005].....	38
Figura 29 Metodología Passi: Modelos y fases [Icar, 1923]	39
Figura 30 Prototipo de Escenario [Home, 2011].....	42
Figura 31 Modelo de Descripción del Dominio [Elab. Propia].....	43
Figura 32 Modelo Identificación de Agentes [Elab. Propia].....	44
Figura 33 Modelo de Identificación de Roles. Escenario: Ambiente Manual [Elab. Propia].....	45

Figura 34 Modelos Identificación de Roles. Escenario: Ambiente Automático [Elab. Propia].....	46
Figura 35 Modelos Especificación de Tareas: Sensor [Elab. Propia]	47
Figura 36 Modelos Especificación de Tareas: Actuador [Elab. Propia]	47
Figura 37 Modelos Especificación de Tareas: Comportamiento [Elab. Propia]	48
Figura 38 Modelo de Descripción de la Ontología del Dominio [Elab. Propia]	48
Figura 39 Modelo de la Ontología de la Comunicación [Elab. Propia]	49
Figura 40 Modelo de Descripción de Roles [Elab. Propia]	50
Figura 41 Modelo de Descripción de Protocolos [Elab. Propia]	51
Figura 42 Modelo de la Definición de Estructura Multiagente [Elab. Propia]	52
Figura 43 Modelo de la Descripción de la Estructura del Agente Sensor [Elab. Propia].....	53
Figura 44 Modelo de la Descripción de la Estructura del Agente Comportamiento [Elab. Propia]	53
Figura 45 Modelo de la Descripción de la Estructura del Agente GUI [Elab. Propia]	54
Figura 46 Modelo de la Descripción de la Estructura del Agente Actuador [Elab. Propia].....	54
Figura 47 Modelo Configuración de Despliegue [Elab. Propia]	55
Figura 48 Idea de Futura Implementación [Home, 2011]	56
Figura 49 Caso de Estudio: Confort [Elab. Propia].....	57
Figura 50 Caso de Estudio: Seguridad – Alarma [Elab. Propia]	58
Figura 51 Prototipo 1 de maqueta [Elab. Propia]	59
Figura 52 Prototipo 1 de maqueta [Elab. Propia]	59
Figura 53 Maqueta Final Frontis [Elab. Propia].....	59
Figura 54 Maqueta Final Interior con la luz encendida	Figura 55 Maqueta
Final con la puerta abierta.....	60
Figura 56 Ubicación de los sensores y actuadores dentro del prototipo.	60
Figura 57 Eclipse y LeJOS	61
Figura 58 Código de conexión con el NXT.....	62
Figura 59 Código de espera de conexión del NXT	62
Figura 60 Configuración de Jade.....	63
Figura 61 Jar de Jade	63
Figura 62 Interfaz del Prototipo - Control Automático	65
Figura 63 Interfaz del Prototipo - Control Manual.....	65
Figura 64 Interfaz del Prototipo - Estado Actual del Hogar.....	65
Figura 65 Escenario básico Uno.....	66
Figura 66 Escenario básico Dos.	66
Figura 67 Mensaje emitido al abrir la puerta.....	66
Figura 68 Ingresar la clave	66

Figura 69 Mensaje emitido al aceptar la clave 66

Lista de Tablas

Tabla 1 Plan de Trabajo Tentativo	3
Tabla 2 Fases de AUP [AgileUP, 2011].....	3
Tabla 3 Tipos de Sensores según Alimentación [Romero C., 2005].....	5
Tabla 4 Tipos de Sensores según el Tipo de Señal Implicada [Romero C., 2005]	6
Tabla 5 Tipos de Sensores según el Ámbito de la Aplicación [Romero C., 2005]	6
Tabla 6 Códigos enviados del robot al PC	64
Tabla 7 Códigos enviados del PC al robot	64

Resumen

La domótica es una tecnología que está en constante crecimiento, y ofrece innumerables ventajas en la vida diaria de las personas, ya que no sólo les brinda seguridad a sus hogares, sino que también les ofrece un mayor nivel de confort. Existen variadas formas para poder desarrollar un prototipo domótico, para este proyecto se trabajara con la robótica, mas puntualmente con los robots Lego Mindstorms. Además, para poder darle más autonomía y flexibilidad al prototipo a desarrollar, se le agregará la tecnología de Agentes, pudiendo así crear diversos ambientes dependiendo las condiciones ambientales.

En este informe se presentará ¿qué es la domótica?, sus principales características y componentes. Además de revisar la robótica y las principales características de los Lego Mindstorms, y los Sistemas Multiagentes. Luego se presentará el diseño del prototipo de sistema, el desarrollo del sistema y finalmente los casos de estudio a realizar.

Palabras Clave: Domótica, Robots Lego, Sensores, Sistemas Multi-Agente.

Abstract

Home automation is a technology that is constantly growing and offers many advantages in the daily life of people, as it not only gives them safety to their homes, but also offers a higher level of comfort. There are various ways to develop a prototype home automation, for this project will work with robotics, more timely with the Lego Mindstorms robots. In addition, to give more autonomy and flexibility to develop the prototype, it will add the Agent technology, thus being able to create different environments depending on environmental conditions.

This report will present what is home automation?, Its main characteristics and components. In addition to reviewing the robotics and the main characteristics of the Lego Mindstorms, and Multi-Agent Systems. Then introduce the prototype system design, system development and finally the case studies to be performed.

Keywords: Home Automation, Lego Robots, Sensors, Multi-Agent System.

1 Descripción del Proyecto

1.1 Introducción

Hoy en día las personas se apoyan en la tecnología para poder solucionar o facilitar su vida en diversos ámbitos, como lo es en la salud, la educación, entre otras. Uno de los ámbitos que se ha buscado facilitar es el confort de las personas para poder aumentar su comodidad y bienestar. Esto se puede lograr con la “Domótica”, palabra proveniente de la unión de las palabras “Domo” que en latín (Domus) significa “casa” y el sufijo “tica” que proviene de la palabra “automática”, es decir, “Home Automation” o “Casa Automática” [Romero C., 2005] . Cabe mencionar que aparte de la domótica también existe la inmótica, un término algo desconocido que hace referencia a la gestión técnica de los edificios, y su mayor diferencia con la domótica, es que está enfocada a ambientes unifamiliares, y abarca edificios más grandes, con distintos fines específicos.

La finalidad de este trabajo es desarrollar un prototipo de un sistema de automatización dirigido a un espacio del hogar, en el cual se pretende poder automatizar el sistema de iluminación del lugar. Para desarrollar este prototipo de sistema se utilizarán los elementos disponibles en la Escuela de Ingeniería Informática de la Pontificia Universidad Católica de Valparaíso. La Escuela dispone de kit de Robots LEGO MINDSTORMS NXT [LEGO, 2006]. Entre los componentes de este kit, se pueden encontrar los sensores, que emulan los sentidos humanos; y los motores, los cuales son el medio para desarrollar las acciones. Estos se conectan a un microprocesador, el cual se encargará de procesar, comunicar y además ejecutar nuestro sistema.

Además de la robótica, se necesita la ayuda de otro tipo de tecnologías para poder complementar el prototipo. Como la finalidad de este trabajo es dar cierta autonomía al hogar, se apoyará de la tecnología de Agentes inteligentes y de esta forma se creará un sistema multiagente. En informática por agente se entiende: “Toda entidad que dé forma autónoma percibe su entorno mediante sensores y actúa en el mismo mediante efectores” [S. Russel, 2003]. La tecnología de agentes permite dar autonomía y flexibilidad al prototipo desarrollado y es adaptable a cualquier tipo de tecnología, como lo es la ingeniería de software, la robótica, las telecomunicaciones e inclusive el comercio electrónico.

El desarrollo de este trabajo se dividió en 5 partes: en la primera se verá que es la domótica, sus componentes básicos, sus beneficios, entre otras cosas. En segundo lugar, estarán los Robots Lego, como se componen, sus características y funcionalidades. En tercer lugar se verá que son los sistemas multiagente, como se componen, y como trabajan. En cuarto lugar se verá el diseño del prototipo de sistema que se está desarrollando. Y por último se verá el desarrollo de la programación de los casos de estudio.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un prototipo de sistema domótico utilizando robots lego y sistemas multiagente.

1.2.2 Objetivos Específicos

- ❖ Comprender las capacidades del robot lego en el área de la domótica.
- ❖ Comprender los sistemas multiagente para el control de los robots en un sistema domótico.
- ❖ Generar el diseño del sistema domótico con robots lego y sistemas multiagente.
- ❖ Implementar el sistema domótico en un caso de estudio con robots lego y sistemas multiagente.
- ❖ Diseñar un escenario de aplicación domótico para validar el sistema.

1.3 Plan de Trabajo Tentativo

El plan de trabajo es esencial en cualquier desarrollo de un proyecto, en primer lugar porque este organiza las metas propuestas para el proyecto y además realiza un seguimiento, asegurándose que se cumplan todas y cada una de las tareas especificadas a desarrollar en los plazos ya determinados, pues depende de estos plazos el éxito o fracaso del proyecto.

Para desarrollar este proyecto se utilizará una metodología de desarrollo AUP, ya que esta se acomoda de mejor manera a la planificación del proyecto, y además otorga las herramientas necesarias para poder controlar el orden en las tareas y el seguimiento de éstas.

El trabajo se divide en 3 partes:

1. En la primera etapa se pretende revisar la información encontrada y rescatar lo más importante y útil que se relacione con las características del problema, además de ver los aspectos teóricos de la domótica. Tiempo estimado: 6 semanas.
2. Con la información de la etapa anterior se definirán la plataforma y el lenguaje de programación a utilizar en el proyecto, y se desarrollarán el modelo de requerimientos del sistema, la de implementación de agentes y el diseño del desarrollo del sistema domótico, se llegará a un diseño final para su futura implementación. Tiempo esperado 5 semanas.
3. En base a los modelos obtenidos se desarrollará un sistema domótico, se revisará su funcionamiento y se realizarán las correcciones pertinentes en caso de ser necesarias. Tiempo esperado 5 meses.

A continuación se presenta un plan de trabajo tentativo para el desarrollo del presente proyecto.

Tabla 1 Plan de Trabajo Tentativo

Actividad	Tiempo
1 Obtener los estados de arte sobre domótica, robots Lego Mindstorms NXT y Sistemas Multiagente.	6 semanas
2 Definir la plataforma y lenguaje de programación a utilizar en el proyecto.	3 semanas
3 Diseño del desarrollo del sistema domótico	2 semanas
4 Realizar sistema domótico	14 semanas
5 Revisión y correcciones.	6 semanas

1.3.1 Metodología

La metodología que se utilizará para desarrollar este proyecto es: Agile Unified Process (AUP).

1.3.1.1 Agile UP.

AUP es una versión más simplificada del Rational Unified Process (RUP) de IBM. Este describe un enfoque simple y fácil de entender para el desarrollo de un software usando técnicas y conceptos que aún se mantienen vigentes en el RUP [AgileUP, 2011].

El AUP está basado en los siguientes principios: El personal sabe lo que está realizando, simplicidad, agilidad, enfocar las actividades de alto nivel, independencia de herramientas y poder adaptar este producto para satisfacer las necesidades propias.

AUP, posee cuatro fases que van transcurriendo de forma consecutiva y que acaban con hitos claros alcanzados, y son:

Tabla 2 Fases de AUP [AgileUP, 2011]

Fase	Meta	Hitos
Inicio	Identificar el alcance del proyecto y diseñar la arquitectura inicial del sistema.	Objetivos de Ciclo de Vida (LCO)
Elaboración	Comprensión de los requisitos y validar la arquitectura del sistema	Arquitectura del Ciclo de Vida (LCA)
Construcción	Desarrollo del sistema y prueba completa en el ambiente de desarrollo.	Capacidad Operacional Inicial (OIC)
Transición	Pruebas de validación y aceptación, y desplegar en los sistemas de producción.	Liberación del Producto (PR)

2 Domótica

Las personas cada día buscan una mejor calidad de vida, un hogar con mayor confort y comodidad, bajo esta necesidad nace la *domótica*. La Real Academia Española (RAE) la define como: *Conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda* [Rae, 1714]. El objetivo principal que posee radica en poder satisfacer distintas necesidades que tiene el hombre con su entorno, entre algunas cosas se puede destacar esta la optimización de la energía, aumento de la seguridad, o dicho de otro modo ayuda a vivir mejor [Domotica.tk, 2011].

Encontrar donde nace exactamente la domótica es bastante complejo, ya que esta no es un hecho concreto, más bien viene de una evolución, que se calcula aparece en un momento clave de la humanidad, la transición entre el siglo XX y XXI [D. Usuarios, 2011]. Se dice que comenzó con las redes de control de los edificios inteligentes y se ha ido adaptando a las necesidades propias de las viviendas.

2.1 Componentes Básicos

Los componentes básicos de la domótica describen algunos de los elementos que deben estar presente en cualquier instalación, como lo son los sensores, actuadores, interfaces, infraestructura y unidad de control [Domotica.tk, 2011].

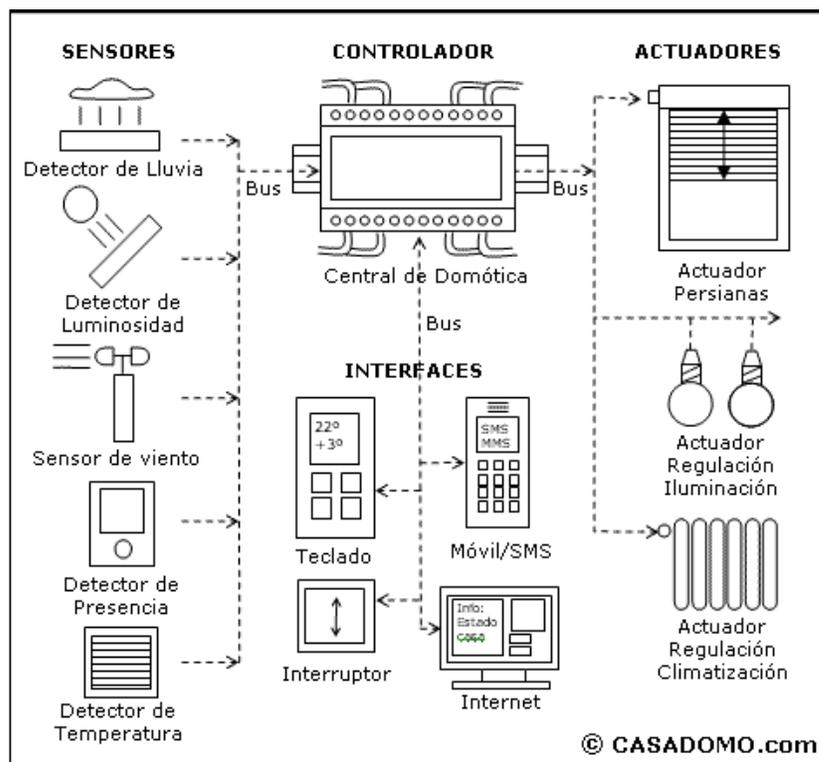


Figura 1 Ejemplo de Dispositivos de Sistemas de Domótica [CasaDomo, 1999]

2.1.1 Sensores

Según la RAE, un sensor es básicamente un dispositivo que detecta una determinada acción (temperatura, presión, etc.), y la transmite adecuadamente [Rae, 1714]. En una instalación domótica los sensores son los encargados de proporcionar toda la información que sea necesaria para luego poder ser gestionada.

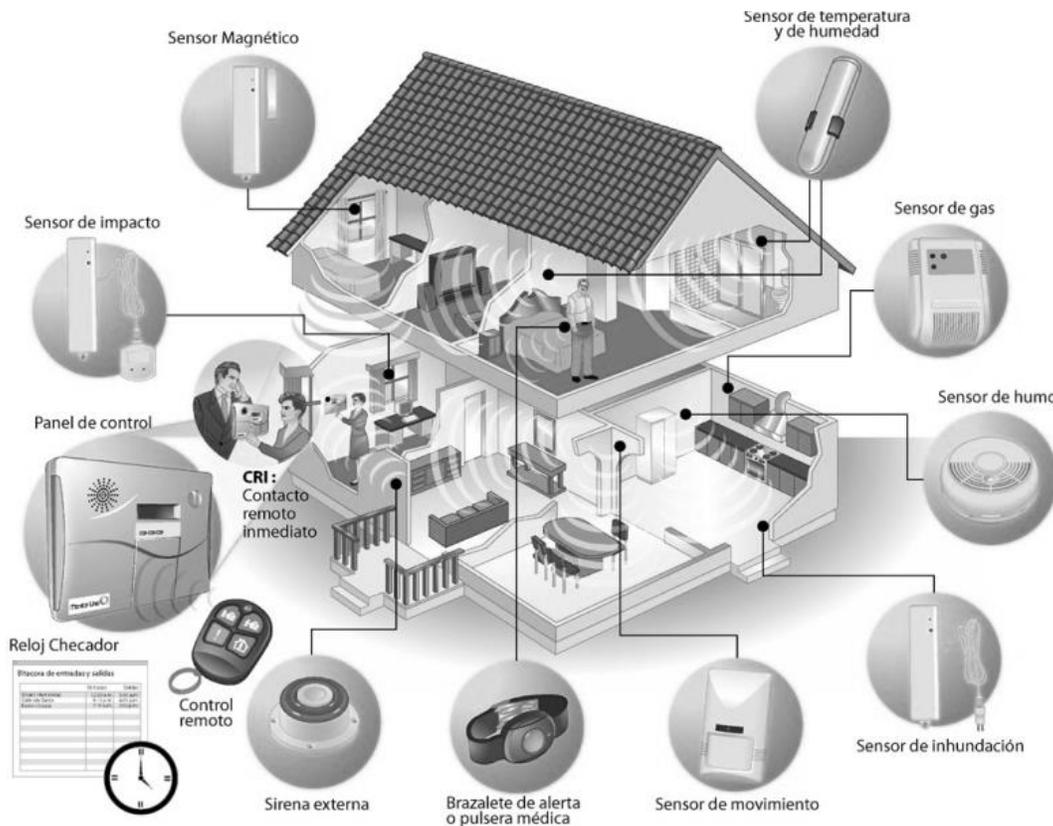


Figura 2 Sensores [Birtingenio, 2011]

Existen diversos tipos de sensores, que se pueden agrupar con algunos criterios de clasificación. A continuación se muestran algunos tipos de sensores existentes [Romero C., 2005]:

Tabla 3 Tipos de Sensores según Alimentación [Romero C., 2005]

Tipo	Alimentación
Activos	Debe ser alimentado eléctricamente a los niveles apropiados. Estos emiten energía sobre los objetos y reciben la señal reflejada por el mismo. Ejemplo: LIDAR (light Detection an Ranging) permite conocer la información de las Alturas y variaciones de altura en superficies calculando el tiempo de retorno de una señal.
Pasivos	No requieren alimentación eléctrica. Utilizan fuentes externas de

	energía para obtener información de los objetos. Ejemplo: La mayoría de los sensores utilizados para la observación de tierra.
--	--

Tabla 4 Tipos de Sensores según el Tipo de Señal Implicada [Romero C., 2005]

Tipo	Tipo de Señal Implicada
Continuos	Proporciona señales continuas. Estos solo pueden transmitir 2 estados, o mejor dicho, solo transmiten la ocurrencia de esos dos estados, ejemplo: cuando detectan la presencia de humo, o de agua.
Discretos	Proporciona señales discretas. Dispone una cantidad finita de entradas y salidas posibles, estos corresponden a los posibles estados de la variable, ejemplo: iluminación o no, presencia o no, circuito abierto o cerrado, entre otros. Son denominados detectores, ya que <i>detectan</i> la presencia o ausencia de algunas variables del entorno.

Tabla 5 Tipos de Sensores según el Ámbito de la Aplicación [Romero C., 2005]

Tipo	Ámbito de la Aplicación
Gestión climática	Sensores de temperatura, termostato, sondas de temperatura para inmersión, etc.
Gestión contra Incendios	Sensores iónicos, termovelocimétricos, sensores ópticos, infrarrojos, etc.
Gestión contra Intrusos	Sensores de presencia por infrarrojos, por microondas o ultrasonido, sensores de apertura de puertas o ventanas, sensores de rotura de cristales, etc.
Control de presencia	Lector de teclado, lector de tarjetas, identificadores corporales.
Control de Iluminación	Sensor de luminosidad
Otros sistemas	Sensores de lluvia, viento, gas, consumo eléctrico o de agua, etc.

2.1.2 Actuadores

Son dispositivos que actúan sobre el medio exterior y afectan físicamente la vivienda. Los actuadores logran convertir una magnitud eléctrica en otro tipo, como puede ser mecánica o térmica, realizando un proceso inverso al que realizan los sensores. Algunos ejemplos de actuadores pueden ser por ejemplo el motor de las persianas, los contactos de un circuito de iluminación, entre otros.

Los actuadores son conectados a las tarjetas de salida de un sistema inteligente. Si la actuación es completa o nula, los actuadores son mandados por señales digitales, en cambio si la actuación es variable, estos serán mandados por señales analógicas. Algunos dispositivos que pueden ser considerados actuadores serían [Romero C., 2005]:

- ❖ **Relé:** son interruptores que permiten conmutar circuitos de potencia más elevada mediante una señal de baja potencia. Su funcionamiento se basa en la actuación de un solenoide recorrido de corriente continua, al pasar la corriente por la bobina se magnetiza el núcleo de hierro y atrae la armadura, lo que provoca la apertura o cierre de contactos eléctricos.

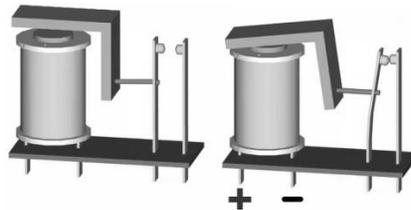


Figura 3 Actuadores: Relé [Radioelectronica, 2011]

- ❖ **Contactores:** son relés de potencia. Una bobina se excita con la tensión de alimentación y cierra unas platinas de cobre, lo que permite el paso de más o menos corriente.



Figura 4 Actuadores: Contactores [Moeller, 2011]

- ❖ **Reguladores o Dimmers:** son dispositivos basados en semiconductores lo que permite regular la potencia que llega a una carga. Un ejemplo en el uso de este en la domótica es el poder controlar la intensidad de las bombillas, luminarias, etc. Para su uso, es esencial prestar atención a las especificaciones de fábrica, pues hay que conocer la carga máxima, su resistencia, inductivas, etc.

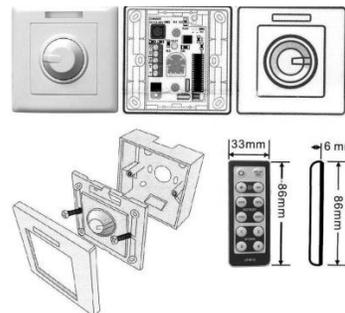


Figura 5 Actuadores: Reguladores o Dimmers [Shoptronica, 2011]

- ❖ **Electroválvulas:** son válvulas las cuales su apertura es controlada mediante una señal eléctrica externa. Su principal uso es para controlar caudales de líquidos o gases, en edificios o viviendas se utilizan para el control de agua y gas, así como también en sistemas de aire acondicionado.



Figura 6 Actuadores: Electroválvulas [Hogartintorero, 2011], [Directindustry, 2011]

- ❖ **Motores Eléctricos:** estos lo que hacen es convertir energía eléctrica en energía mecánica, para así poder generar movimiento. Algunos ejemplos de este aplicado a la domótica pueden ser los ventiladores, bombas, entre otros.



Figura 7 Actuator: Motores Eléctricos [Mahersolhogar, 2011]

- ❖ **Resistencia Eléctrica:** se utiliza para elevar la temperatura del medio donde estas se encuentran, para esto lo que hace es hacer pasar a través de un conductor una corriente eléctrica, que produce el calentamiento de este. Algunos ejemplos pueden ser radiadores, calefactores, secadores, etc.



Figura 8 Actuator: Resistencia Eléctrica [Electromoda, 2011][Solac, 2011]

2.1.3 Interface

Las interfaces hacen referencia a los dispositivos, ya sea pantallas, móvil, Internet, conectores u otros, y los formatos, como binario o audio, en que se muestra la información del sistema a los usuarios u otros sistemas y donde estos pueden interactuar.



Figura 9 Interfaz [[Dte.Upct, 2011]

2.1.4 Infraestructura

Su objetivo es llevar la información que producen los sensores hasta el sistema de control, y alimentarlos con una tensión eléctrica adecuada. Existen diferentes topologías de cableado, entre las que se pueden distinguir 3:

❖ **Bus:** todos los sensores y actuadores se encuentran directamente conectados [W. Stallings, 2004].

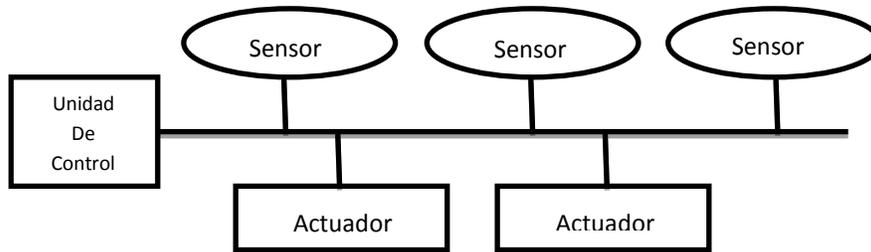


Figura 10 Topología: Bus [Elab. Propia]

En la topología bus se puede diferenciar entre varias posibilidades, como son [Romero C., 2005]:

Sistemas que se comunican entre sí mediante un cable, y éste además, proporciona la energía para funcionar, es decir el cable realiza el papel de transmisor y alimentador. Posee un ahorro en la instalación pero su velocidad de transmisión es limitada.

- ❖ Sistemas que necesitan una clave para datos y otra para la alimentación eléctrica.
- ❖ Sistemas hídricos que son una mezcla de los dos anteriores.

En la topología de bus se puede dar que haya una o varias líneas de buses unidas a un bus principal, este tipo de bus, son conocidos como árboles. Entre sus ventajas destaca la reducción de cable, y entre sus inconvenientes, la caída de todo el sistema en caso de un cortocircuito, o el colapso de información al existir una sobre carga de datos [Romero C., 2005].

Se utilizan cuando hay un número de puntos reducidos a controlar, y el nivel de seguridad no es muy importante, o la velocidad no necesita ser elevada.

- ❖ **Centralizadas:** Cada estación está directamente conectada a un nodo central común. En general existen dos alternativas para el funcionamiento del nodo central. Una de estas es el funcionamiento en modo de difusión, en el que la transmisión de una trama por parte de una estación se retransmite sobre todos los enlaces de salida del nodo central, en este caso pese a que físicamente es como una estrella, su lógica es del tipo bus. [W. Stallings, 2004]

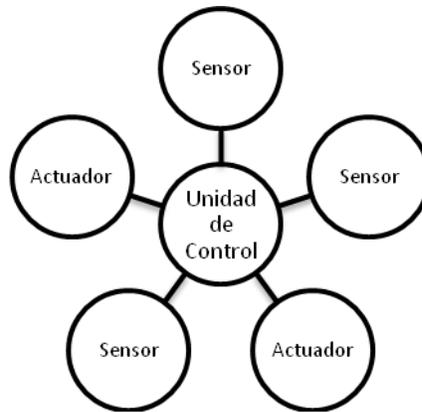


Figura 11 Topología: Centralizada [Elab. Propia]

- ❖ **Mixta:** combina las dos topologías anteriores.
 - *Mixta estrella/bus:* tienen varias estrellas unidas a un bus.
 - *Mixta estrella/estrella:* cada planta tiene una estrella y todas las estrellas están unidas al núcleo.

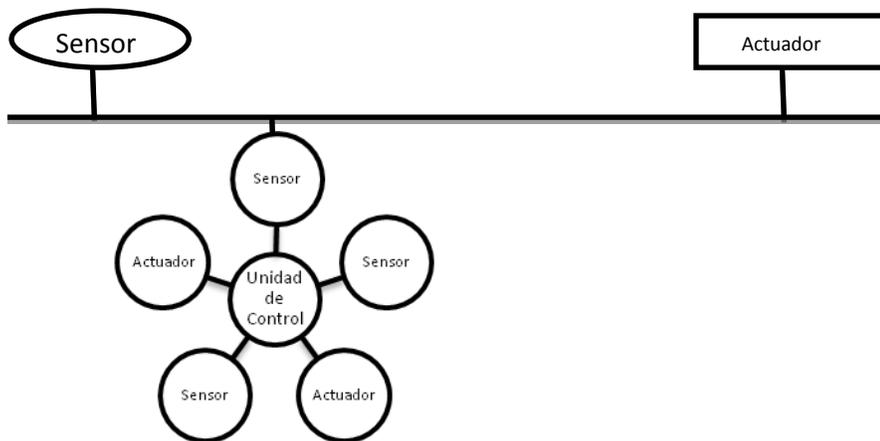


Figura 12 Topología: Mixta [Elab. Propia]

2.1.5 Unidad de Control

La unidad de control es el encargado de gestionar toda la instalación, recibiendo las señales entregada por los sensores, y emitiendo las señales que llegarán a los actuadores, además de posibilitar la conexión con la interfaz de usuario. En las unidades de control se pueden distinguir 2 sistemas [Romero C., 2005]:

- ❖ **Sistemas Centralizados:** La unidad de control está conectada a un único dispositivo, en el cual se ejecuta el programa previamente instalado.
- ❖ **Sistemas Distribuidos:** El control se alberga en cada uno de los componentes, es decir, se encuentra descentralizado. Las instalaciones son más flexibles e

independientes, aunque la programación se vuelve más complicada por el hecho de tener que hacerlo por cada uno de los componentes. Deben poseer un protocolo de comunicación.

En caso que la unidad central exista como tal, su principal característica son el número de entradas y salidas que permite conectar, y estas pueden ser: Entradas Digitales, Entradas Analógicas, Salidas Digitales y Salidas Analógicas.

Además de las entradas y salidas, la unidad de control puede estar compuesta por diversos componentes como son [Romero C., 2005]:

- ❖ **Hardware de Proceso de Datos:** es el cerebro de los datos el que decide cómo se actuará en función de los datos recibidos. Se pueden diferenciar varios tipos de procesadores:
 - *Centrales Micro-procesadoras:* Poseen una instalación sencilla, y poca flexibilidad en cuanto a crecimiento. Puede ser usado en luces, calefacción e incluso sistemas de anti-intrusión, pero no emite mensajes de despertador, ni digitaliza imágenes de vídeo, etc.
 - *Autómatas Programables:* actúa sobre el exterior en función de los datos recibidos. Son buenos en el almacenamiento masivo de datos, reconocimiento de voz, digitalización de imágenes, entre otros. Son idóneos para sistemas distribuidos, ya que pueden hacer su trabajo e informar y dar órdenes de sistemas superiores.
 - *Ordenadores:* dispone de microprocesadores más potentes y rápidos que los dos anteriores, son programables en lenguaje de propósito general, tiene capacidad de memoria y almacenamiento grande, pueden transmitir información a otros ordenadores a grandes velocidades, entre otras cosas.
 - *Controladores Embebidos:* sistemas mono-placa que disponen de un micro-controlador, junto con los sensores y actuadores necesarios. Son muy habituales en electrodomésticos y en general en sistemas cuya venta justifique tal elección de diseño, ya que implica una mayor inversión.
 - *Hardware de Entrada y Salida:* estas son las partes más distintivas de los sistemas de control es el nexo de unión entre el procesador inteligente y los sistemas externos, sensores y actuadores. Estos pueden ser incluidos en la misma unidad central o como módulos independientes que se conecten a este.
- ❖ **Hardware de Relación con el Usuario:** es la interacción que puede tener el usuario con los diferentes dispositivos instalados en la vivienda, y se pueden agrupar en:
 - *Programación Diaria:* interacción habitual.
 - *Parametrización:* modificar la programación a cada sistema para cambiar la actuación de este, según los deseos de cada usuario.

- *Simulación de Presencia*: el sistema acciona diversos dispositivos para así simular la presencia humana.
- *Control y Monitorización remota*: puede ser mediante vía telefónica o internet.

2.2 Servicios a Gestionar

La domótica se puede requerir por diversos motivos, por lo que esta trata de adaptarse a cada una de esas necesidades. Existen una gran cantidad de dispositivos que pueden ser automatizados, como puede ser la calefacción, iluminación y otros, que tratan de ser agrupados dependiendo la necesidad del usuario en confort, seguridad, ahorro energético, comunicaciones, entretenimiento y discapacitados. Aunque puede darse que alguna aplicación quede en más de un área, como lo es por ejemplo, la iluminación, que podría quedar en confort y ahorro energético [Romero C., 2005].

2.2.1 Gestión de Energía

La gestión de la energía es la que se encarga de controlar y optimizar el gasto energético de cada uno de los sistemas que utilizan energía en el edificio, y el uso que más se le da, es para reducir los gastos de los usuarios. Esta gestión divide al edificio en circuitos prioritarios (en ningún momento se corta la corriente) y circuitos no prioritarios (se controla en función de la energía consumida). La gestión de la energía se encarga de [Romero C., 2005]:

- ❖ *Uso racional de la energía*: es consumir solo la energía necesaria sin desperdicios.
- ❖ *Prioridad en la conexión de cargas*: establece tasas máximas de consumo simultáneo y prioridades en los actuadores, para así poder desconectarlos en orden si llegasen a pasar la tasa de consumo máxima.
- ❖ *Uso de tarifas especiales*: su base es aprovechar los horarios donde las tarifas de consumo son más económicas. Para lograr esto utiliza sistemas receptores de energía que posean un funcionamiento acumulador como por ejemplo, los sistemas de acumulación de agua, calefacción por acumulación, entre otros.
- ❖ *Utilización de sistemas acumulación*: se alimenta en horarios que tengan un menor costo económico, para luego entregar esa energía acumulada en forma de calor, para esto utiliza sistemas temporizadores para poder activarse.
- ❖ *Zonificación de la calefacción y aire acondicionado*: la vivienda o edificio se divide en zonas, esto permite poder regular y programar independientemente cada una.
- ❖ *Programa de la climatización*: el usuario puede programar según alguna necesidad o deseo, puede modificar en cualquier momento esta programación, y puede seleccionar entre un modo manual o programado.

Hay diversos dispositivos que son utilizados para gestionar la energía, entre ellos encontramos [Romero C., 2005]:

- ❖ **Racionalizador**: un aparato mide el consumo energético de una instalación, y cuando este llegue a los niveles prefijados conecta/desconecta unos u otros dispositivos. Su objetivo es que en ningún momento se supere la máxima potencia de la instalación.
- ❖ **Sistemas Acumuladores**: este permite utilizar las tarifas nocturnas (que tienden a ser más baratas que las diurnas).
- ❖ **Control selectivo de las cargas**: este se encarga de ir desconectando dispositivos, según su prioridad, en función al nivel de consumo sobrepasado con respecto al nivel máximo.

2.2.2 Gestión de Confort

La gestión de confort, es la encargada de facilitar un mayor nivel de comodidad al usuario en las actividades que desarrolle dentro de su vivienda. Su principal objetivo es la interacción del usuario con el medio que lo rodea, y no tiene una mayor relevancia el consumo energético o la seguridad. Para poder lograr su objetivo, éste debe poder controlar lo mejor posible las variantes físicas que afectan y/o pueden modificar su entorno. En la gestión del confort lo que más importa es el bienestar y rendimiento de trabajo de las personas, ejemplo la calidad de la luz, temperatura, entre otros.

Encontramos dos orientaciones distintas: primero la orientación a la vivienda (cuyo objetivos es el confort y la comodidad de sus habitantes por calidad de vida), y segundo a los edificios (cuyo objetivo es el confort de los trabajadores para lograr una mejor calidad en las condiciones laborales) [Romero C., 2005].

Entre alguna de las aplicaciones que le podemos dar a la gestión de control, destacan:

- ❖ **Regulación de la Iluminación**: controlar la intensidad de las luces, o la cantidad de luz en las habitaciones, el sistema para regular esto puede ser manual o automático. La administración del sistema de iluminación puede ser: *Autónomo* (cada habitación independiente) o *Centralizado* (controla la unidad central y utiliza programación horaria).

Existen 3 modos de control:

- *Modo bi-estable*: es bastante sencillo, ya que se basa en el todo o nada, es decir o las lámparas están todas prendidas o apagadas.
- *Modo analógico sin regulación*: este modifica el grado de luminosidad de una o varias lámparas, mediante el control electrónico de la tensión o de la corriente suministrada. La administración se puede realizar mediante impulsos, tiempo de pulsación o potenciómetro regulable.

○ *Modo analógico con regulación*: este es más complejo, permite modificar el nivel de iluminación teniendo en cuenta distintas variables como son, nivel de la iluminación exterior, nivel de iluminación interior, hora, día de la semana, etc. Este sistema funciona atendiendo los valores de los sensores que hacen que la intensidad de las lámparas asociadas al regulador sean mayor o menor.

- ❖ ***Regulación de la Temperatura***: esta depende de si se pretende aumentar o disminuir la temperatura, y para esto existirán dos sistemas: calefacción o aire acondicionado. Estos sistemas de climatización consumen mucha energía, por lo tanto hay que hacer un control energético.
- ❖ ***Control de Automatismos***: estos son sistemas que controlan algún tipo de automatismo, la idea es controlar cualquier tipo de dispositivo electrónico. Controlan normalmente acciones que conllevan cierto grado de incomodidad o que necesitan precisión. Algunos ejemplos pueden ser el accionamiento automático para persianas y toldos o el accionamiento automático para electrodomésticos.
- ❖ ***Elementos auxiliares aplicados al confort***: esto son elementos auxiliares, son utilizados en funciones complementarias al sistema de confort, algunos ejemplos pueden ser: mandos infrarrojos, mandos por radiofrecuencia, control a través de internet, entre otros.
- ❖ ***Otros Servicios***: unidades de control activadas por la voz, accionamiento automático de riego, aspiración, ascensores, y otros.

2.2.3 Gestión de la Seguridad

Esta es una de las áreas más importantes de la domótica, pues de ella depende la integridad física de las personas e inmuebles. Su objetivo principal es la protección frente a los distintos agentes o factores que ponen en peligro la seguridad. Los objetivos más importantes son [Romero C., 2005]:

- ❖ Detectar situaciones de peligro o riesgo.
- ❖ Avisar mediante sistemas sonoros o vía módem.
- ❖ Realizar acciones orientadas a las personas y a las instalaciones.

Sus tareas básicas son:

- ❖ ***Prevención***: se deben determinar potenciales fuentes de peligro.
- ❖ ***Reconocimiento***: consiste en validar la señal autenticando su procedencia.
- ❖ ***Reacción ante alarmas***: Puede ser manual, donde las personas toman decisiones, o automática, donde actúa el sistema cortando gas y la electricidad, abriendo puertas, etc.

2.2.3.1 Tipos de sistemas de seguridad

Podemos identificar los siguientes tipos de seguridad:

- ❖ **Sistemas de alarmas técnicas:** están son activadas cuando se produce una variación en un parámetro físico o químico en el medio. Son muy útiles para la detección de incendios, inundaciones, escapes de gas, entre otras cosas. Posee salidas acústicas, luminosas y telefónicas para poder avisar al usuario de esta alarma.
- ❖ **Sistemas antirrobo:** se encarga de evitar el acceso de gente extraña a la propiedad y disuadirla en sus intentos. Usa detectores de presencia, sensores de rotura de cristales, etc., además de simuladores de presencia que encienden luces y abren y cierran puertas.
- ❖ **Sistemas de control de acceso:** estos permiten controlar el paso de personas mediante detectores de metales, barreras infrarrojas, y otros. Incluso identifican a las personas que entran y salen mediante tarjetas magnéticas de identificación, activación por voz, o cualquier otra señal biométrica. Otra forma es la identificación manual, a través de un video portero con una pequeña cámara, un cable de video y una pantalla.
- ❖ **Sistemas de alarmas médicas:** estos controlan parámetros biológicos como son la presión arterial, la azúcar en la sangre, etc. Disponen de sensores en el cuerpo y emisores de señales de alarma local o remota por módem.

2.2.4 Gestión de las Comunicaciones

Esta se encarga de captar, transportar, almacenar, procesar y difundir datos o información, otra manera de decirlo, es que, es la gestión de la información de la casa a distancia: salud de los ocupantes, tele-formación, teletrabajo, etc. Sus objetivos principales serian [Romero C., 2005]:

- ❖ **Control remoto de equipos e instalación:** activar remotamente los equipos y las instalaciones domésticas.
- ❖ **Transmisión de alarmas:** enviar hacia el exterior cualquier alarma que le produzca daño al inmueble.

Los elementos de este sistema son:

- ❖ **Emisor de la señal:** son los que generan la información, la codifican y luego la ponen en el medio. Estos pueden ser emisores o receptores. La información puede ser de dos tipos: datos y protocolos de control, o señalización.
- ❖ **Receptor de señal:** estos se encargan de decodificar la información y presentársela al destinatario.

- ❖ **Medio físico de transmisión:** en estos circula toda la información. Puede ser par trenzado, cable coaxial, fibra óptica, y otros.

Las comunicaciones se pueden dar de dos formas, internas o externas. El primer tipo permiten la transmisión y el intercambio de información dentro de la instalación, por ejemplo, el circuito cerrado de televisión. Y el segundo tipo, permiten la transmisión o el intercambio de información de la instalación al exterior, como por ejemplo, control remoto del sistema a través de internet.

2.2.5 Gestión del Entretenimiento

Es un nuevo tipo de servicio el cual está estrechamente relacionado con los servicios de gestión de la comunicación. Algunos ejemplos pueden ser: video bajo demanda, videoconferencias y grabación de video, tv interactiva, publicidad interactiva y canales virtuales, emisiones deportivas, noticias, videos musicales y audio en tiempo real, guía de programación, juegos de consolas y juegos de tv interactiva y descargas de cd, mp3 y otros formatos [Romero C., 2005].

2.2.6 Gestión de Servicios para Discapacitados

Para las personas mayores o problemas de movilidad, la domótica les ofrece algunos servicios como: tele-gestión, conexión directa al exterior, contar con un apoyo exterior, automatizar todos los elementos de la vivienda, mando de control a distancia único y de fácil funcionamiento, luces guías, interruptores con símbolos y sensibles al tacto, o apertura automática de puertas [Romero C., 2005].

2.3 Estándares

Existen diversos estándares para la domótica, aunque aún siguen surgiendo nuevos. Los principales estándares existentes son [Romero C., 2005] [C. Jara, 2009].

2.3.1 BatiBus

BatiBus es un estándar europeo, desarrollado por Merlin Gerin, AIRELEC, EDF y LANDIS&GYR, los que luego formaron el BCI (BatiBus Club International). Sus principales características son [Romero C., 2005]:

- ❖ Bus simple de una sola línea que permite la intercomunicación entre todos los módulos en sistemas de control como: calor, aire acondicionado, luces y otros.
- ❖ Se utiliza como medio físico, un par trenzado, aunque también se puede utilizar cable telefónico o electrónico.
- ❖ Protocolo de comunicación basado en CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance), es decir, cada elemento está autorizado a comunicarse cuando lo desea, siempre y cuando la línea esté disponible.
- ❖ Permite cualquier topología de red (anillo, estrella, bus, árbol, u otra).

- ❖ El cable también le proporciona energía a los sensores.
- ❖ La dirección de los módulos se identifica al instalarlos.

2.3.2 CEBus

Consumer Electronic Bus es un estándar norteamericano desarrollado por la EIA (Electronic Industries Association), y se caracteriza por [Romero C., 2005]:

- ❖ Comunicación distribuida que permite un ancho banda suficiente para audio y video.
- ❖ Permite casi cualquier tipo de medio de comunicación como línea de alimentación, par trenzado, fibra ópticas, entre otros.
- ❖ Puede transmitir a 8000 bit/seg., utilizando mensajes en paquetes. El bus que lo componen está formado por 8 pares trenzados: 3 de audio, 4 de video y uno para el control CEBus.
- ❖ El mensaje es independiente del medio y lleva la dirección del destino.

2.3.3 EHS

European Home System, se define como un sistema de red completo, con todas las funciones domóticas, de forma modular, expansible y configurable. Es un sistema distribuido, donde, cada unidad conectada en la red, negocia automáticamente su dirección de red, para luego darla a conocer, y busca otras unidades que puedan estar interesadas en ella. De sus características se destacan [Romero C., 2005]:

- ❖ Tiene presente todos los niveles OSI.
- ❖ Posee 256 niveles de direccionamiento jerárquico en cada sección que se puede unir por routers.
- ❖ Las unidades se auto-configuran al conectarlas al bus.

2.3.4 EIB

European Installation Bus es un estándar europeo, orientado a la gestión técnica. Se trata de un sistema descentralizado y estandarizado. Entre sus ventajas encontramos [Romero C., 2005]:

- ❖ **Adaptable y modular:** si se produce una modificación en la utilización del edificio o una ampliación, no se precisa modificar el cableado, ya que todos los elementos están conectados a una misma línea de bus, solo habría que programar de nuevo a los componentes.
- ❖ **Reduce el mantenimiento:** todos los sistemas están conectados entre sí.
- ❖ **Ahorra tiempo:** por tener un cableado simple, se reducen los tiempos de montaje.

- ❖ *Es ampliable*: todos los componentes se pueden conectar al bus disponible, y como además es compatible con sistemas superiores, puede ser conectado con otros sistemas de control que se encuentren en la vivienda o edificio.

2.3.5 Konnex

Es una iniciativa de 3 asociaciones europeas, con el objetivo principal de crear un único estándar europeo para la automatización de las viviendas. Algunos objetivos son [Romero C., 2005]:

- ❖ Crear un único estándar para la domótica y la inmótica que cubra todas las necesidades y requisitos de las instalaciones profesionales y residenciales de ámbito europeo.
- ❖ Aumentar la presencia de buses domóticos en áreas como la climatización.
- ❖ Mejorar las presentaciones de los diversos medios físicos de comunicación incidiendo en la tecnología de radiofrecuencia.
- ❖ Introducir nuevos modos de funcionamiento que permitan aplicar una filosofía Plug&Play a muchos de los dispositivos típicos del edificio.
- ❖ Contar con empresas proveedoras de servicios, con el objetivo de potenciar las instalaciones de tele-gestión técnica de los edificios.

2.3.6 LonWorks

Local Operating Network, es similar a EIB, pero mucho más difundido en EEUU que en Europa. Esta puede utilizar una gran variedad de medios de transmisión, como pueden ser aire, par trenzado, fibra óptica, y otros. Los LonWorks son una tecnología muy robusta y fiable por lo que está especialmente indicada para la automatización industrial, ámbito del que procese [Romero C., 2005].

Su arquitectura es un sistema abierto a cualquier fabricante que quiera usar esta tecnología sin depender de sistemas propietarios, que permite reducir los costes y aumentar la flexibilidad de la aplicación de control distribuida. Cualquier dispositivo LonWorks, o nodo, está basado en un micro-controlador especial llamado Neuro Chip.

Se puede destacar del Neuro Chip [Domotica.tk, 2011]:

- ❖ Tiene un identificador único, el Neuro ID, que permite direccionar cualquier nodo de dentro de una red.
- ❖ Tiene un modelo de comunicaciones que es independiente del medio físico sobre el que funciona, es decir, los datos pueden transmitirse sobre cables de par trenzado, fibra óptica, entre otros.

2.3.7 X-10

Es un protocolo de comunicación, el cual funciona a través de la red eléctrica de la vivienda. Por este protocolo se pueden interconectar dispositivos compatibles con X-10, y lograr que transmitan información por los cables eléctricos, creando así una red de dispositivos. Los dispositivos X-10 son modulares, por lo que pueden extraerse o ingresarse a un sistema con una mínima configuración adicional.

Los protocolos X-10 llevan más de 10 años en el mercado, y gracias a esta madurez y a la tecnología empleada, estos tienen un precio muy competitivo. Ha llegado a ser el líder del mercado norteamericano residencial y de pequeñas empresas, dado que los proyectos son realizados principalmente por el usuario final o electricistas sin conocimientos de automatización.

2.4 Trabajos Relevantes

En esta sección se mostrarán algunos trabajos de domótica, estos no han usado a robots Lego, si no que otras tecnologías disponibles, sin embargo el fin de ambos trabajos es bastante similar.

2.4.1 Claudio Jara, “Arquitectura Domótica utilizando dispositivos X10 y comunicación mediante Web Services”

Este proyecto describe una arquitectura domótica capaz de dotar al hogar de algún grado de inteligencia, pudiendo ésta, responder satisfactoriamente ante eventos indeseados pero previstos. Esta arquitectura permite generar acciones oportunas frente a situaciones de riesgo, para esto utiliza sensores que detectan factores no deseados tanto internos como externos. Este sistema además, posee herramientas dedicadas al confort y ahorro energético de la vivienda u oficina, dotando al hogar de dispositivos capaces de controlar distintos componentes como luz, calefacción, persianas, entre otros [C. Jara, 2009].

El elemento primordial de este trabajo para lograr su objetivo de seguridad y confort, es dotar a los dispositivos domóticos con el protocolo de comunicación X10, ya que, su fácil manejo y modularidad hacen de esta una herramienta potente y sencilla. Además, algo importante a destacar de este protocolo, es que usa el cableado eléctrico para funcionar, es decir, no necesita cables extras. Para que se pueda dar la comunicación entre un set de viviendas con el fin de alertar sobre eventos indeseados y tomar acciones ante esto, se utilizó la tecnología Web Services [C. Jara, 2009].

Para poner a prueba esta arquitectura se usaron dos escenarios de prueba, estos fueron [C. Jara, 2009]:

- ❖ **Ejecución de acciones remotas:** se trató de encender una lámpara a través de la aplicación web optimizada para el iPhone y a través de un mensaje de texto.
 - Para hacerlo desde la página web, desde el iPhone se accede al sistema web, que por un previo logeo, lista todos los dispositivos que puede controlar de la estación

- Y para realizarlo a través de mensajes de texto, el usuario previamente ingresando a la estación mediante la interfaz para el iPhone crea un mensaje de texto con el ID de estación y el código de actuador.
- ❖ **Alarma por sensor de movimiento:** para esto se debe configurar la estación para que realice las acciones deseadas, en este escenario se hizo que encendiera una lámpara, mandara un correo electrónico y un mensaje de texto.

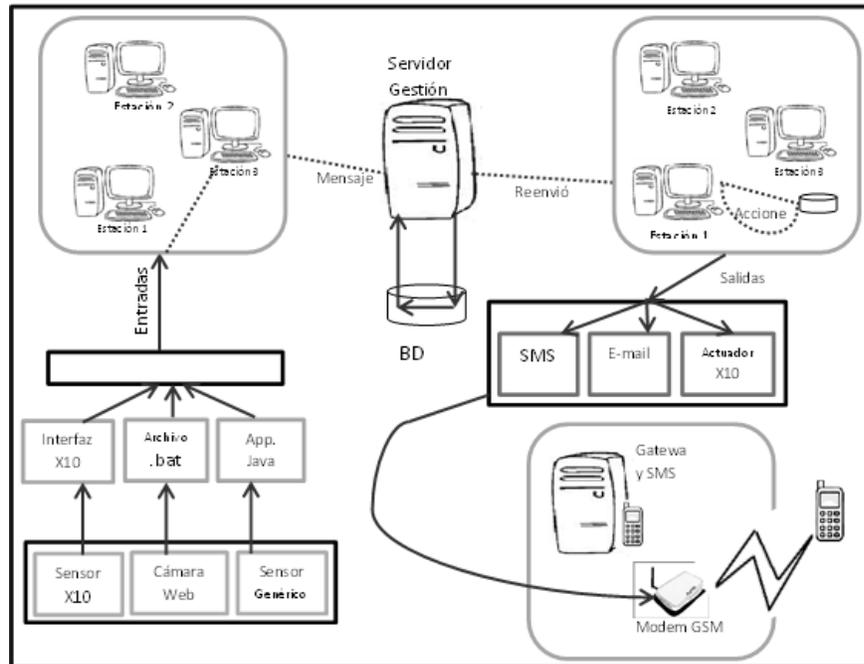


Figura 13 Modelo de la Arquitectura [C. Jara, 2009]

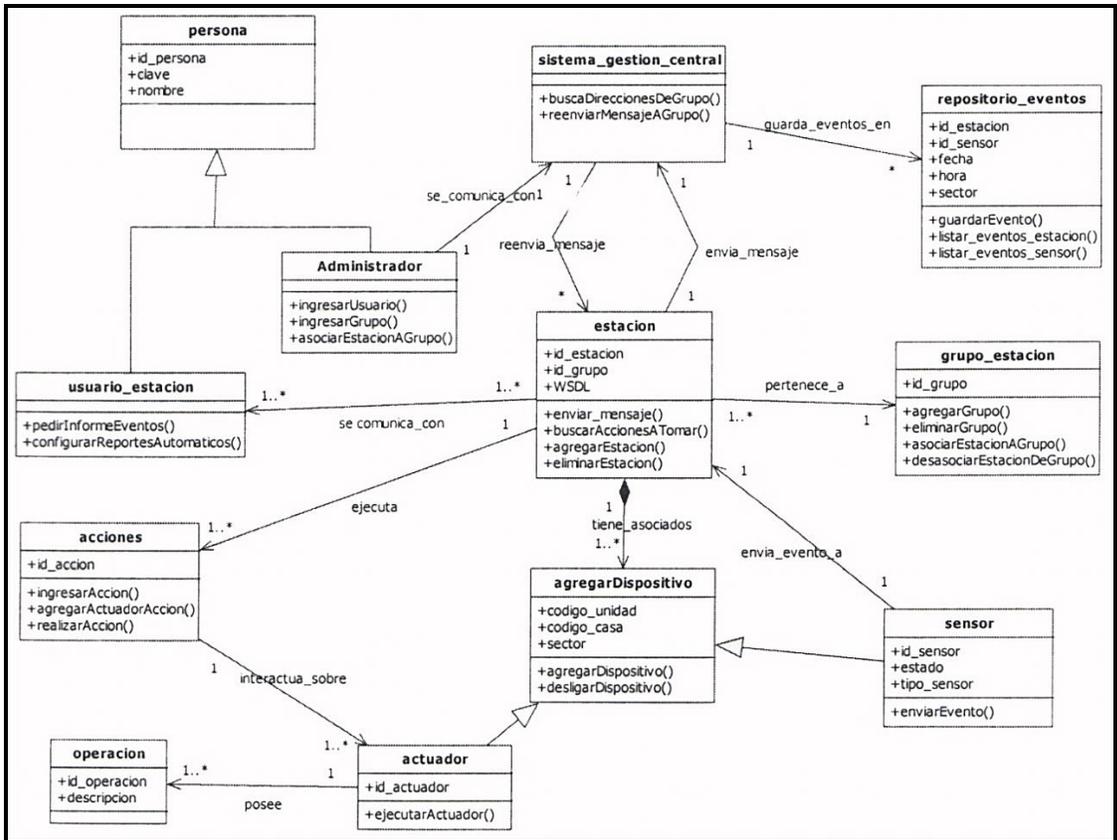


Figura 14 Diagrama de Clase [C. Jara, 2009]

2.4.2 Juan P. Cofré, Gabriel Moraga, “Usabilidad y Accesibilidad Aplicada a Herramientas Domóticas”

Aproximadamente 500 millones de personas a nivel mundial posee algún tipo de discapacidad que, de una u otra manera, restringen o impiden a la persona desenvolverse de forma plena en su entorno y en la sociedad. Del mismo modo, existe una cantidad no cuantificada de la población que posee necesidades especiales que afectan su desempeño en las actividades de la vida cotidiana. En cualquiera de los dos casos anteriormente mencionados, las actividades cotidianas se pueden volver complejas, una verdadera odisea, o en algo imposible de realizar. Puede creerse que el hecho de no poder tener control completo o al menos parcial del entorno es un problema mínimo. Un sujeto discapacitado o con necesidades especiales puede solicitar la ayuda de un tercero que les facilite o realice las diversas tareas por él, sin embargo, dicha aseveración no soluciona el hecho de que la persona sigue siendo dependiente y persiste el sentimiento de ausencia de control de sus acciones ni de su propia vida [J.P.Cofré, 2010].

A partir de lo antes expuesto, se plantea la necesidad de desarrollar un prototipo de herramienta que permita a las personas con necesidades especiales de tipo motor y/o cerebral leve, tener mayor accesibilidad y control de su entorno directo. El objetivo es lograr que las “simples tareas cotidianas” se vuelvan menos difíciles de realizar. Para llevar esto a cabo, el desarrollo del prototipo de herramienta empleará conceptos de áreas como la usabilidad, accesibilidad, domótica, y algunos de electrónica básica. Al mismo tiempo, el control del entorno se centralizará en un dispositivo móvil, en este caso, un *Smartphone* con interfaz táctil [J.P.Cofré, 2010].

Pese a que hoy en día existen dispositivos domóticos que permiten automatizar un entorno, ya sea una vivienda o un espacio de trabajo. Éstos no se encuentran enfocados principalmente a mejorar el grado de accesibilidad de la persona con el medio, más bien buscan alcanzar el concepto de casa inteligente. Resulta necesario destacar que al crear un prototipo de herramienta cuya finalidad es mejorar el nivel de accesibilidad, se debe potenciar el hecho de que esta sea lo más usable posible, teniendo claridad respecto a que los conceptos de usabilidad y accesibilidad deben ir de la mano en el desarrollo, pero que cada uno busca objetivos diferentes. El proyecto espera abarcar al mayor número de usuarios posibles, teniendo en consideración que cumplan con los requisitos mínimos necesarios para poder emplear el prototipo a desarrollar [J.P.Cofré, 2010].

2.4.2.1 Solución Propuesta

La solución que se propone es el desarrollo de un prototipo de herramienta, que sea capaz de ayudar en la rehabilitación integral del usuario, específicamente mejorando el grado de accesibilidad que tiene en su entorno directo, específicamente su hogar [J.P.Cofré, 2010].

Por lo portátil y comunes que ya se han vuelto, se ha escogido un teléfono móvil para incluir las funcionalidades del prototipo que se pretende construir. De esta manera, no es necesario incluir otro dispositivo al set habitual de uso diario. Ya que la tecnología ha tenido un avance significativamente grande, logrando abaratar los costos de este tipo de interfaz, se ha preferido trabajar con dispositivos táctiles (*touch screen*), ya que así se tendrá una gama más

amplia de configuración para lograr una mejor usabilidad del prototipo [J.P.Cofré, 2010].

2.4.2.1.1 Diagramado

A continuación se presenta el diagramado de la solución propuesta mediante diagramas de casos de uso y de secuencia [J.P.Cofré, 2010].

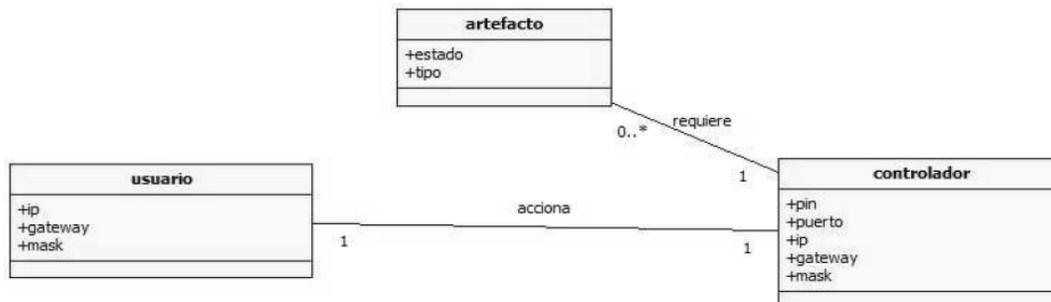


Figura 15 Diagrama de Clases [J.P.Cofré, 2010].



Figura 16 Prototipo Interfaz [J.P.Cofré, 2010]

3 Robots Lego Mindstorms NXT

Lego Mindstorms NXT, es un kit de robótica creado por la empresa Lego. Estos robots son capaces de interactuar con su ambiente mediante sensores y desplazarse en él, por medio de sus motores. Todas estas acciones que logran hacer, son controladas por un micro-procesador, programable que trae cada kit [LEGO, 2006].

3.1 Características del Bloque Lego Mindstorms NXT

El bloque NXT es una versión mejorada del Lego Mindstorms RCX. El bloque RCX es considerado el propulsor de los bloques programables de lego, y fueron creados en 1998. En el 2006 es lanzado el Lego Mindstorms NXT, ésta la vendió en dos versiones: la *Retail* y la *Education Base*, además de un kit para desarrolladores, según las características de los programas que quisieran desarrollar, y se distinguieron tres tipo:

- ❖ *Software Developer Kit (SDK)*: incluía los controladores de los puertos USB, archivos ejecutables y referencias a los byte codes.
- ❖ *Hardware Developer Kit (HDK)*: incluye la documentación y esquema para los sensores de NXT.
- ❖ *Bluetooth Developer Kit (BDK)*: documentos con los protocolos usados para la comunicación Bluetooth.

3.2 Micro-controlador

El micro-controlador NXT posee mayores capacidades de ejecución de programas que el RCX. El micro-controlador de NXT posee un ARM7 de 32 bits, con 256 Kbyte de memoria Flash y 64 Kbyte de memoria RAM. A diferencia de RCX que poseía un Hitachi H8/3000, 16Kbyte de memoria ROM, solo tenía 512 byte de memoria RAM [LEGO, 2006] [LegoRCX, 1998]. Los bloques NXT traen cuatro entradas para conectar los sensores y tres salidas para conectar los motores. El bloque se comunica con el computador por el cable USB y la interfaz gráfica, la cual ya va en su versión 2.0. Además de este modo de comunicación, también puede hacerlo mediante Bluetooth, este no solo permite una comunicación NXT-computador, sino que también permita NXT-NXT, celular-NXT, y otros.

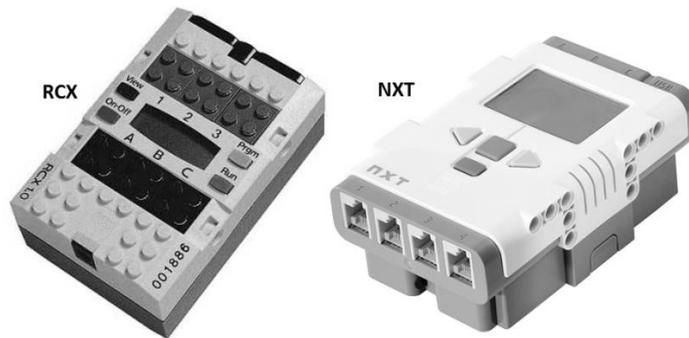


Figura 17 Brick RCX y NXT [LegoImag, 2001]

3.3 Sensores

Los sensores son los que le permiten al robot interactuar con su entorno. Existen diversos tipos, por ejemplo, los básicos que son, el sensor de luz, ultrasonido, tacto y sonido, además hay otros que pueden adquirirse de forma independiente como pueden ser el sensor de color, acelerómetro, giroscopio, compás, y otros.

A continuación se describirán los sensores básicos que componen el kit [LEGO, 2006]:

- ❖ **Sensor de Luz:** este es uno de los dos sensores que proveen de vista al robot. Este sensor distingue entre la luz y la oscuridad, y puede medir la intensidad lumínica de un objeto.



Figura 18 Sensor de Luz [LegoImag, 2001]

- ❖ **Sensor de Ultrasonido:** este es otro sensor que le proporciona el sentido de la vista al robot. Este le permite distinguir al robots si hay algún objeto en su camino, este utiliza para funcionar el principio científico de los murciélagos, es decir, calcula en tiempo en que una onda sonora tarda en rebotar, con esta información puede calcular a que distancia hay un objeto. Es capaz de medir distancias en centímetros o pulgadas, con un rango de 0 a 255 centímetros.



Figura 19 Sensor de Ultrasonido [LegoImag, 2001]

- ❖ **Sensor de Tacto:** este le proporciona al robot el sentido del tacto. Este sensor puede distinguir cuando está siendo presionado por algo, y cuando se ha liberado.



Figura 20 Sensor de Tacto [LegoImag, 2001]

- ❖ **Sensor de Sonido:** este le proporciona al robot el sentido de la audición. Le permite detectar los decibeles y decibeles ajustados.



Figura 21 Sensor de Sonido [LegoImag, 2001]

3.4 Programación

Para programar un robot del set Lego Mindstorms, este trae un software “Mindstorms NXT Labview”, el cual corresponde a un lenguaje gráfico, orientado a niños desde los 8 años en adelante. Para poder interactuar a un nivel más profundo con el kit, existen diversas alternativas, aunque con un nivel de programación más complejo. Algunas de estas alternativas son ICommand, Lejos NXJ, RobotC, entre otros.

3.4.1 ICommand

ICommand es un paquete de Java que sirve para controlar el bloque NXT a través de conexión bluetooth. Utiliza el estándar de firmware de Lego NXT para recibir los comandos en java desde un computador. ICommand fue lanzado en el 2006, aún existen razones para seguir utilizándolo, pero como fue diseñado para las personas que deseaban empezar a programar mientras estaba listo Lejos NXJ, ya no los seguirán actualizando y varias funcionalidades ya están obsoletas. [M. Hu, 1962]

3.4.2 Lejos NXJ

Lejos es una pequeña máquina virtual de Java. Es un conjunto de utilidades y clases Java que conforman un entorno de programación Java completo. LeJOS NXJ consiste en: [LEGO, 2006]

- ❖ Firmware de reemplazo para el NXT que incluye una máquina virtual de Java.
- ❖ Una librería de clases Java que implementa la API de leJOS NXJ
- ❖ Un conector para las clases de usuario Java para formar un archivo binario que pueda ser cargado y ejecutado en el NXT.
- ❖ Herramientas para el PC para la actualización del firmware, cargar programas, depuración, y muchas otras funciones.

- ❖ Un API PC para programas de computadores por escrito que se comunican con los programas de leJOS NXJ a través de secuencias de Java por medio de Bluetooth o USB, o usando el Protocolo de Comunicaciones de LEGO (LCP).

LeJOS NXJ incluye todas las clases de la API NXJ, así como las herramientas utilizadas para cargar el código con el brick NXT leJOS NXJ ofrece lo siguiente [LEGO, 2006]:

- ❖ Lenguaje orientado a objetos (Java).
- ❖ Temas preventivos (tareas).
- ❖ Matrices, incluyendo multidimensional.
- ❖ Recursión.
- ❖ Sincronización.
- ❖ Excepciones.
- ❖ Java incluyendo los tipos float, long, y string.
- ❖ La mayoría de las clases java.lang, java.io y java.util.
- ❖ Una bien documentada “*Robotics APP*”.

Hay muchas ventajas de utilizar leJOS NXJ más que el NXT-G u otros entornos de programación para el NXT. Estos incluyen [LEGO, 2006]:

- ❖ Utiliza el lenguaje estándar de la industria Java.
- ❖ Es compatible con programación orientada a objetos.
- ❖ Se trata de un proyecto de código abierto con muchos colaboradores.
- ❖ Tiene plugins tanto para Eclipse y Netbeans.
- ❖ Tiene soporte multiplataforma - Windows, Linux y Mac OS.
- ❖ Es mucho más rápido que el NXT-G.
- ❖ Tiene soporte completo para Bluetooth, USB, I2C y protocolos RS-485.
- ❖ Es compatible con otros algoritmos de robótica probabilística, como los filtros de Kalman.
- ❖ Ofrece clases de comportamiento que admite la arquitectura de subsunción para facilitar la programación de comportamientos del robot complejo.
- ❖ Ofrece decenas de programas de ejemplo.
- ❖ El sitio web cuenta con foros en línea para ayudar a resolver cualquier problema que pueda tener, para compartir ideas de proyectos, y para comunicarse con el equipo de desarrollo.

- ❖ Se admite la comunicación Bluetooth con otros dispositivos, tales como los receptores GPS.
- ❖ El uso de menú es fácil de utilizar.
- ❖ Es ampliamente utilizado por las universidades y otros centros educativos.
- ❖ Tiene soporte para simples aplicaciones de visión artificial

3.4.3 RobotC

RobotC es un lenguaje de programación basado en entorno Windows para escribir y depurar programas. Es una solución multiplataforma que permite a los estudiantes aprender el tipo de c- basado en la programación, usada en educación avanzada y aplicaciones profesionales. Algunos de los beneficios que ofrece son [RobotC, 2005]:

- ❖ Utiliza el estándar de la industria-C lenguaje de programación.
- ❖ Moderna interfaz gráfica de Windows con una interfaz visual estándar.
- ❖ Herramientas de depuración adicionales que el usuario pueda ver los estados en tiempo real de todos los motores y sensores. Estos son vistos como el programa se está ejecutando, no como una aplicación independiente.
- ❖ Una sola solución que sirva tanto para VEX Cortex y PIC VEX.
- ❖ Más de 100 programas de ejemplo con una extensa documentación para que los estudiantes y los aficionados pueden empezar a aprender a programar

4 Agentes Inteligentes y Sistemas Multiagente

4.1 Agentes

Definir un agente no es sencillo, entre los autores del área aun no llegan a un consenso para darle una definición única. Se puede partir diciendo que el termino viene del latín “agere” que significa hace y que agente deriva del participio “agens”.

Algunas de las definiciones que se pueden encontrar de agentes son:

- ❖ 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, Wooldridge [M. Wooldrige, 1999].
- ❖ Toda entidad que de forma autónoma perciba su entorno (real o simulado) mediante sensores y que actúe en el mismo mediante efectores, Russel & Norwig [S. Russel, 2003].

Como se mencionó anteriormente en ambas definiciones se dijo que un agente puede realizar acciones de forma autónoma, es decir, debe ser capaz de actuar sin la interacción humana, o de otros agentes y debe tener el control de sus acciones y estados internos. Esta capacidad de ser autónomo es lo que le da a un agente el calificativo de inteligente. Wooldridge, dice que: “Un agente inteligente, es capaz de realizar acciones autónomas y flexibles en la búsqueda de sus objetivos” [M. Wooldrige, 2002]. Esta flexibilidad de los agente inteligentes está dado por:

- ❖ **Reactividad:** debe poder percibir los cambios en su ambiente, y reaccionar oportunamente a ellos.
- ❖ **Pro-Actividad:** no sólo actuar en respuesta a su ambiente, si no también exhibir su comportamiento orientado a metas.
- ❖ **Sociabilidad:** debe poder interactuar con otros agentes y seres humanos, como un medio para lograr sus metas.

Otras características relevantes de los agentes, son: [Ana Mas, 2005]:

- ❖ **Razonamiento y aprendizaje:** estos logran que el agente se comporte “inteligente”, que logre aprender de las experiencias previas.
- ❖ **Movilidad:** son capaces de desplazarse entre los nodos de una red y ejecutarse en distintas plataformas.
- ❖ **Veracidad:** no debería comunicar información falsa a propósito.
- ❖ **Aprendizaje** cambiar su comportamiento, en base a sus experiencias previas.
- ❖ **Benevolencia:** prestar ayuda a otros agentes, y no entrar en conflicto con los objetivos propios.

- ❖ **Inteligencia:** hace referencia al método utilizado para desarrollar la lógica de los agentes, y está estrechamente relacionado con el lenguaje entre agentes.
- ❖ **Racionalidad:** actuar de forma racional con miras a cumplir sus objetivos.

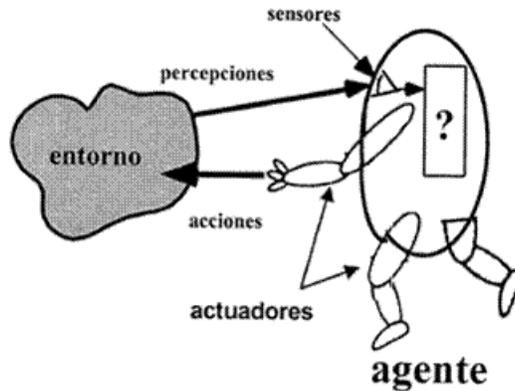


Figura 22 Agente [Erbol, 2011]

4.2 Arquitectura de Agentes

La arquitectura de los agentes está determinada por la estructura interna del mismo, como también por la organización de un sistema multiagente. La arquitectura determina los mecanismos que utiliza el agente para poder reaccionar a estímulos, actuar, comunicarse, etc. [Ana Mas, 2005].

Podemos clasificar las arquitecturas en 3 tipos atendiendo al tipo de procesamiento empleado [Ana Mas, 2005]:

- ❖ **Deliberativas:** utilizan modelos de representación simbólica del conocimiento, se basa en la hipótesis de los sistemas de símbolos-físicos enunciada por Newell y Simons, según esta hipótesis, un sistema de símbolos físicos que es capaz de manipular estructuras simbólicas, puede exhibir una conducta inteligente.

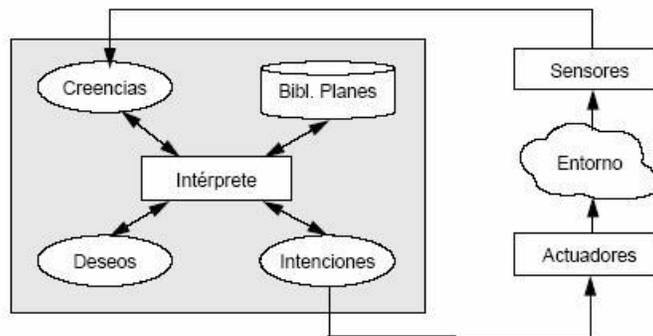


Figura 23 Arquitectura Deliberativa [Iglesias, 1998]

- ❖ **Reactivas:** estas cuestionan la viabilidad del paradigma simbólico, y proponen una que actúa siguiendo un enfoque conductista, con un modelo de estímulo y respuesta. Mantiene una serie de patrones que se activan bajo ciertas condiciones de los sensores, y poseen un efecto directo en los actuadores.

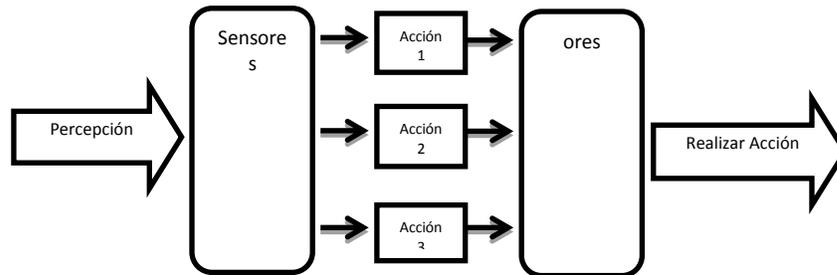


Figura 24 Arquitectura Reactivas [Iglesias, 1998]

- ❖ **Híbridos:** estas pretenden combinar aspectos de las dos arquitecturas ya vistas. Esta combina módulos reactivos con los deliberativos. Los primeros se encargan de procesar estímulos que no necesitan deliberación, mientras que los segundos determinan las acciones que deben realizarse para satisfacer los objetivos locales y cooperativos de los agentes.

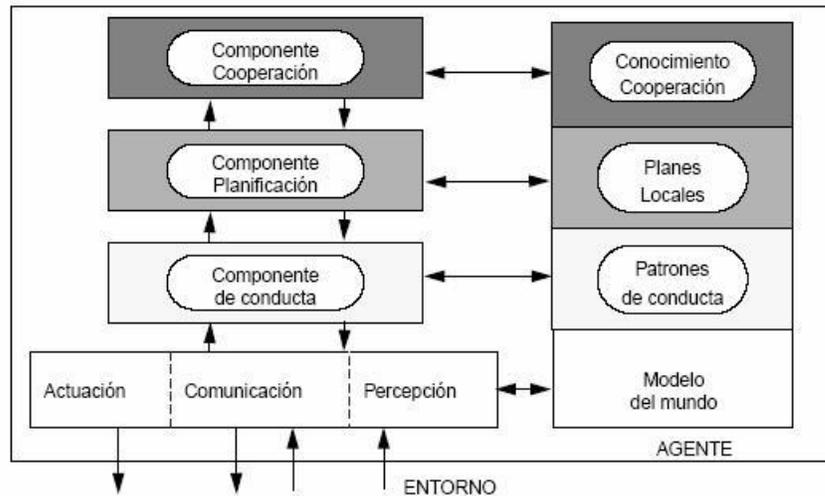


Figura 25 Arquitectura Híbridos [Iglesias, 1998]

4.3 Infraestructura Agente

Para que los agentes puedan comunicarse entre sí y entenderse, necesitan unas regulaciones, estas son conocidas como infraestructuras de agentes [Ana Mas, 2005], y son de 3 tipos:

- ❖ **Ontologías:** estas entregan un significado común de los conceptos, así los agentes se entienden. “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” [Neches, 1991]. Estas proveen una forma de describir los conceptos de una manera no ambigua, alguno de los lenguajes utilizados para describirlo pueden ser: XOL (Ontology eXchange Language, US bioinformatics community) u OML (Ontology Markup Language, University of Washington).

- ❖ **Lenguajes de Comunicación (ACL’s):** estos permiten especificar las formas de comunicación, es decir protocolos, entre los agentes, logrando que estos puedan intercambiar mensajes que sean entendibles. Para que el lenguaje sea entendido por todos, se deben tener claro 3 aspectos: Sintaxis (definición de la estructura de símbolos de la comunicación), Semántica (Denotación que entregan los símbolos) y la pragmática (como se interpretan los símbolos).
- ❖ **Protocolos de interacción (AIP’s):** estos logran los intercambios estructurados de mensajes. Los protocolos de interacción se definen como protocolos de comunicación con una secuencia permitida de mensajes entre agentes, junto a la restricción de contenido de este.

4.4 Comunicación entre agentes

Una característica básica de la noción de un agente es que este sociabiliza. Esto implica que puede establecer una comunicación con otros agentes, y por ende provocar cambio en sus objetivos y/o creencias.

Un acto comunicativo, se da cuando un agente lleva a cabo una serie de acciones que tienen como objetivo modificar la estructura cognitiva de otro agente. La mayoría de los actos comunicativos se realizan mediante el lenguaje hablado, en la lingüística, se utiliza el término “Actos del habla” para referirse a todo tipo de actos comunicativos [Searle, 1969]. El agente que toma la iniciativa de comenzar la comunicación se llama “hablante” y al que va dirigida esta acción se llama “oyente” [Rizo, 2007]. Los actos de habla deberían provocar ciertos efectos sobre el conocimiento del oyente, se denomina “efecto perlocutivo”, al efecto que el hablante intenta provocar en el oyente, y “efecto elocutivo”, al efecto que realmente produce.

Existen diversas categorías de los actos de habla, y serian:

- ❖ **Representativos:** exponen una determinada proposición como representación de estado de cosas del mundo.
- ❖ **Directivos:** intenta que el oyente actúe de tal modo que su conducta concuerde con el contenido proposicional.
- ❖ **Expresivos:** expresar agradecimiento, felicitaciones, bienvenida o pedir disculpas.
- ❖ **Declarativos:** actos que producen un cambio en el estado del mundo.

Se distinguen varias formas posibles para la comunicación [IGLESIAS, 1998], algunas de ellas son:

- ❖ **Arquitecturas de pizarra:** Los agentes se comunican mediante un esquema que permite publicar mensajes al resto de los agentes. Consta de:
 - *La pizarra:* Es una base de datos global donde acceden los agentes para publicar los mensajes. La información se suele estructurar en niveles de hipótesis jerárquicos.
 - *Fuentes de conocimiento:* Módulos especializados que intercambian mensajes.
 - *Mecanismos de control:* Para gestionar la concurrencia.
- ❖ **Paso de mensajes:** Los agentes dirigen mensajes a otros agentes, identificados por una dirección, nombre o *alias*. Este mecanismo se ha importado de la programación orientada a objetos concurrente.
- ❖ **Comunicación de alto nivel:** Consiste en posibilitar la comunicación en el nivel simbólico que requieren los agentes BDI para comunicar sus deseos, intenciones y objetivos. Para ello se definen actos comunicativos y conversaciones que incluyen una descripción semántica convenida de antemano.

4.4.1 Niveles y Protocolos de Comunicación.

Los protocolos de comunicación están especificados en niveles. El nivel inferior de un protocolo especifica el método de interconexión, el nivel medio especifica el formato o la sintaxis de la información que es transferida, y el nivel superior especifica el significado o semántica de la información, esta última, la semántica, no solo se refiere a la esencia del mensaje sino también al tipo de mensaje. [Rizo, 2007]

Existen 2 tipos de protocolos de comunicación, el binario, que involucra un emisor y un receptor, y el n-ario también llamado broadcasting o multicasting (un único emisor con múltiples receptores). Un protocolo está especificado por una estructura de datos con los siguientes campos: emisor, receptor(es), lenguaje del protocolo, funciones de codificación y decodificación, y acciones realizadas por el(los) receptor(es).

4.5 Sistemas Multiagente

En el punto anterior se dieron características de los agentes. Bajo esta idea se puede describir que es un sistema multiagente (SMA). Un SMA, según Weiss, es una red de agentes los cuales interactúan entre ellos, para poder resolver problemas que van más allá de sus capacidades individuales o de sus conocimientos [Y. Shoham, 1999].

Las características de un SMA son:

- ❖ Cada agente tiene información o capacidades incompletas, es decir, tienen un punto de vista limitado, y por ende no pueden resolver un problema solo.
- ❖ No existe un control global del sistema, es descentralizado.

- ❖ Los datos están dispersos entre los agentes.
- ❖ Poseen comunicación asíncrona.

Los SMA proporcionan una gran ventaja, y es que permiten la integración de agentes existentes en un gran sistema. Alguno de los beneficios esperables por el uso de esta tecnología son [K. Sycara, 1999]:

- ❖ **Mayor velocidad y eficiencia:** esto se da ya que los procesos son asíncronos y corren en paralelo.
- ❖ **Robustez y fiabilidad:** esto se da gracias a que el sistema puede eventualmente enfrentar situaciones en las que uno o más agentes fallecen.
- ❖ **Escalabilidad y flexibilidad:** dado que es sencillo integrar nuevos agentes al sistema.
- ❖ **Costos:** un agente es una unidad de bajo costo, comparado con el sistema como un todo.
- ❖ **Utilidad y factibilidad de desarrollar:** diseños modulares son más fáciles de desarrollar, mantener y reusar que los monolíticos.

La tecnología multiagente es una gran forma de conseguir esas metas de eficiencia y mayores retornos, en un ambiente atemporal, sin restricciones de distancia y expuesto a cambios continuamente [Rizo, 2007]. Sin embargo, esta tecnología también con lleva algunos desafíos, algunos de estos son:

- ❖ Descomposición del problema, asignación de tareas y la síntesis de los resultados parciales.
- ❖ El manejo de las percepciones en distintos entornos, para permitirle a los agentes mantener modelos coherentes en el ambiente.
- ❖ Implementación de un control descentralizado, y la construcción de mecanismos de coordinación eficientes.
- ❖ Diseño de algoritmos de planeación y aprendizaje eficientes.
- ❖ Representación del conocimiento y cómo un agente podrá razonar a partir de acciones, planes y/o conocimientos de otros agentes.
- ❖ Comunicación entre agentes. Los protocolos y lenguaje a utilizar, además del modo en que se comunicaran, que, cuando y con quien.
- ❖ Como enfrentarán las situaciones conflictivas negociando y resolviendo
- ❖ Forma de organización de los agentes para formar estructuras organizacionales, y como se asignarán los roles.
- ❖ Como se asegura que el sistema sea comporte de manera coherente y estable.

4.6 Metodologías de desarrollo de Sistemas Multiagente

Para el desarrollo de un MAS se hace necesario el uso de metodologías que propongan una estructura para poder desarrollar estos sistemas. El objetivo de estas metodologías es convertir el proceso 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. Existen varias metodologías, algunas de ellas se presentarán a continuación.

4.6.1 BDI

BDI (Belief-Desire-Intention), es un tipo de agente racional, el cual posee un conjunto de actitudes mentales: Creencias (Belief), Deseos (Desire) e Intenciones (Intention). Este modelo se ha desarrollado para proporcionar soluciones en entornos dinámicos o inciertos, en la que el o los agentes sólo tienen una visión parcial del problema, y es muy posible que tenga un número limitado de recursos. El conjunto de actitudes mentales y los planes son una parte fundamental para el estudio del sistema [Ana Mas, 2005].

Las características de los componentes de esta metodología, son los siguientes [Ana Mas, 2005]:

- ❖ **Creencias:** Representan el estado de información del agente, dicho de otro modo, son los conocimientos que este tiene de su entorno. Las maneras en que se pueden representar el estado del entorno son variables, como por ejemplo el valor de una variable. Estas creencias pueden contener reglas de inferencia, esto permite un encadenamiento hacia adelante para así poder inferir nuevo conocimiento. Lo normal es que esta información sea guardada en una base de datos.
- ❖ **Deseos:** Estos representan el estado de motivación que tiene el agente, es decir, los objetivos o situaciones que el agente quiere lograr o causar.
- ❖ **Intenciones:** representa el estado deliberativo del agente, es decir, lo que el agente ha elegido hacer.
- ❖ **Planes:** son las secuencias de acciones que un agente puede realizar para alcanzar una o varias de sus intenciones. Un plan puede incluir otro plan, ejemplo el plan llamar por celular, puede incluir al plan recargar el celular.

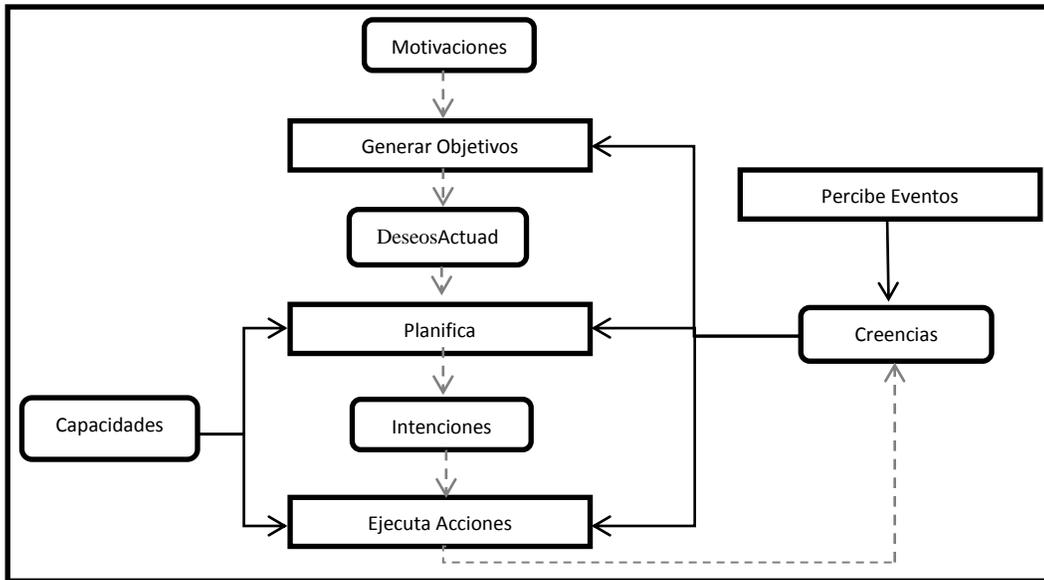


Figura 26 Esquema del ciclo mental interno de un agente BDI [Dccia, 2011]

4.6.2 Gaia

Esta metodología [M. Wooldrige, 2000] considera un sistema que está basado en agentes como una sociedad u organizaciones. Esta organización consta de una colección de roles, los cuales tienen ciertas relaciones entre ellos, y toman parte de los patrones de interacción con los otros roles. Cada uno de estos roles está definido por cuatro atributos, que son [Ana Mas, 2005]:

- ❖ **Responsabilidades:** determinan la funcionalidad, y esta descrita como propiedades de disponibilidad y seguridad.
- ❖ **Permisos:** definen los derechos de los roles para utilizar los recursos.
- ❖ **Actividades:** son aquellas computaciones que pueden realizarse sin interactuar con otros roles.
- ❖ **Protocolos:** modos de interactuar con otros roles.

El análisis de Gaia tiene 3 pasos y son [Ana Mas, 2005]:

1. Identificar los roles del sistema, estos pueden ser: individuos, departamentos u organizaciones.
2. Para cada rol, identificar y documentar los protocolos asociados, esto permite definir un modelo de interacciones.
3. Elaborar un modelo de roles con la información anterior, como resultado de esto se obtendrá un modelo de roles elaborado con sus permisos y responsabilidades, protocolos y actividades en las que participan.

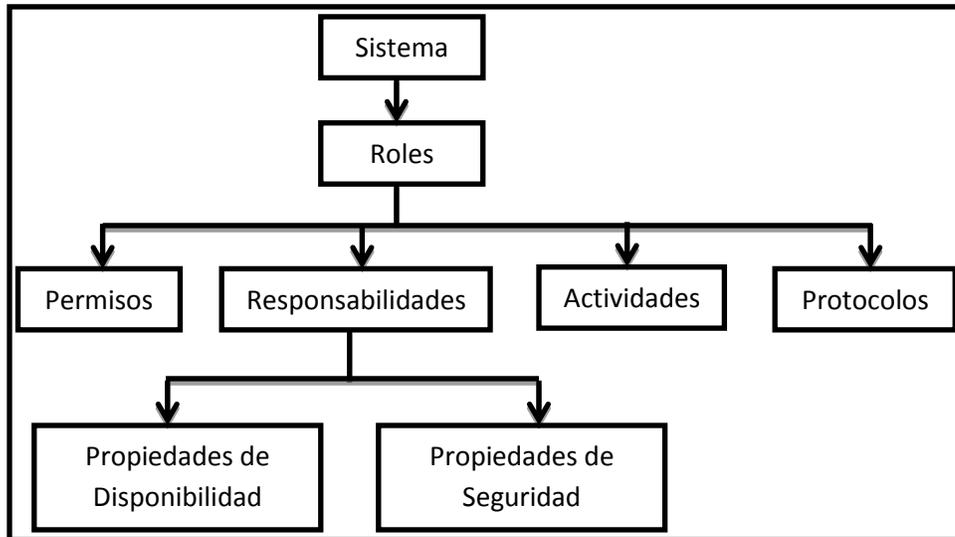


Figura 27 Organización de la Metodología Gaia [Upv, 2011]

4.6.3 MASE

Mase (Multi-agent System Software Engineering) trata todo el ciclo de desarrollo, desde la descripción del problema hasta la implementación de la solución. Adopta el paradigma orientado a objeto y considera a los agentes como una especialización de los objetos.

Mase tiene 2 etapas: análisis y diseño. La etapa de análisis consta de 3 pasos: primero la captura de los objetivos del sistema, luego viene el análisis de los casos de uso, y por último con los diagramas obtenidos es posible derivar los roles (estos son la base para la definición de las clases de agentes) y las tareas asociadas. Y la etapa de diseño consta de 4 fases: primero la creación de los agentes, luego la construcción de las conversaciones, después el ensamblado de las clases de los agentes, y por último el diseño del sistema.

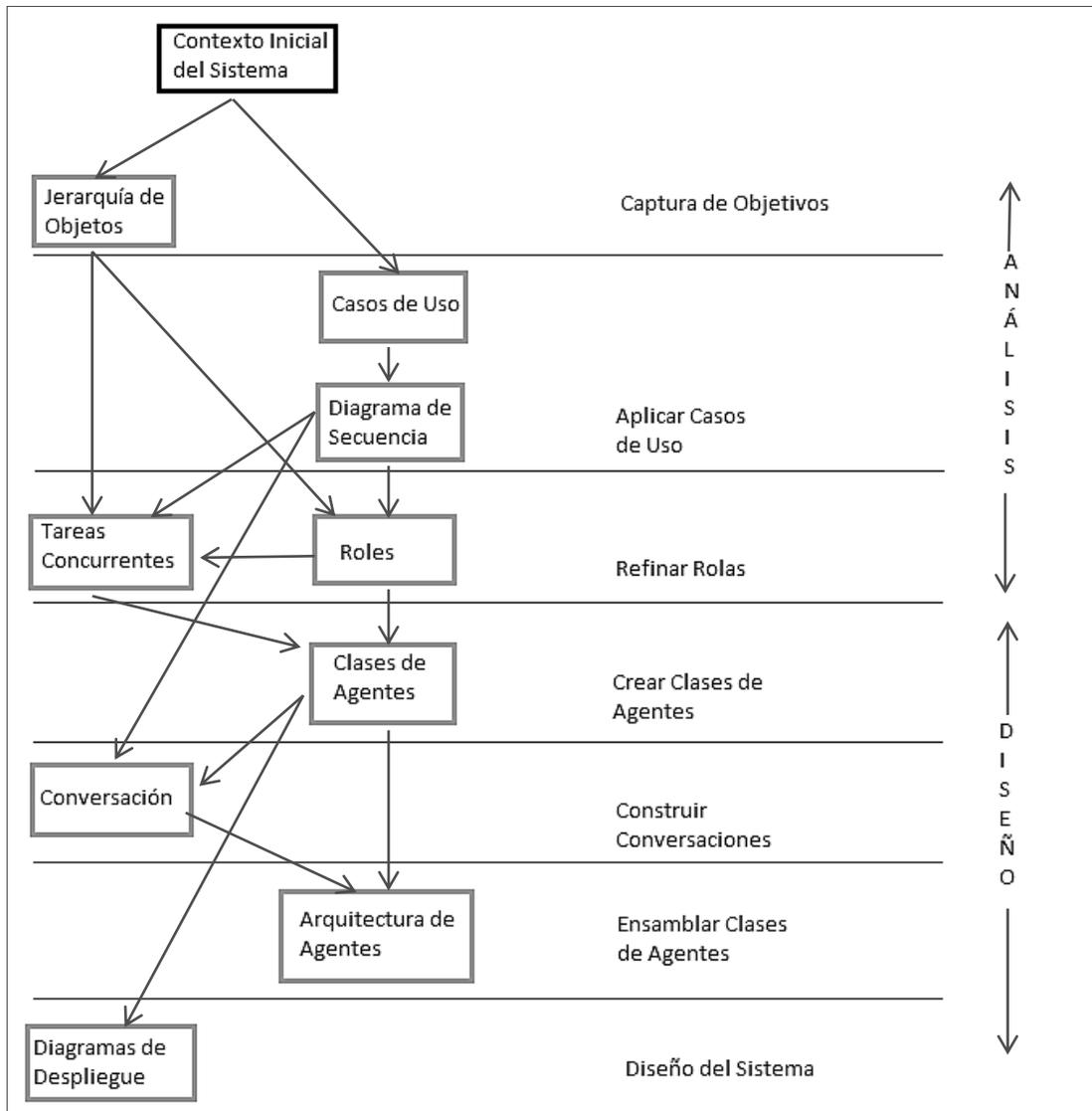


Figura 28 Fases de MaSE [Ana Mas. 2005]

4.6.4 PASSI

PASSI (Process for Agent Societies Specification and Implementation) es una metodología paso a paso para elaborar un proyecto orientado a agentes. Con el objetivo de diseñar y desarrollar sociedades multiagente, integrando modelos de diseño y conceptos de la ingeniería en software orientada a objetos y sistemas multiagente utilizando notación UML. El proceso de diseño de se compone por 5 modelos, y el proceso de construcción del sistema multiagente se compone por 12 pasos [Icar, 1923].

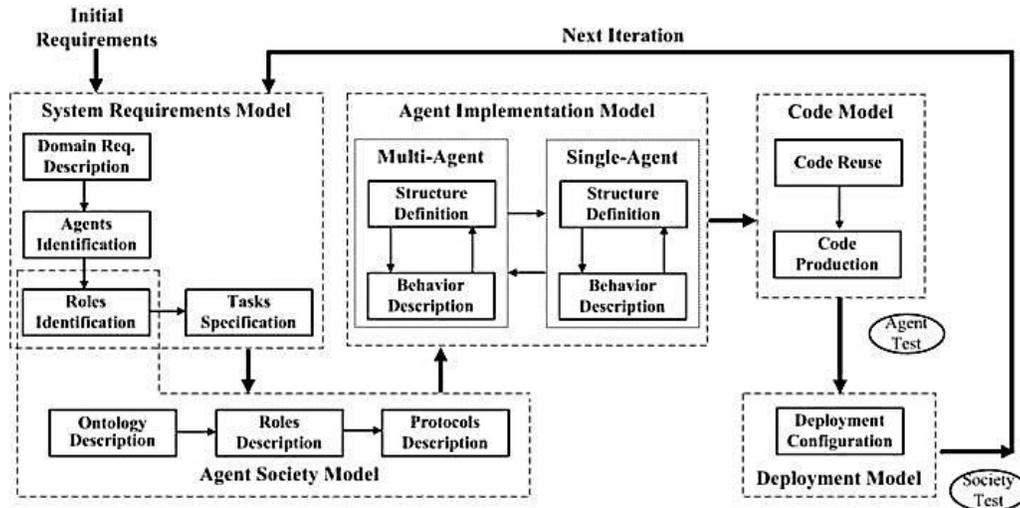


Figura 29 Metodología Passi: Modelos y fases [Icar, 1923]

Las fases y metodologías de PASII son [Icar, 1923]:

- ❖ **Modelo de requerimientos del sistema:** modelo antropomórfico de los requerimientos del sistema en términos de agente y objetivos. Este modelo involucra cuatro pasos para su desarrollo:
 - **Descripción del Dominio (D.D.):** descripción funcional del sistema utilizando diagramas de casos de uso convencionales.
 - **Identificación de Agentes (A. Id.):** separación de las responsabilidades de los agentes, representados como paquetes UML.
 - **Identificación de Roles (R. Id.):** uso de los diagramas de secuencia para explorar las responsabilidades de cada agente a través de escenarios específicos para cada rol.
 - **Especificación de Tareas (T. Sp.):** especificar las capacidades de cada agente por medio de los diagramas de actividad.
- ❖ **Modelo de Sociedad de Agentes:** modelo sobre las acciones y dependencias sociales entre los agentes que participan en la solución. Este modelo consta de tres fases:
 - **Descripción de la Ontología (O. D.):** mediante el uso de diagramas de clase y restricciones OCL para describir el conocimiento individual de los agentes y la pragmática de sus acciones.
 - **Descripción de los Roles (R. D.):** utiliza diagramas de clase para mostrar los distintos roles que puede poseer cada agente, las áreas que involucran a estos roles, las capacidades de comunicación y las dependencias entre agente.

- **Descripción de protocolos (P. D.):** usa los diagramas de secuencia para especificar la gramática de cada protocolo de comunicación en términos de performatives.
- ❖ **Modelo de implementación de Agentes:** el modelo de la arquitectura de la solución en términos de clases y métodos. Consta de dos fases:
 - **Definición de Estructura de Agentes (A. S. D.):** describe la estructura de la solución mediante el uso de diagramas de clase.
 - **Descripción del Comportamiento de los Agentes (A. B. D):** describe el comportamiento de los agentes individualmente mediante el uso de diagramas de actividad.
- ❖ **Modelos de Código:** da la solución a nivel de código y consta de dos fases:
 - **Reutilización de Código C. R.):** utiliza librerías de clases y diagramas de actividad con el código reusable asociado.
 - **Compleitud de Código (C. C.):** código fuente del objetivo del sistema.
- ❖ **Modelo de Despliegue:** es un modelo de la distribución de las partes del sistema a través de las unidades de procesamiento del hardware y su migración entre las unidades de procesamientos. Consta de una fase:
 - **Configuración de Despliegue (D. C.):** describe la ubicación de los agentes en las unidades de proceso disponibles y todas las restricciones de migración y movilidad, mediante el uso de diagrama de despliegue.

4.7 Plataformas Multiagente

La finalidad de las plataformas es permitir la interconexión y ejecución de los agentes en un SMA. Algunas se basan en el estándar FIPA y otras en la propia arquitectura de agente. A continuación se presentan algunas de las plataformas existentes.

4.7.1 JADE

JADE (Java Development Framework) es un middleware que cumple las especificaciones FIPA y que simplifica la implementación de sistemas multiagente, a través de un conjunto de herramientas gráficas que soportan la depuración y las fases de despliegue. Está basado en la arquitectura de comunicación peer-to-peer [Jade, 2003].

Jade proporciona varios agentes ya implementados, los cuales pueden ser utilizados con diversos propósitos, y están ligados a la plataforma, como son [Ana Mas, 2005]:

- ❖ **Directorio Facilitador (DF):** proporciona los servicios de directorio y puede conectarse a otros agentes al mismo tiempo para gestionar, si fuera preciso, una estructura jerárquica de dominios.

- ❖ **Agent Management System (AMS):** proporciona un servicio de páginas blancas y de ciclo de vida, manteniendo una lista de los identificadores de los agentes (AID) y el estado en el que están. Cada agente particular, lo primero que tiene que hacer es registrarse en el AMS, para así obtener un identificador válido y quedar registrado en el sistema.
- ❖ **Sniffer:** ayuda en la depuración de las interacciones, entre agentes. Este agente puede intercambiar mensajes entre distintos agentes, y mostrarlos gráficamente para poder depurar sociedades de agentes, viendo el intercambio de mensajes que hay entre ellos.
- ❖ **Instrospector:** permite controlar el ciclo de vida de un agente en ejecución y los mensajes que este intercambia.

4.7.2 FIPA-OS

FIPA Open Source, es un toolkit basado en componentes que permiten el desarrollo rápido de agentes compatibles con FIPA. Provee una arquitectura basada en componentes para permitir el desarrollo de agentes específicos del dominio que pueden utilizar los servicios de los agentes de la plataforma FIPA [Poslad, 2001].

El objetivo principal de FIPA-OS es reducir las barreras actuales en la adopción de la tecnología de FIPA, a través de complementar los documentos de la especificación técnica con código fuente abierto. Está diseñado para soportar los estándares de agentes FIPA, el modelo de referencia de este define los componentes centrales del paquete de distribución de FIPA-OS, que serían: El Facilitador de Directorios (DF), El Servicio de Administración de Agentes (AMS), el Canal de Comunicación de Agentes (ACC) y el Transporte de Mensajes Interno de la Plataforma (IPMT) [Poslad, 2001].

Además de los componentes obligatorios del modelo de referencia de FIPA, incluye soporte para: Diferentes tipos de Shells, para producir agentes que se puedan comunicar con los demás, usando para ello las facilidades de FIPA-OS, Soporte para la comunicación de agentes multicapa, Administración de conversaciones y mensajes, Herramientas de visualización y diagnóstico, y otro [Poslad, 2001].

5 Propuesta de Desarrollo

Para poder poner a prueba todas las técnicas, conceptos y tecnologías vistas anteriormente, realizaremos un prototipo, y como el proyecto va enfocado a los hogares, simularemos un ambiente lo más parecido a este. En este caso se recreará una sala de estudio de una casa. Los motores y sensores de los robots Lego, serán los sensores y actuadores de nuestro prototipo domótico, que estarán controlados por el micro controlador de lego, el cual será programado en Java, y comunicado, a través de bluetooth con los agentes alojados en el computador.



Figura 30 Prototipo de Escenario [Home, 2011]

5.1 Modelado del Sistema

5.1.1 Modelo de Requerimientos del Sistema

5.1.1.1 Modelo de Descripción del Dominio

En este modelo se identifican todas las entidades que interactúan con el sistema y se entrega una descripción funcional del sistema. Se desarrolla utilizando diagramas de casos de uso. En este sistema tenemos el actor “Administrador” es el encargado de “Administrar el Sistema”, esta administración se divide en varias sub tareas que son: gestionar sistema y controlar sensores. [Cossen, 2003]

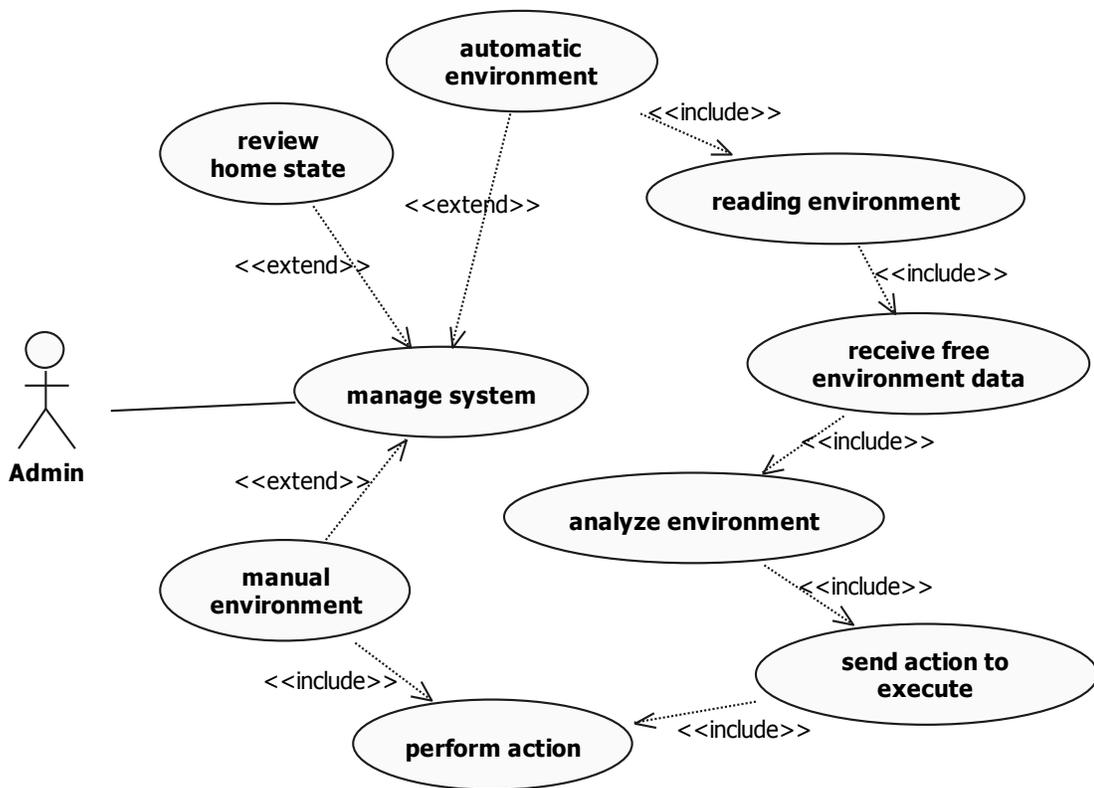


Figura 31 Modelo de Descripción del Dominio [Elab. Propia]

5.1.1.2 Modelo de Identificación de Agentes

Este modelo se realiza encapsulando el Modelo de Requerimientos del Sistema, y así van apareciendo los agentes, los cuales se mantendrán a lo largo de todo el proceso. [Cossen, 2003]

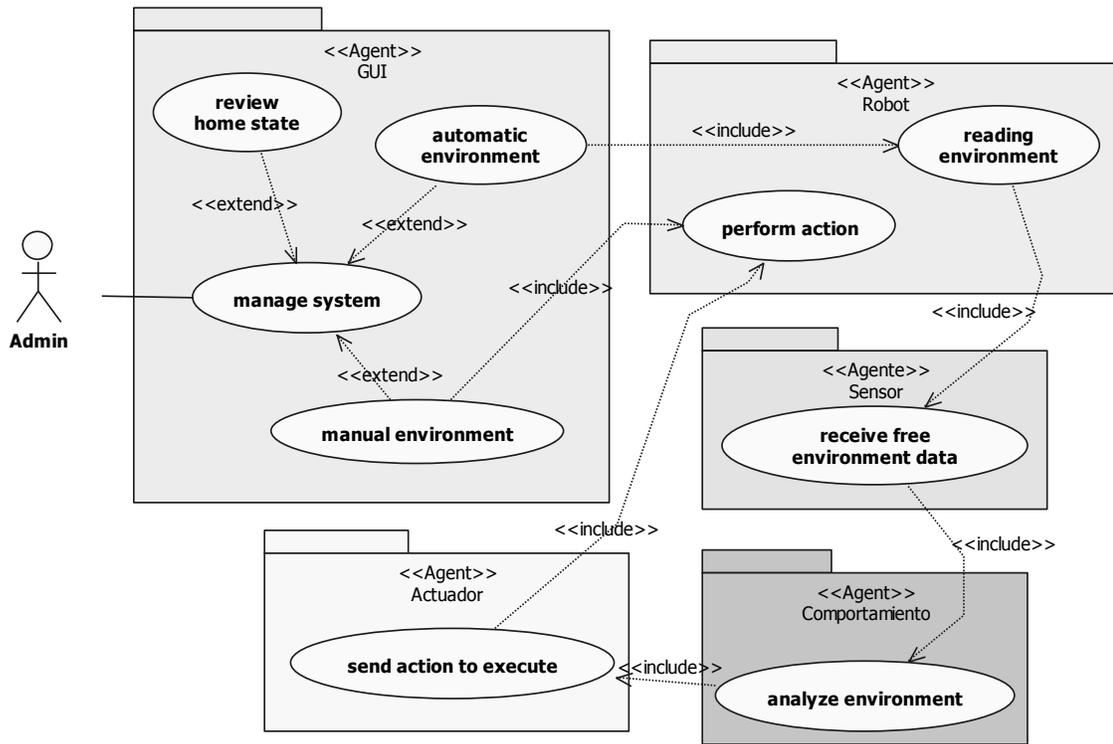


Figura 32 Modelo Identificación de Agentes [Elab. Propia]

5.1.1.3 Modelo de Identificación de Roles

Este modelo permite representar los diversos roles que desenvuelven los agentes en los distintos escenarios. Hay que tener presente que un agente puede poseer distintos roles, según el escenario que enfrente. Para desarrollarlo se utilizan diagramas de secuencia. [Cossen, 2003]

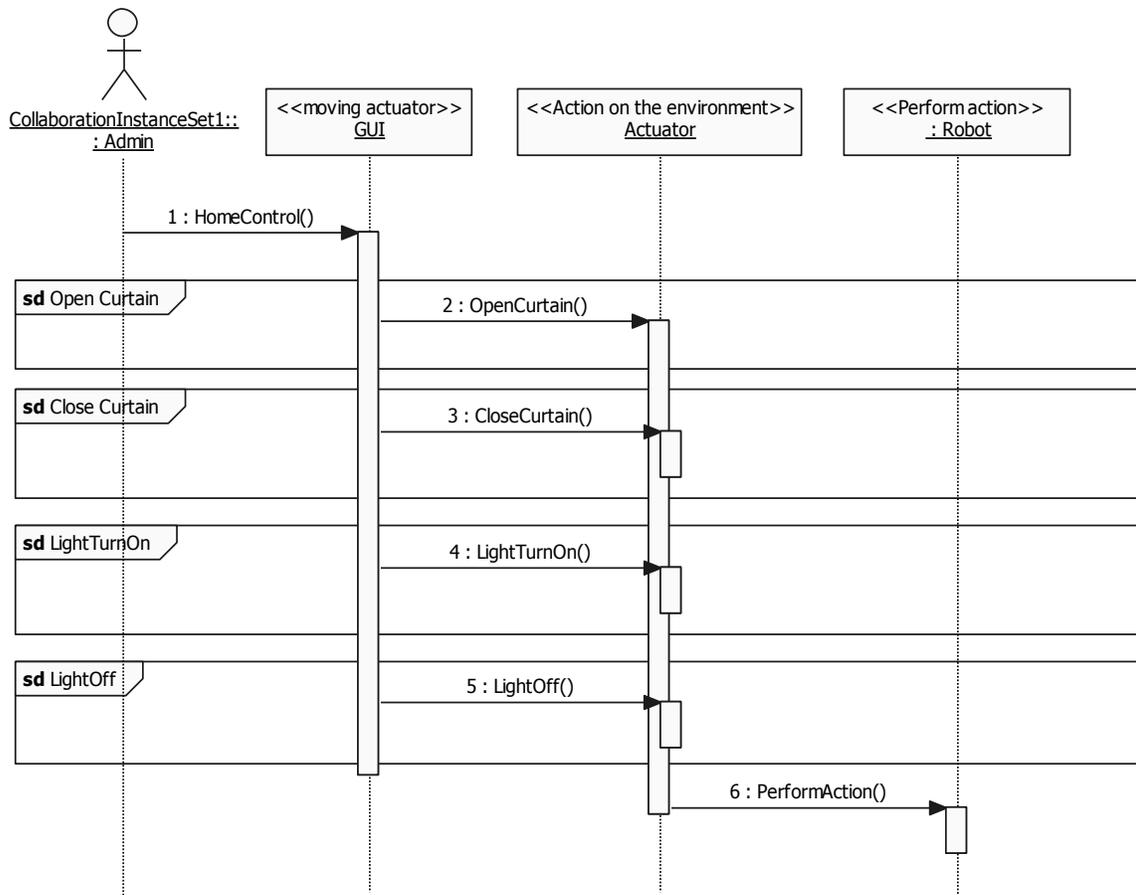


Figura 33 Modelo de Identificación de Roles. Escenario: Ambiente Manual [Elab. Propia]

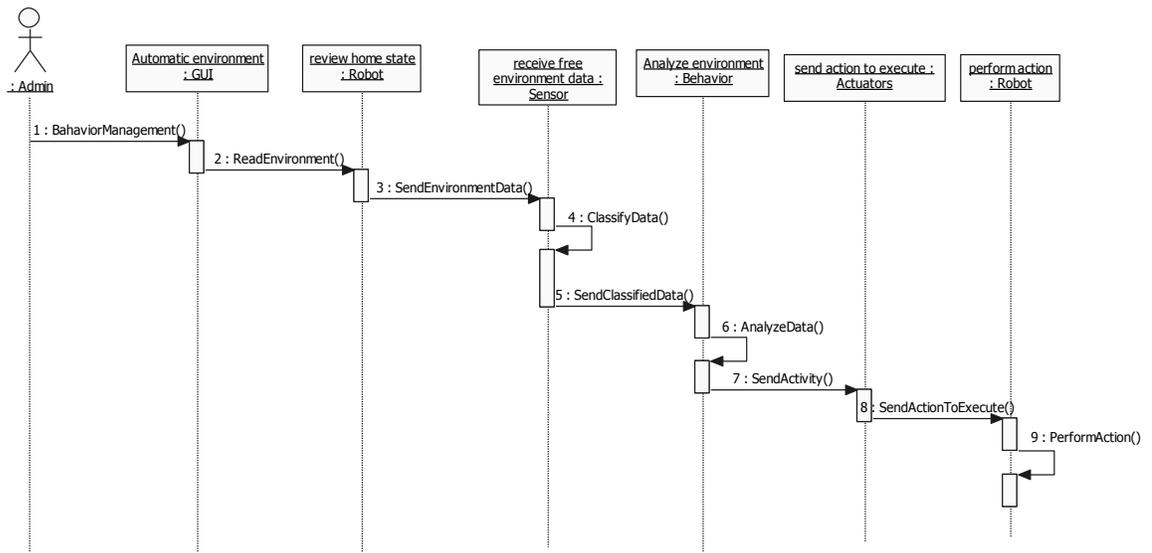


Figura 34 Modelos Identificación de Roles. Escenario: Ambiente Automático [Elab. Propia]

5.1.1.4 Modelo Especificación de Tareas

Esta fase utiliza diagramas de actividad para describir todas las actividades que realizan los agentes para obtener sus objetivo. Para esto se utilizan dos calles (Swim-Lanes), una para el agente al que se le están describiendo las tareas, y el otro para los demás agentes involucrados. [Cossen, 2003]

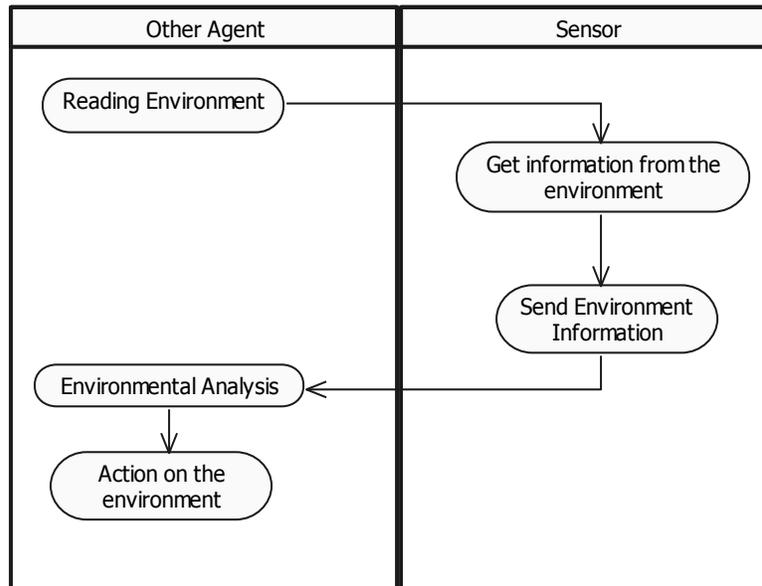


Figura 35 Modelos Especificación de Tareas: Sensor [Elab. Propia]

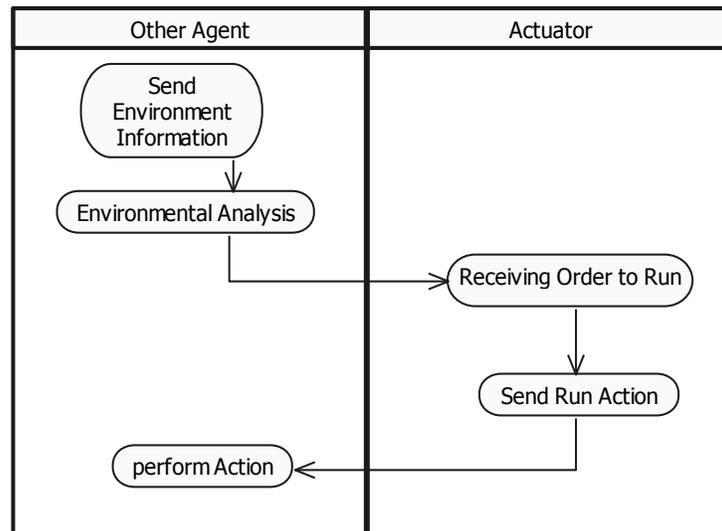


Figura 36 Modelos Especificación de Tareas: Actuator [Elab. Propia]

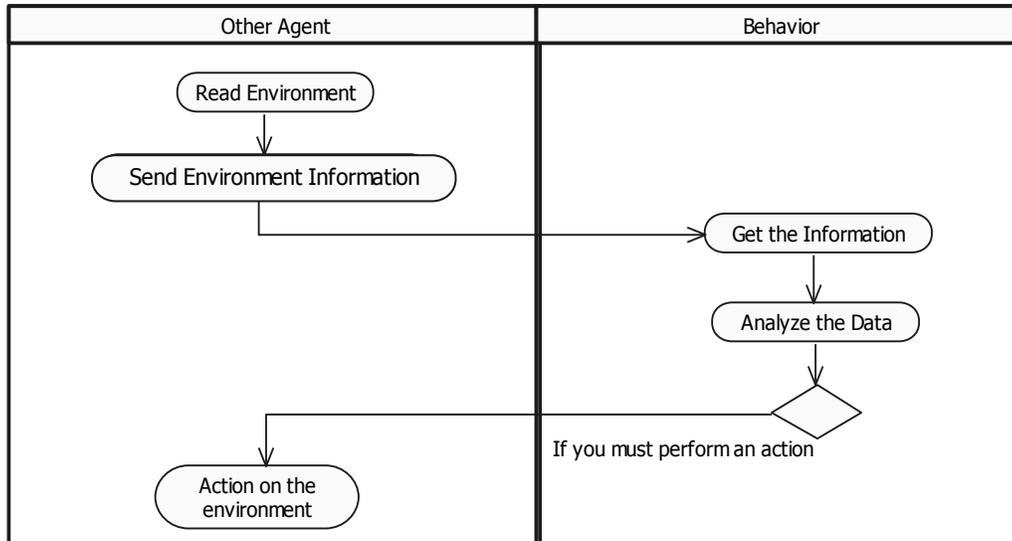


Figura 37 Modelos Especificación de Tareas: Comportamiento [Elab. Propia]

5.1.2 Modelo de Sociedad de Agentes

5.1.2.1 Modelo de Descripción de la Ontología del Dominio

Mediante un diagrama de clases se representa la ontología del dominio, este representa las entidades mediante clases y además las interacciones entre ellos. [Cossen, 2003]

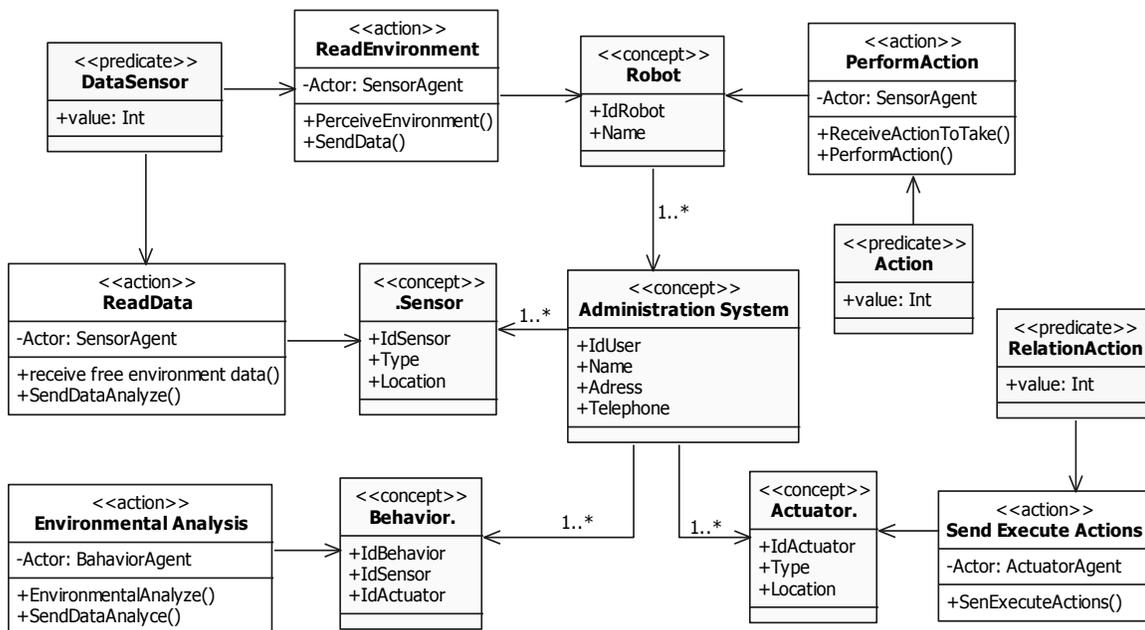


Figura 38 Modelo de Descripción de la Ontología del Dominio [Elab. Propia]

5.1.2.2 Modelo de Descripción de la Ontología de Comunicación

Este representa la comunicación entre agentes, mediante diagramas de clases. Cada agente se describe en términos de conocimientos y se muestran los roles jugados durante la comunicación. [Cossen, 2003]

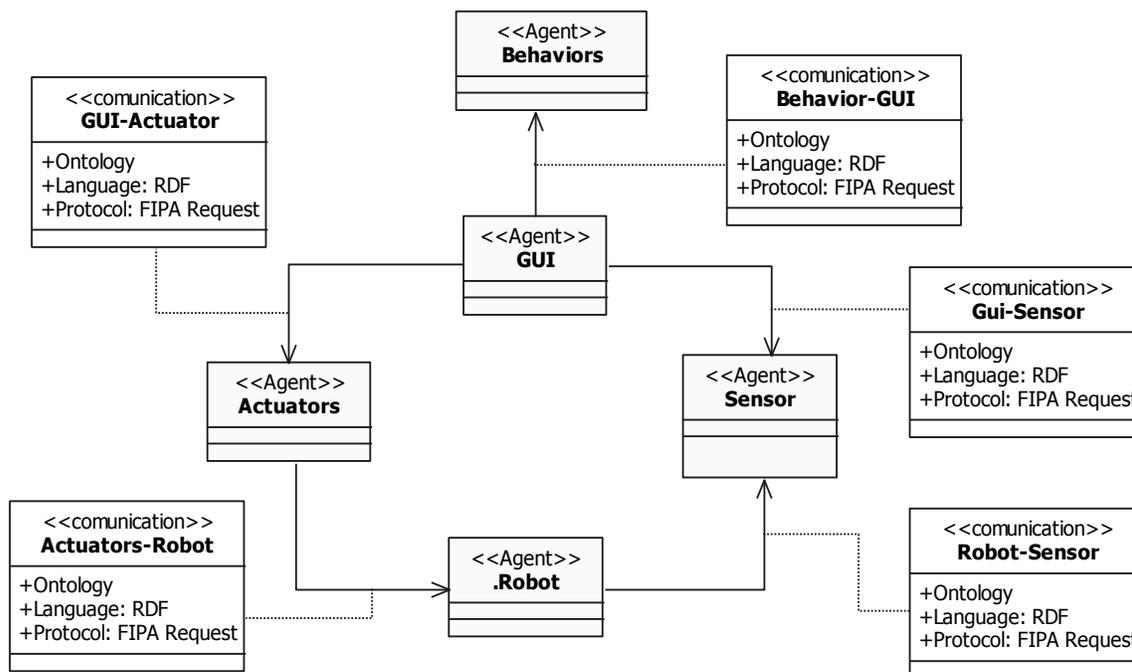


Figura 39 Modelo de la Ontología de la Comunicación [Elab. Propia]

5.1.2.3 Modelo de Descripción de Roles

Esta refleja es ciclo de vida de cada agente, identificando los roles desempeñados, las colaboraciones que necesita y las comunicaciones en las cuales participa. Esta información se representa con diagramas de clase, y los roles de cada agente son agrupados en paquetes. [Cossen, 2003]

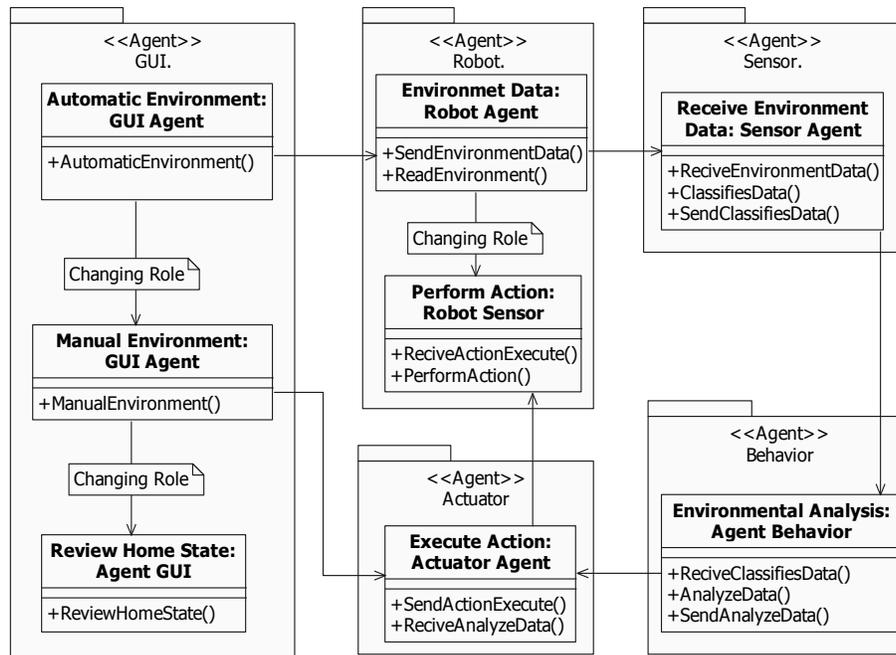


Figura 40 Modelo de Descripción de Roles [Elab. Propia]

5.1.2.4 Modelo de Descripción de Protocolos

Mediante un diagrama de secuencia, se pretende representar los protocolos de comunicación estándares, que pueden ser utilizados en el proceso.

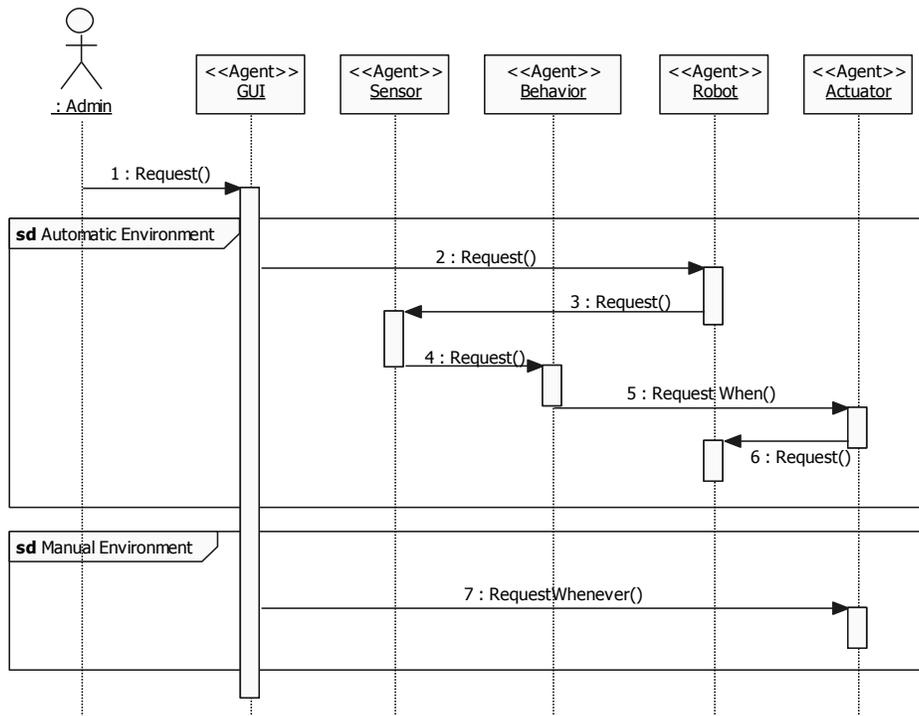


Figura 41 Modelo de Descripción de Protocolos [Elab. Propia]

5.1.3 Modelo de Implementación de Agentes

5.1.3.1 Modelo de Definición de Estructura de Agentes

Estos representan dos niveles de detalles. Primero es el de los agentes individualmente, mostrando sus métodos y atributos, además de las tareas internas. Todos estos métodos y atributos que fueron definidos en este modelo, se utilizarán para la generación del código (Próxima fase de modelado). Y el segundo nivel es el multiagente, el cual se enfoca en la arquitectura del sistema general, en el cual se ven los agentes y las tareas que realiza. [Cossen, 2003]

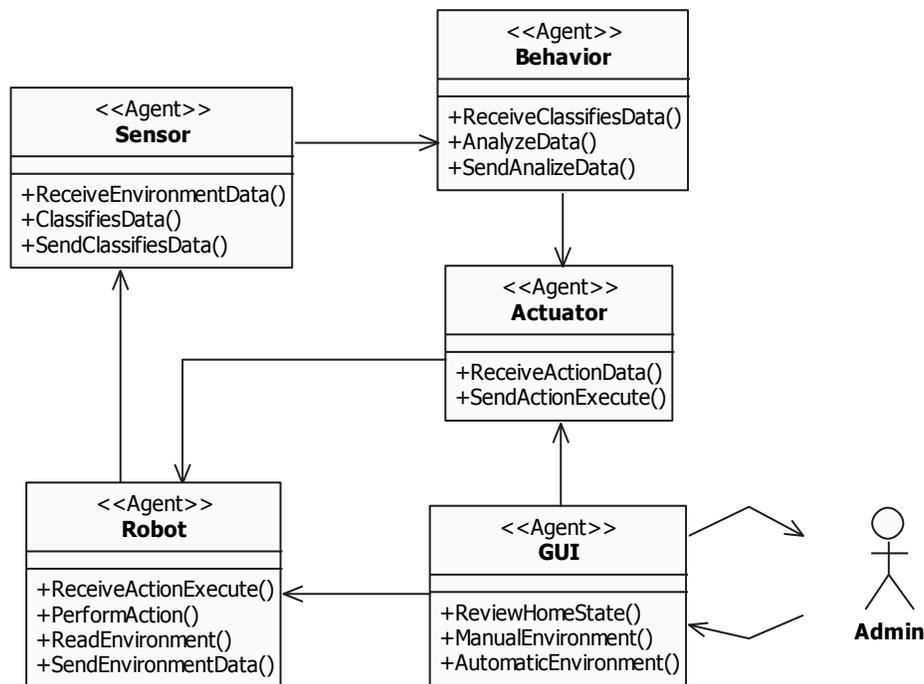


Figura 42 Modelo de la Definición de Estructura Multiagente [Elab. Propia]

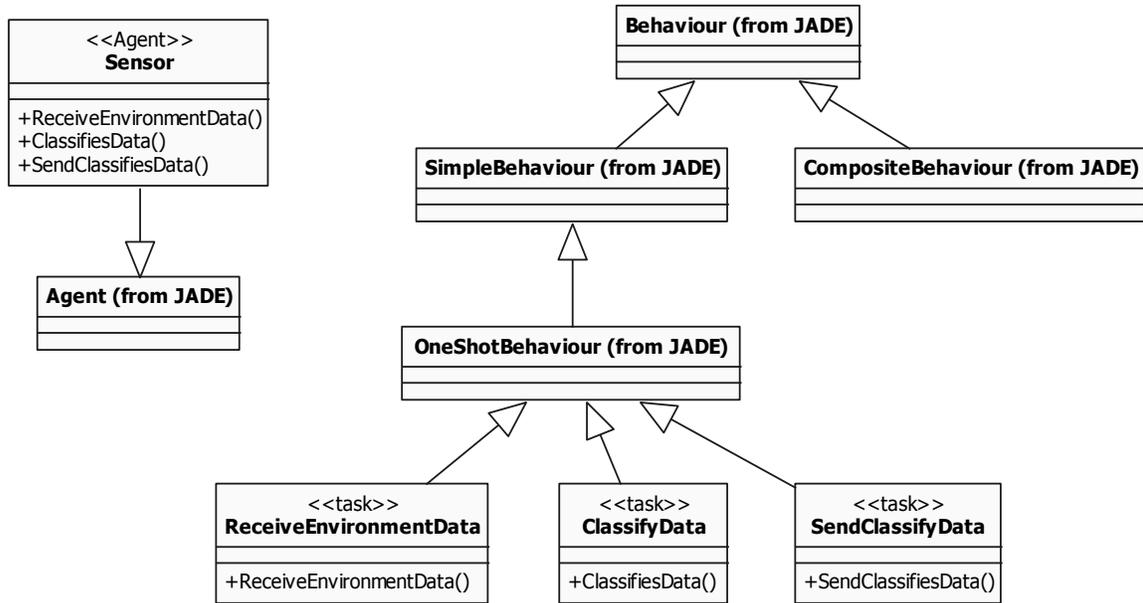


Figura 43 Modelo de la Descripción de la Estructura del Agente Sensor [Elab. Propia]

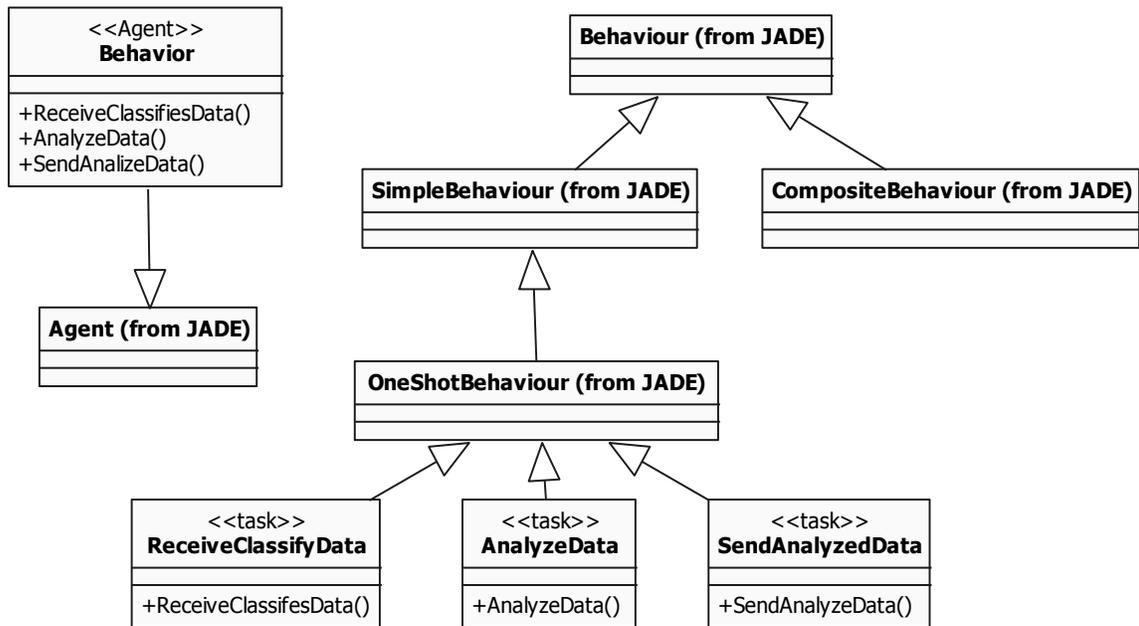


Figura 44 Modelo de la Descripción de la Estructura del Agente Comportamiento [Elab. Propia]

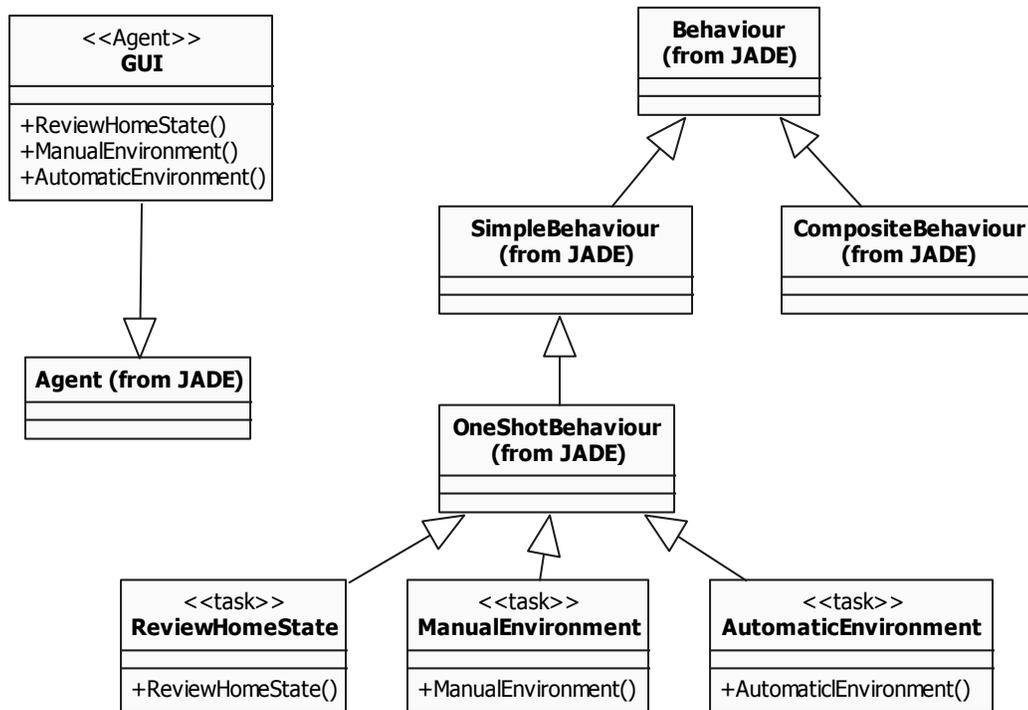


Figura 45 Modelo de la Descripción de la Estructura del Agente GUI [Elab. Propia]

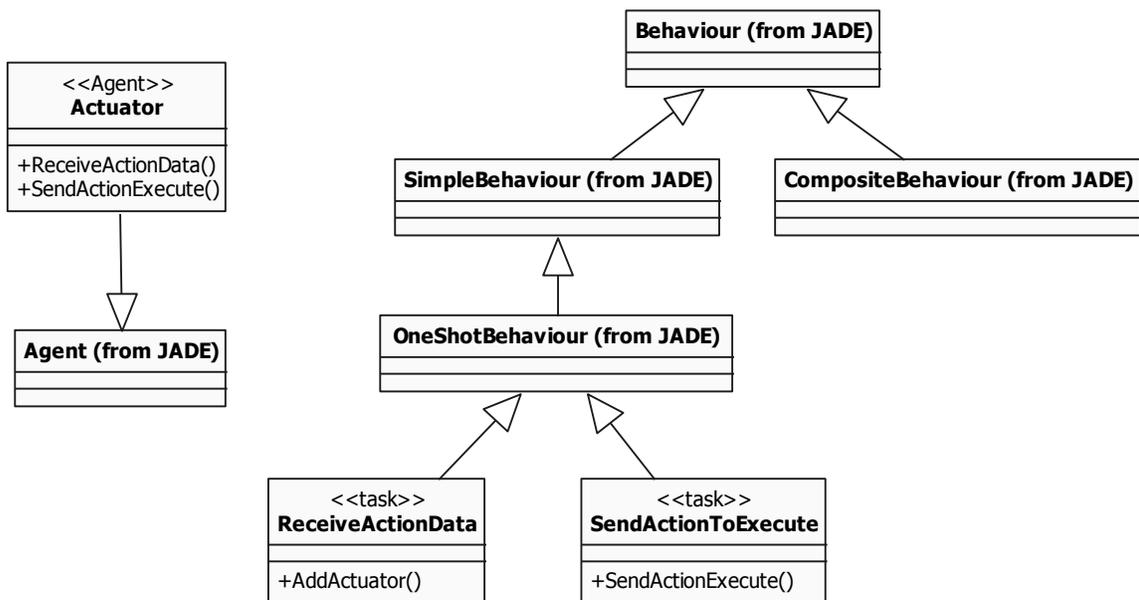


Figura 46 Modelo de la Descripción de la Estructura del Agente Actuador [Elab. Propia]

5.1.4 Modelo de Despliegue

5.1.4.1 Modelos de Configuración de Despliegue

Mediante diagramas de despliegue se describen la dispersión de los agentes a través de otras plataformas disponibles y sus movimientos. [Cossen, 2003]

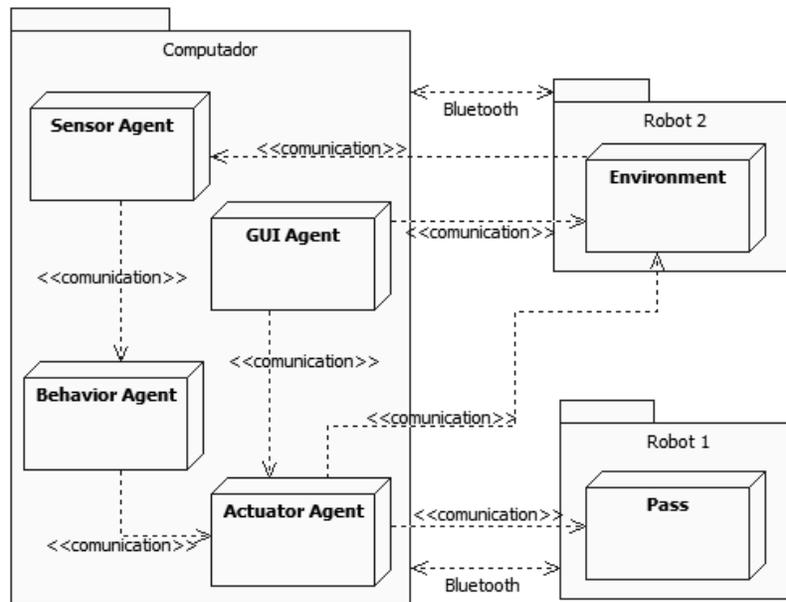


Figura 47 Modelo Configuración de Despliegue [Elab. Propia]

5.2 Esquema del Sistema

El sistema será implementado en un espacio del hogar, donde cada sensor y motor de los robots lego tendrán una posición establecida. Se contará con 2 bricks, uno que se encargará de leer el ambiente y controlar a los actuadores, y el segundo trabajará como alarma, es decir, se encargará de pedir una clave cada vez que la puerta sea abierta y de realizar los sonidos que correspondan, dependiendo si la clave fue correcta o no. Los agentes que estarán alojados en el computador serán los encargados de aportar cierta independencia al sistema, pudiendo controlar y decidir cuándo deben realizar ciertas acciones los actuadores, dependiendo de las lecturas recibidas que fueron realizadas por los sensores. La finalidad de esto es lograr un sistema autónomo, que puede ser controlado manual o automáticamente, aplicando los conceptos ya vistos de domótica, robótica y la tecnología de agentes.

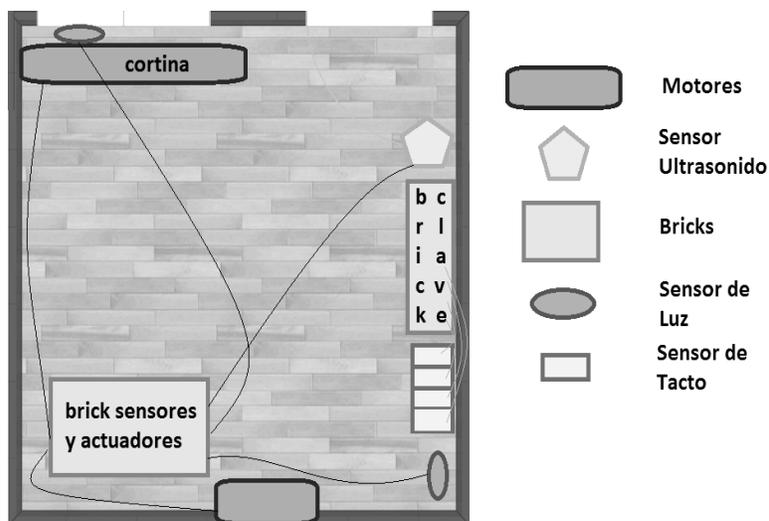


Figura 48 Idea de Futura Implementación [Home, 2011]

En la Figuras 48 se muestran unas imágenes de propuesta de implementación, y podemos ver los sensores y actuadores que involucrará el sistema. Se utilizarán dos bricks, uno para el ambiente el cual contendrá tres sensores: uno de ultrasonido y dos de luz, además de dos actuadores: uno que abrirá y cerrará las cortinas y otro para prender y apagar la luz de la habitación. El segundo brick será para la clave y este tendrá cuatro sensores de tacto para generar dicha clave. El centro de control será el computador que tendrá contenido a los agentes, y serán quienes analicen la información de los sensores y envíen las acciones a realizar.

5.3 Caso de Estudio

Para probar el prototipo se realizarán dos casos de estudio. El primero estará enfocado al confort de las personas, será implementado con un sensor de luz de los robots lego, el cual dependiendo de las variaciones de luz abrirá o cerrará las cortinas, y encenderá la luz. Y el segundo enfocado en la seguridad, que será implementar una alarma.

5.3.1 Caso de Estudio: Confort

El primer caso de estudio que veremos está asociado al confort, y este será el abrir o cerrar una cortina, y prender la luz de la habitación, dependiendo de la luz ambiente. Para realizar este caso de estudio como se aprecia en la Figura 49, se utilizarán dos sensores de luz de los robots NXT, uno se encontrará en la ventana y el otro dentro del hogar, y estará conectado a un brick. Este a su vez tendrá conectado dos motores: uno para poder abrir o cerrar las cortinas y otro para poder prender o apagar la luz de la habitación. En la parte de Software, se utilizará la tecnología de agentes para analizar la información de los sensores del robot y poder indicar cual o cuales son las acciones necesarias a seguir.



Figura 49 Caso de Estudio: Confort [Elab. Propia]

5.3.2 Caso de Estudio: Seguridad

Para este caso de estudio se generará una alarma, la cual será construida por 4 sensores de tacto y uno de ultrasonido. Los de tacto estarán conectados a un brick, y el de ultrasonido estará conectado al otro brick, que sería el mismo usado en el caso de estudio anterior. El sensor de ultrasonido le indicará al computador cuando abran la puerta, y este le avisará al otro brick para que pida la clave. Esta clave se genera presionando en un orden predeterminado los sensores de tacto. Se tendrán 4 intentos para ingresarla correctamente, en caso de ingresarla incorrectamente sonará una alarma.

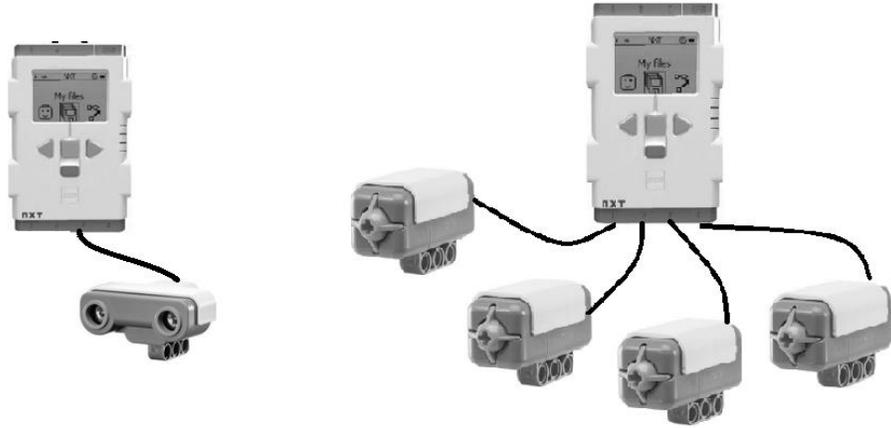


Figura 50 Caso de Estudio: Seguridad – Alarma [Elab. Propia]

6 Implementación de la Propuesta

6.1 Creación de la Maqueta para presentar el caso de Estudio

Para poder presentar el caso de estudio se creó una maqueta que representa un espacio del hogar, en una primera instancia sus medidas fueron de 25x25x50 cm. Como se muestra en las figuras 52 y 53

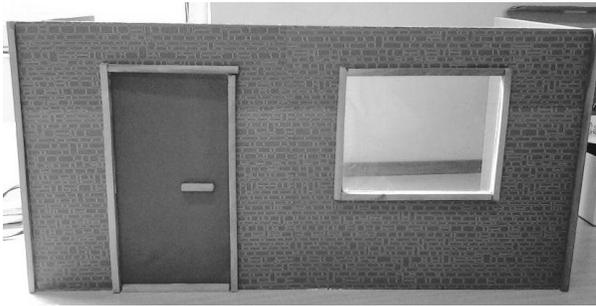


Figura 51 Prototipo 1 de maqueta [Elab. Propia]

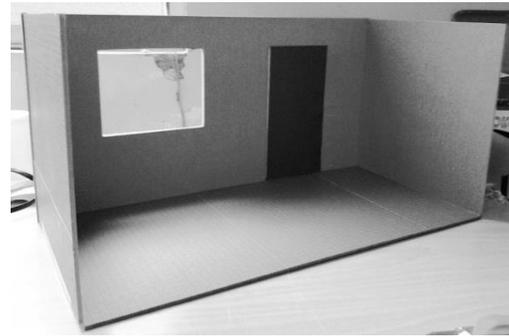


Figura 52 Prototipo 1 de maqueta [Elab. Propia]

Luego se comprobó que a la maqueta le faltaba un poco de espacio, por lo cual, se decidió realizar una ampliación, quedando de 50x50x50. Además se le agregó una instalación eléctrica para poder realizar de mejor forma la prueba número 1 “confort”.

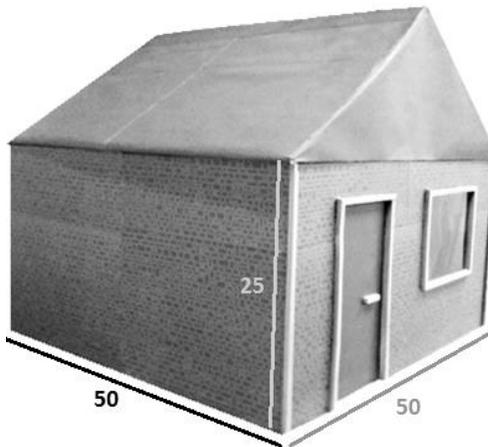


Figura 53 Maqueta Final Frontis [Elab. Propia]



Figura 54 Maqueta Final Interior con la luz encendida

[Elab. Propia]



Figura 55 Maqueta Final con la puerta abierta

[Elab. Propia]

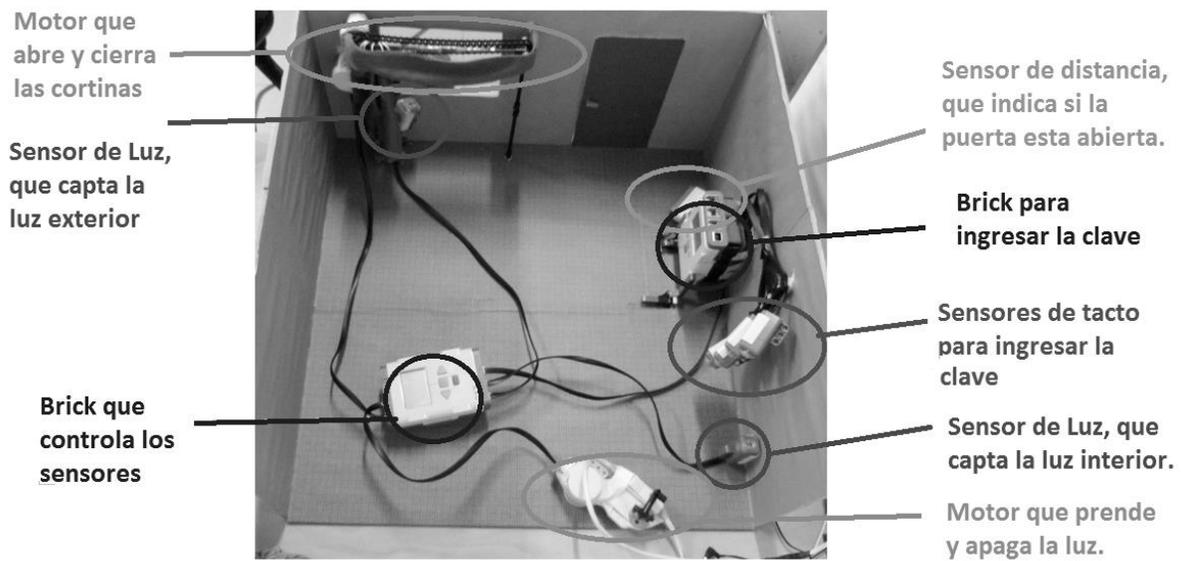


Figura 56 Ubicación de los sensores y actuadores dentro del prototipo.

6.2 Herramientas a utilizar

Para el desarrollo de este trabajo se utilizó como entorno de desarrollo: “Eclipse”

6.2.1 Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE). Fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para Visual Age, ahora es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. La primera versión de Eclipse fue la 3.0,

lanzada en 2004, actualmente la última versión es la 4.2 lanzada en el 2012. (Eclipse, 2012)

Eclipse se puede descargar de forma gratuita desde su página. A continuación veremos la configuración de eclipse para LeJOS y luego para Jade.

6.2.1.1 LeJOS y Eclipse

Eclipse se integra perfectamente a LeJOS, a diferencia de otros IDE. Para configurarlo con LeJOS basta con instalar el plugin desde la página "<http://lejos.sourceforge.net/tools/eclipse/plugin/nxj/>". Este plugin agregará una pestaña al menú llamada LeJOS NXJ, la cual nos permite actualizarle el firmware a los robots, además agregará dos tipos de proyecto. El primero es el NXT-Project los cuales son programas cargados y ejecutados en el mismo brick del NXT. El segundo tipo es el PC-Project, el cual es un tipo de proyecto el cual se ejecuta en el computador y le envía acciones al NXT para que este realice, se pueden conectar mediante Bluetooth o USB.

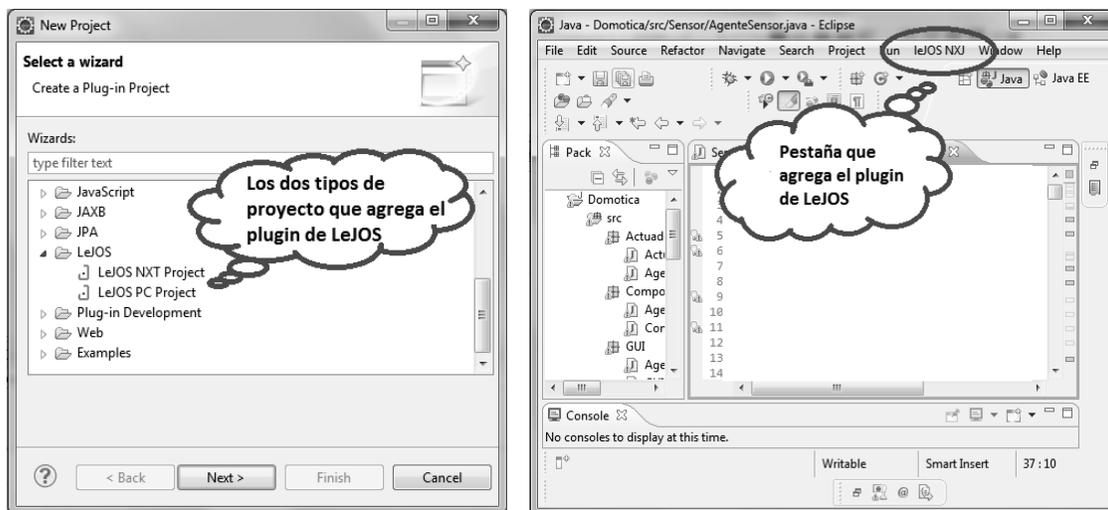


Figura 57 Eclipse y LeJOS

En este proyecto se utilizaron estos dos tipos de proyectos LeJOS. En el PC-Project se programaron los agentes y se le envía la información al NXT de que es lo que debe realizar, como puede ser leer algún sensor, o mover algún motor. En el otro proyecto, NXT-Project, el NXT recibirá la información que le envía el computador, y en base a esto realiza la acción debida.

En la Figura 57 se puede observar el código para que el computador busque a un NXT es este caso “NIKITA” y se conecten. Este código debe ir en el PC-Project. En la Figura 58 vemos el código para que el NXT espere hasta que algún dispositivo mediante Bluetooth se comuniquen con él. En ambos casos, se utilizan las variables `DataOutputStream` y `DataInputStream`, estas son las encargadas de enviar y recibir datos respectivamente.

```

public DataOutputStream dos; //Para enviar instrucciones al NXT
public DataInputStream dis; //Para recibir información del NXT
public void conectar() {

    //Conección
    conn = new NXTConnector(); //Crea la conexión con NXT
    conn.addLogListener(new NXTCommLogListener(){ //Para que escuche la conexión
        public void logEvent(String message) {
            System.out.println("BTSend Log.listener: "+message);
        }
        public void logEvent(Throwable throwable) {
            System.out.println("BTSend Log.listener - stack trace: ");
            throwable.printStackTrace();
        }
    });
    /**Conectamos al robots con el PC*/
    boolean connected = conn.connectTo("NIKITA"); // NIKITA == es el nombre del NXT
                                                //con el que se conectara

    if (!connected) {
        System.err.println("No se pudo conectar con NIKITA");
        System.exit(1);
    }

    dos = new DataOutputStream(conn.getOutputStream()); |
    dis = new DataInputStream(conn.getInputStream());
}

```

Figura 58 Código de conexión con el NXT

```

LCD.clear();
LCD.drawString("Preparando Conexión",0,0);
LCD.refresh();

BTConnection btc = Bluetooth.waitForConnection();

LCD.clear();
LCD.drawString("Conectado",0,0);
LCD.refresh();

DataInputStream dis = btc.openDataInputStream();
DataOutputStream dos = btc.openDataOutputStream();

```

Figura 59 Código de espera de conexión del NXT

Para LeJOS con Eclipse, no es necesario agregar los .jar, dado que estos vienen añadidos en por defecto en ambos tipos de proyecto LeJOS.

6.2.1.2 Eclipse, LeJOS y Jade

Al comenzar a investigar como poder relacionar estos tres componentes, se pensó que sería un poco complicado, pero, en realidad poder combinarlos no tiene mayor complejidad. En el tipo PC-Project se debe modificar lo mismo que en un proyecto java, es decir, cambiar en las propiedades de ejecución del proyecto el main-project por "jade.Boot", y agregar como argumento "-gui". Además de estas modificaciones hay que agregar las librerías para trabajar jade que serían el "jade.jar" y "commons-codec-1.3.jar".

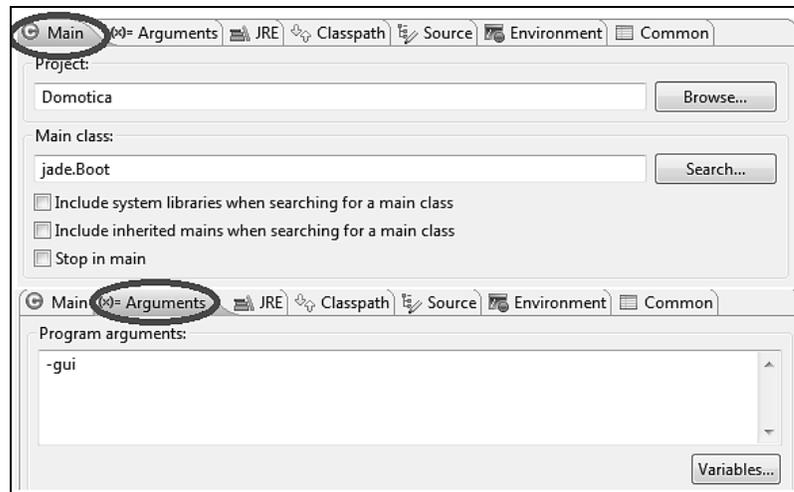


Figura 60 Configuración de Jade

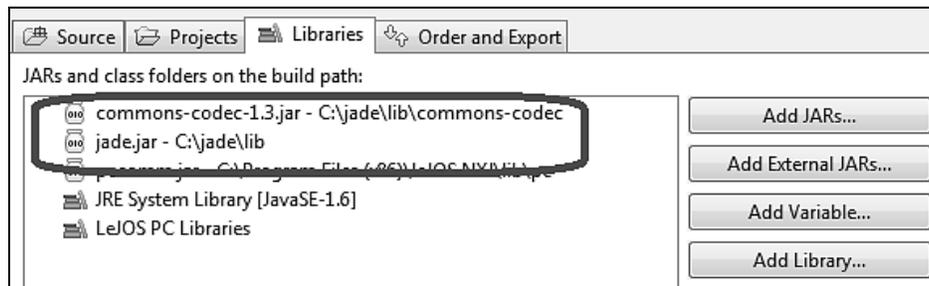


Figura 61 Jar de Jade

6.3 Comunicación entre PC y NXT-Project

Para la comunicación entre el robot y PC se utilizaron unos códigos tipo entero, para que así el robot pueda entender qué es lo que el PC necesita que realice, y para que el PC sepa cuáles son las condiciones del hogar. A continuación se presentan las tablas en las cuales se presenta la lógica de los códigos.

Tabla 6 Códigos enviados del robot al PC

Código	Que indica	Descripción
1 (valor leído)	La intensidad de luz que lee el sensor que se encuentra en la ventana.	Primero se envía un “1”, y seguido se envía el valor que está leyendo el sensor.
2 (valor leído)	La intensidad de luz que lee el sensor que se encuentra dentro del hogar.	Primero se envía un “2”, y seguido se envía el valor que está leyendo el sensor.
3 (0 ó 1)	Si la puerta está abierta o cerrada.	Primero se envía un “3”, y seguido se envía “1” si la puerta está abierta y “0” si está cerrada.

Tabla 7 Códigos enviados del PC al robot

Código	Descripción
41	Abrir las cortinas
42	Cerrar las cortinas
43	Prender la luz
44	Apagar la luz
98	Trabajar de modo Automático
99	Trabajar de forma Manual

6.4 Desarrollo del prototipo

Para desarrollar el prototipo de sistema domótico, se realizó una ardua investigación para comprender la comunicación entre Robot y PC. Se realizaron los dos casos de estudio en un mismo programa, el cual va evaluando las condiciones del hogar y realizando las acciones correspondientes de forma automática, pero de la misma manera, hay acciones que se pueden realizar de forma “manual” desde el computador, como abrir o cerrar las cortinas.

La interfaz para poder controlar el hogar manualmente es muy sencilla y cuenta con 7 botones. Uno para controlar la casa automáticamente, otro para controlarla manualmente, este último activa cuatro botones que son: Abrir las cortinas, Cerrar las cortinas, Prender la luz y Apagar la luz, y el último que muestra el estado de hogar, indicando el estado de la puerta, cortina y luz.

En la Figura 62 se puede observar a la interfaz con el modo automático activo. La figura 63 muestra al estado manual activado, se puede apreciar que aparecen dos botones bloqueados, esto se da porque esas condiciones ya están dadas, es decir, la luz está prendida y la cortina cerrada. En la Figura 64 se puede apreciar la interfaz que muestra el estado actual del sistema.



Figura 62 Interfaz del Prototipo - Control Automático



Figura 63 Interfaz del Prototipo - Control Manual

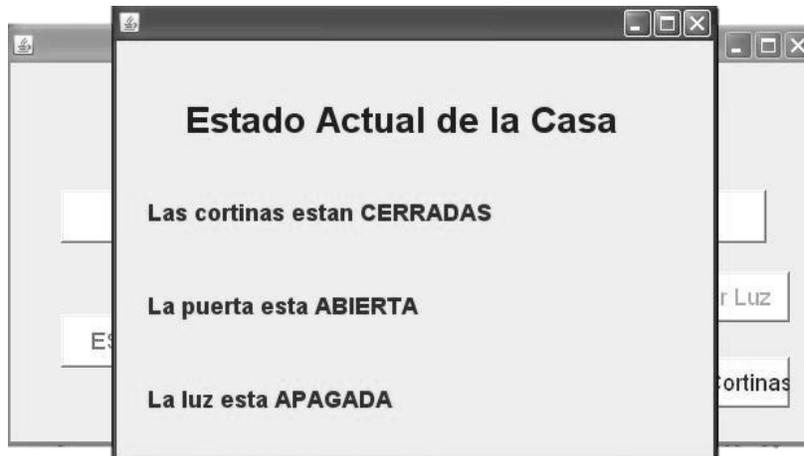


Figura 64 Interfaz del Prototipo - Estado Actual del Hogar

Las condiciones del sistema están dadas por dos escenarios básicos de luminosidad. El primero es cuando hay poca luminosidad al exterior del hogar y dentro del hogar, y la luz de la casa está apagada, en este caso se cerrará la cortina. En caso que estas hubiesen estado abiertas, se prenderá la luz. El segundo caso, es cuando estando la luz apagada del hogar, la luminosidad externa es alta, en este caso, se abrirá la cortina considerando que esta se encuentre cerrada.

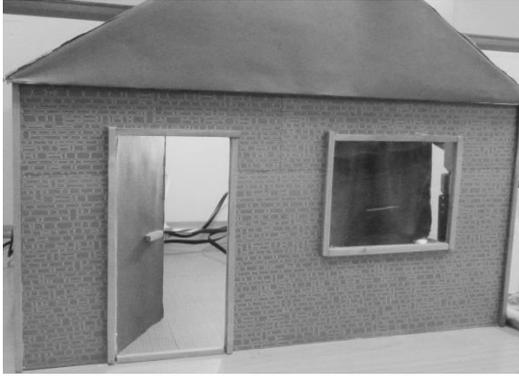


Figura 65 Escenario básico Uno.



Figura 66 Escenario básico Dos.

Además de la luminosidad del hogar también se trabajó en la seguridad de este, para esto se utilizó un sensor de distancia, el cual cuando detecte que la puerta ha sido abierta, enviará la información al PC, y este le avisará al otro brick para que solicite la clave. Este brick posee cuatro sensores de tacto, que representan los números del 1 al 4 y la clave está representada por cuatro dígitos, en este caso la clave es “1-2-3-4”. Existen cuatro intentos para ingresar la clave correctamente antes que suene la alarma.

En la Figura 67 se muestra la pantalla del brick luego que la puerta haya sido abierta. En la Figura 68 se muestra el ingreso de la clave, que se realiza presionando los sensores de tacto en un cierto orden. Y por último en la Figura 69 se muestra la pantalla del brick luego de haber introducido la clave correctamente.

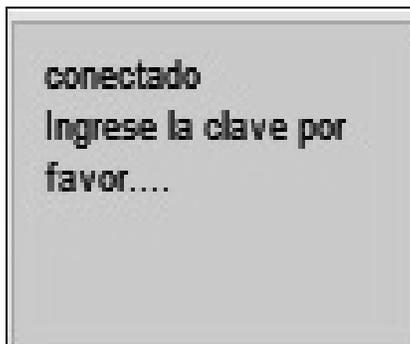


Figura 67 Mensaje emitido al abrir la puerta

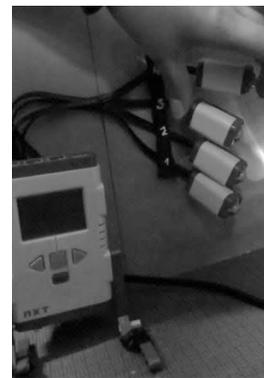


Figura 68 Ingresar la clave

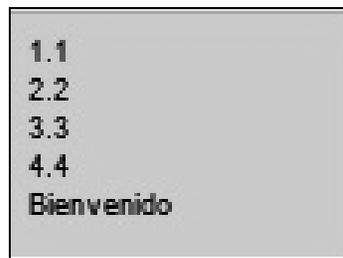


Figura 69 Mensaje emitido al aceptar la clave

7 Conclusiones

Controlar los hogares desde el lugar de trabajo u otros espacios, otorga cierto grado de seguridad y comodidad a muchos usuarios. En este sentido, utilizando la domótica es posible realizar diversas tareas como: apagar luces, abrir/cerrar ventanas o puertas, detectar movimiento y otras actividades.

Para realizar estas tareas se pueden usar diversas tecnologías, y para este proyecto se utilizaron dos. La primera tecnología, es la robótica, en éste caso se utilizaron Robots Lego, los que pueden complementar la domótica, ya que comparten una gran variedad de conceptos y principios, como pueden ser los sensores y actuadores. En la práctica, ambas técnicas incorporan sensores que utilizan para interactuar con su entorno, además de motores que le permiten hacer cambios en un espacio determinado. La segunda tecnología que se integra a la domótica aportando precisión y entidades son los Sistemas Multiagente. Estos tienen ciertas tareas establecidas, por lo que le aportan a la domótica perfiles de trabajo que le permiten armonizar el ambiente acorde a las necesidades del usuario.

Con respecto a la primera herramienta es inusual relacionar de modo práctico la compatibilidad entre ella y la domótica, ya que existen otros protocolos quizás un poco más accesibles que logran resultados bastante similares. Para este proyecto se determinó usar la robótica de forma más específica con los robots lego, por dos razones: en primer lugar, estos son un recurso que posee la Escuela de Informática de la Pontificia Universidad Católica de Valparaíso. En segunda instancia, se pretende demostrar que la domótica no se limita a las tecnologías más usadas, sino que se puede emplear de diversas formas.

La etapa de diseño en un proyecto, ya sea, de un sistema o prototipo, es fundamental, ya que este puede ayudar a ordenar las ideas que rodean a los objetivos del proyecto, además logra dar una visión general del sistema que se desea desarrollar a futuro, y finalmente puede permitir detectar errores en etapas tempranas del proyecto. De esta forma, se puede planear algún tipo de plan de contingencia para el desarrollo el sistema. El diseño inclusive puede ayudar a comprender de una manera más clara la complejidad del problema, pudiendo considerar tópicos que en un principio se pasaron por alto.

En el transcurso de proyecto 1, se dejó establecido el estudio teórico y estado del arte que engloba el proyecto, además se diseñó, gran parte del sistema multiagente que se desarrollarán en la siguiente etapa, y finalmente se planteó la idea del prototipo, con dos pruebas que ayudaron a validar el sistema. Estas pruebas se pensaron para poder abarcar diversos puntos de la domótica, dándole un principal énfasis a la seguridad y al confort.

En la etapa de implementación, en Proyecto 2, se vio lo fácil que es programar teniendo el diseño terminado. Aunque hay diversos problemas que no se podían prever en la etapa de diseño, los cuales provocaron algunos problemas que retrasaron la programación del prototipo. Pero pese a todos estos percances, se logró salir delante de buena manera, gracias a la investigación realizada por cada contratiempo.

8 Referencias Bibliográficas

[Ana Mas, 2005] Ana Mas, *Agentes de Software y Sistemas Multiagente: Conceptos, arquitectura y aplicaciones*, Publicado por Prentice Hall, Madrid, 2005.

[Jara, 2009] Jara, C. *Arquitectura domótica utilizando dispositivos X10 y comunicación mediante Web Services*, Trabajo de título, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile, 2009.

[Cossen, 2003] Cossentino, M. y Sabatucci, L. *Modeling Notation Source PASSI*, Foundation for Intelligent Physical Agents Foundation for Intelligent Physical Agents, Italia, 2003

[Iglesias, 1998] Iglesias, C. Garijo, M. González, J. Velasco, J. *Analysis and design of Multiagent systems using MAS-Common KADS*, Proc. 4th Int. Workshop on Agent Theories, Architectures, and Languages, España, 1998.

[J.P.Cofré, 2010] Cofré, J. Moraga, G. *Usabilidad y Accesibilidad Aplicada a Herramientas Domóticas*, Trabajo de título, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile 2010.

[K. Sycara, 1998] Sycara, K. *Multiagent System*, Artificial Intelligence Magazine, Volumen 19, No. 2 Intelligent Agent Summer, páginas 79-92, USA, 1998.

[K. Sycara, 1999] Brugali, D. Sycara, K. *Agent Technology: A New Frontier for the Development of Application Frameworks?*, editors *Object-Oriented Application Frameworks*. John Wiley 1999.

[M. Hu, 1962] Hu, Ming-kuei. *Visual Pattern Recognition by Moment Invariants*, IRE Transaction on Information Theory, Febrero 1962, IRE Transactions on Information Theory, volume 8, página 179-187.

[M. Wooldridge, 1999] Wooldridge, M. *Multiagent System: A Modern Approach to Distributed Artificial Intelligence*. Editado por. G. Weiss, publicado por MIT Press, Massachusetts, USA. 1999.

[M. Wooldridge, 2000] Wooldridge, M. *Agent-Oriented Software Engineering: The State of the Art.*, Lecture Notes in Computer Science, volume 1957, año 2001, páginas 1-28.

[M. Wooldridge, 2002] Wooldridge, M. *Econometric Analysis of Cross Section and Panel Data*, Publicado por MIT Press, Cambridge, London, England, Massachusetts, USA. 2002.

[Neches, 1991] Neches, R. *Enabling Technology for Knowledge Sharing*. Publicado en AI Magazine, vol. 12 No 3, 1991.

[Poslad, 2001] Poslad S., Buckle P., Hadingham R., *Open Source, Standards and Scaleable Agencies*, International Workshop on Infrastructure for Scalable Multi-Agent Systems, Oxford, 2001.

[Rizo, 2007] Rizo, R. Llorens F. y Pujol, M. *Arquitecturas y Comunicación entre Agentes*, Material del Curso Agentes Inteligentes, Depto. De Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, España, 2007.

[Romero C., 2005] Romero C., Vázquez F., Castro C. (2005). *Domótica e Inmótica: Viviendas y Edificios Inteligentes*. Madrid, España: RA-MA Editorial.

[Searle, 1969] Searle, John. *Speech acts: An Essay in the Philosophy of Language*, Publicado por Cambridge University Press, United Kingdom, 1969

[S. Russel, 1999] Russel, S y Norvig P. *Artificial Intelligence: A Modern Approach*. Publicado por Prentice Hall, 1era Edition, 1999.

[Russel-Norvig, 2003] Russel, S. Norvig, P. *Artificial Intelligence a Modern Approach*, Publicado por Prentice Hall, New Jersey, USA, 2003.

[W. Stallings, 2004] Stallings, William. *Comunicaciones y Redes de Computadores*. Publicado por Prentice Hall, 7ma Edition, Madrid, 2004.

[Y. Shoham, 1999] Shoham, Y. *Agent-oriented Programming Artificial Intelligence*. Publicado por MIT press, Massachusetts, USA. 1999.

[Radioelectronica, 2011] Sitio web que contiene la Figura 4 Actuadores: Relé se puede visitar desde <http://www.radioelectronica.es/images/art001/rele.html>, visitada por última vez el 19 de marzo del 2013.

[Rae, 1714] Sitio web oficial de la Real Academia Española, desde este se consultaron definiciones de algunos conceptos, se puede visitar desde www.rae.es, visitada por última vez el 19 de marzo del 2013.

[RobotC, 2005] Sitio web oficial de RobotC se puede acceder desde <http://www.robotc.net/download/>, visitado la última vez el 19 de marzo del 2013.

[Shoptronica, 2011] Sitio web que contiene la Figura 6 Actuadores: Reguladores o Dimmers se puede visitar desde http://www.shoptronica.com/images/Dimmer%20lt3200-700mA_00.jpg, visitado la última vez el 19 de marzo del 2013.

[Solac, 2011] Sitio web que contiene la Figura 9 Actuador: Resistencia Eléctrica se puede visitar desde <http://www.solac.com/images/noticias/Secador%20SP7160.jpg>, visitado por última vez el 05 de abril del 2012.

[Upv, 2011] Sitio web de la Figura 27 Organización Metodología Gaia se puede visitar desde <http://www.upv.es/sma/teoria/metodologias/Metodologias.pdf>, visitado por última vez el 28 de noviembre del 2012.

[AgileUP, 2011] Sitios web referentes a la metodología Ágil UP, se pueden visitar desde los sitios que se listan a continuación <http://www.cc.una.ac.cr/AUP/html/overview.html> Figura 1 Fases e Hitos de AUP <http://www.cc.una.ac.cr/AUP/images/phasesAndMilestones.gif> Tabla 2

Fases de AUP <http://www.cc.una.ac.cr/AUP/html/phases.html>, visitadas por última vez el 10 de Marzo de 2013.

[Artesanal, 2011] Sitio web de la Figura 30 Prototipo de Escenario se puede visitar desde http://www.foroartesanal.com.ar/search.php?search_author=lindossuenios visitadas por última vez el 28 de Junio de 2012.

[Birtingenio, 2011] Sitio web de la Figura 2 Sensores se puede visitar desde <http://www.bitsingenio.com/wp-content/uploads/2011/09/sensores1.jpg>, visitadas por última vez el 10 de Marzo de 2013.

[CasaDomo, 1999] Sitio web del portal de CasaDomo, El portal del Edificio y Hogar Digital. www.casadomo.com visitada por última vez el 10 de Marzo de 2013.

[Dccia, 2011] Sitio web de la Figura 26 Esquema del ciclo mental interno de un agente BDI se puede visitar desde <http://www.dccia.ua.es/dccia/inf/asignaturas/AI/docs/SMA.pdf>, página 9, visitada por última vez el 10 de Marzo de 2013

[Directindustry, 2011] Sitio web de la Figura 7 Actuadores: Electroválvulas se puede visitar desde <http://www.directindustry.es/prod/sms-tork/electrovalvulas-para-vapor-14091-30346.html>, visitado por última vez el 10 de Marzo de 2013.

[Domoticaviva, 2011] Sitio web del portal Domótica Viva sitio especializado en domótica, se puede visitar desde <http://www.domoticaviva.com/PHP/newsphp.php?id=761>, visitado por última vez el 10 de Marzo de 2013.

[Domotica.tk, 2011] Sitio web del portal Domotica.tk, Las Nuevas Tecnologías al servicio de hombre en el hogar, se puede visitar desde http://www2.udec.cl/~carlosalvial/domotica/pages/que_es.htm, visitado por última vez el 10 de Marzo de 2013.

[Dte.Upct, 2011] Sitio web de la Figura 9 Interfaz, se puede acceder desde <http://www.dte.upct.es/investigacion/proyectos/imagen/domotica2.gif>, visitado por última vez el 10 de Marzo de 2013.

[D. Usuarios, 2011] Sitio web del portal Domótica usuarios, Revista de Domótica, se puede acceder desde <http://www.domoticausuarios.es/>, visitado por última vez el 10 de Marzo de 2013.

[Eclipse, 2012] Sitio web del IDE para desarrollo Eclipse, se puede visitar desde <http://www.eclipse.org/>, visitado por última vez el 10 de Marzo de 2013.

[Electromoda, 2011] Sitio web de la Figura 8 Actuator: Resistencia Eléctric, se puede visitar desde <http://www.electromoda.com/oportunidades/45-145-large/calector-halogeno-hasta-1600w-5-tubos.jpg> visitado por última vez el 10 de Marzo de 2013.

[Erbol, 2011] Sitio web de la Figura 18 Agente se puede visitar desde http://3.bp.blogspot.com/_P4qey_zbHIo/SnHqUj0VNCI/AAAAAAAAABo/deQAwnbJcjQ/s320/2.bmp, visitado por última vez el 10 de Marzo de 2013.

[Hogartintorero, 2011] Sitio web de la Figura 7 Actuadores: Electroválvulas se puede acceder desde http://www.hogartintorero.com/Recambios/Paginas_generales/Electrovalvulas/8300125_Electrovalvula_G.jpg, visitado por última vez el 10 de Marzo de 2013.

[Icar, 1923] Sitio oficial de Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte prestazioni. Se puede acceder desde www.pa.icar.cnr.it/passi/index.html <http://www.pa.icar.cnr.it/passi/Passi/PassiIndex.html>, visitado por última vez el 10 de Marzo de 2013.

[Home, 2011] Sitio oficial de Autodesk Homestyler, se puede acceder desde <http://es.homestyler.com/home> visitado por última vez el 10 de Marzo de 2013.

[Imple, 2011] Las imágenes referenciadas fueron extraídas de distintas páginas, se pueden acceder desde: El sol <http://www.ueom.com/wp-content/uploads/dibujos-del-sol-1.jpg> La Ventana <http://dibujos.net/images/painted2/2011009/e595a02736e1beaa979524566c7d9054.png> Y las piezas de Lego [LegoImag, 2001] visitadas por última vez el 10 de Marzo de 2013.

[Jade, 2003] Sitio oficial de la plataforma de agentes Java Agent Development Framework (JADE), se puede acceder desde <http://jade.tilab.com/>, visitada por última vez el 10 de Marzo de 2013.

[LEGO, 2006] Sitio web oficial del kit de robótica LEGO mindstorms NXT se puede visitar desde <http://mindstorms.lego.com/en-us/whatisnxt/default.aspx> <http://mindstorms.lego.com/en-us/support/files/default.aspx>, visitado por última vez el 10 de Marzo de 2013.

[LegoImag, 2001] Sitio web de las imágenes de lego Mindstorms NXT, se pueden visitar desde http://complubot.educa.madrid.org/pruebas/lego_nxt_version_educativa/lego_nxt.jpg http://cfievalladolid2.net/tecno/cyr_01/robotica/images/pag1_2.jpg, visitadas por última vez el 28 de Junio de 2012.

[LegoRCX, 1998] Sitio web de la imagen del robot LEGO Mindstorms RCX se puede acceder desde http://cfievalladolid2.net/tecno/cyr_01/robotica/lego.htm visitados por última vez el 28 de Junio de 2012.

[Lejos, 2006] Sitio web oficial de la api Lejos NXJ se puede visitar desde <http://lejos.sourceforge.net/nxj.php> <http://lejos.sourceforge.net/nxt/nxj/tutorial/Preliminaries/Intro.htm> visitado por última vez el 28 de Junio de 2012.

[Mahersolhogar, 2011] Sitio web de la imagen Figura 7 Actuador: Motores Eléctricos se puede visitar desde http://www.mahersolhogar.com/upload/fotos/productos/ventilador_331229550_grande.jpg visitado por última vez el 28 de Junio de 2012.

[Moeller, 2011] Sitio web de la imagen Figura 4 Actuadores: Contactores. Se puede visitar desde <http://www.moeller.com.mx/images/AnimXstart/4.JPG> visitado por última vez el 28 de Junio de 2012.